

*Ralf Eickhoff*

***Fehlertolerante neuronale  
Netze zur Approximation  
von Funktionen***

**Bibliografische Information Der Deutschen Bibliothek**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar

©Heinz Nixdorf Institut, Universität Paderborn – Paderborn – 2007

Das Werk einschließlich seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung der Herausgeber und des Verfassers unzulässig und strafbar. Das gilt insbesondere für Vervielfältigung, Übersetzungen, Mikroverfilmungen, sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

Satz und Gestaltung: Ralf Eickhoff

Hersteller: W. V. Westfalia Druck GmbH  
Druck · Buch · Verlag  
Paderborn

Printed in Germany

# Geleitwort

Verbindendes Forschungsziel der von mir geleiteten Fachgruppe Schaltungstechnik ist der systematische Entwurf und der bedarfsgerechte Einsatz von mikroelektronischen Systemen in konkreten Anwendungen der Informations- und Automatisierungstechnik. Unsere Aktivitäten umfassen Arbeiten auf System- und Schaltkreisebene sowohl in digitaler als auch analoger Schaltungstechnik. Besondere Berücksichtigung finden massiv-parallele Realisierungsvarianten sowie die Bewertung der Ressourceneffizienz entsprechender Implementierungen. Ressourceneffizienz bedeutet hier, mit den physikalischen Größen Raum, Zeit und Energie sorgfältig umzugehen.

Die enormen Fortschritte in der Mikroelektronik werden in naher Zukunft die Integration von über 1 Milliarde Transistoren auf einem Quadratcentimeter Siliziumfläche erlauben. Bei effektiven Kanallängen der MOS-Transistoren von deutlich unter 100 Nanometern und bei Schichtdicken des Gate-Oxids von wenigen Atomlagen gelangt man bei der Fertigung allerdings zunehmend an die Grenzen der Silizium-Technologie. Insbesondere die Zuverlässigkeit der realisierten Bauelemente wird bei derart winzigen Strukturgrößen zunehmend ein Problem, da es immer unwahrscheinlicher wird, dass alle Bauelemente einer komplexen Schaltung fehlerfrei funktionieren. Vorbilder für fehlertolerante Systeme sind in der Natur im Verlaufe der biologischen Evolution in großer Anzahl realisiert worden. Es liegt daher nahe, natürliche Wirkprinzipien in die moderne Technik zu übertragen. In unserem Forschungsschwerpunkt „Kognitronik“ konzentrieren wir uns diesbezüglich auf die Modellierung neuronaler Informationsverarbeitungsprinzipien und deren Nutzung in technischen Systemen. Neuronale Netzwerkmodelle bedienen sich intern einer massiv-parallelen Informationsverarbeitung und haben im Allgemeinen eine modulare, reguläre Systemstruktur.

Herr Eickhoff befasst sich in seiner Dissertation mit der Frage, inwieweit ausgewählte neuronale Netzwerkmodelle auch die Eigenschaft der Robustheit und der Fehlertoleranz aufweisen. Speziell analysiert er diese Eigenschaften für lokale Basisfunktionsnetzwerke am Beispiel der Funktionsapproximation, indem er die Robustheit neuronaler Netze auf die gleichgradige Stetigkeit von Funktionen zurückführt und eine formale Abschätzung der Auswirkungen von Rauschen in den Eingängen und den internen Modellparametern auf die Ausgabe der betrachteten Netzwerkmodelle durchführt. Herr Eickhoff weist nach, dass bei freier Wahl der Parameter die beiden ausgewählten Netzwerkmodelle (RBF, LCNN) einen beliebig großen Fehler im Ausgang produzieren können, d. h., nicht robust im Sinne der geforderten gleichgradigen Stetigkeit sind. Soll diese erreicht werden, müssen die Netzwerkparameter - hier insbesondere die Gewichte der Neuroneneingänge - beschränkt werden. Er zeigt ferner auf, wie sich Robustheit und Fehlertoleranz durch eine Erhöhung der Redundanz, d.h. durch zusätzliche Neurone im Netzwerk, verbessern lassen. Wie viele

und an welche Stellen zusätzliche Neurone eingefügt werden sollen, entscheidet er über die Bewertung der Bedeutung, die dem Anteil des Neurons am Ausgabefehler entspricht, und der Empfindlichkeit einzelner Neurone gegenüber Rauschen bzw. Parameterschwankungen. Die Ergebnisse seiner Analysen lässt er abschließend in konkrete Schaltungsentwürfe einfließen.

Herr Eickhoff hat seine Dissertation im Graduiertenkolleg „Automatische Konfiguration in offenen Systemen“ durchgeführt. Der Deutschen Forschungsgemeinschaft sei an dieser Stelle noch einmal ausdrücklich für die finanzielle sowie strukturelle Unterstützung und das damit verbundene Vertrauen gedankt.

Prof. Dr.-Ing. Ulrich Rückert

# Fehlertolerante neuronale Netze zur Approximation von Funktionen

Zur Erlangung des akademischen Grades

DOKTORINGENIEUR (Dr.-Ing.)

der Fakultät Elektrotechnik, Informatik und Mathematik

der Universität Paderborn

vorgelegte Dissertation

von

Dipl.-Ing. Ralf Eickhoff

aus Anröchte

Referent: Prof. Dr.-Ing. Ulrich Rückert

Korreferent: Prof. Dr.-Ing. Klaus Meerkötter

Tag der mündlichen Prüfung: 11. Juli 2007

Paderborn, den 9. August 2007

D14/230



# Danksagung

Diese Arbeit entstand während meiner Zeit als Stipendiat im Graduiertenkolleg „Automatische Konfigurierung in offenen Systemen“ des Heinz Nixdorf Instituts in der Fachgruppe Schaltungstechnik an der Universität Paderborn. Ich möchte mich herzlich bei der Deutschen Forschungsgemeinschaft für die Gewährung des Stipendiums bedanken.

Mein besonderer Dank gilt allen Personen, die zu dem Gelingen dieser Arbeit beigetragen haben und mich in dieser Zeit unterstützt haben. Ein besonderer Dank gilt meinem Doktorvater, Prof. Dr.-Ing. Ulrich Rückert, der mir die Anfertigung dieser Arbeit in seiner Fachgruppe ermöglichte. Ebenso möchte ich mich herzlich bei Prof. Dr.-Ing. Klaus Meerkötter für die Übernahme des Korreferats sowie für seine Hinweise und Anmerkungen bedanken.

Meinen ehemaligen Kollegen danke ich für das angenehme Arbeitsklima und die außerfachlichen Unternehmungen. Ein besonderer Dank gilt dabei meinen Kollegen Markus und Tim für die zahlreichen Diskussionen.

Ein herzlicher Dank gilt meiner Familie und Marion, die mich in dieser Zeit unterstützt und immer wieder neu motiviert haben, und somit in einer besonderen Art und Weise zu der Arbeit beigetragen haben.

Ralf Eickhoff

Dresden, den 9. August 2007



## Zusammenfassung

Aufgrund neuester Erkenntnisse in den kognitiven Wissenschaften und dem Biologieingenieurwesen ist es möglich, die Funktionsweise und die Zusammenhänge in den grundlegenden biologischen Systemen besser zu verstehen. Parallel zu dieser Entwicklung ermöglichen die aufkommenden Halbleitertechnologien und neuartige Bauelemente einen Entwurf von komplexen Systemen mit einer Vielzahl an Komponenten, was den aktuellen Trend eines System-On-Chip übersteigen wird. Zusätzlich nimmt die Zuverlässigkeit der Bauelemente in diesen Fertigungsprozessen ab, was zusammen mit der hohen Komplexität den Einsatz adaptiver Systeme erfordert. In dieser Arbeit wird die Fragestellung untersucht, ob sich Systeme, die biologischen Prinzipien nachempfunden sind, für einen Einsatz unter diesen Randbedingungen eignen und Alternativen zu klassischen Entwurfsverfahren darstellen. Im Speziellen werden dabei die Robustheit und die Fehlertoleranz dieser neuartigen, informationsverarbeitenden Systeme untersucht und Verfahren vorgestellt, mit denen diese Eigenschaften verbessert werden können. Dafür werden Maße entwickelt, mit denen sich die Qualität der Robustheit und Fehlertoleranz bewerten lässt. Weiter wird der Einsatz dieser biologischen Prinzipien in aufkommenden Silizium-Technologien anhand analoger und digitaler Entwurfstechniken analysiert.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Funktionsapproximation durch neuronale Netze</b>	<b>5</b>
2.1	Funktionsapproximation . . . . .	5
2.2	Neuronale Netze . . . . .	12
2.2.1	Multilayer Perceptron . . . . .	12
2.2.2	Local Cluster Neuronal Network . . . . .	16
2.2.3	Basisfunktionen . . . . .	20
2.3	Technologische Randbedingungen . . . . .	26
2.3.1	Skalierung der CMOS-Technologie . . . . .	27
2.3.2	Alternativen zu CMOS . . . . .	30
2.3.3	Störungen und Rauschen . . . . .	33
2.3.4	Fehler und Defekte . . . . .	35
<b>3</b>	<b>Analyse von Fehlertoleranz und Robustheit</b>	<b>37</b>
3.1	Robustheit neuronaler Netze . . . . .	37
3.1.1	Basisfunktionen . . . . .	39
3.1.2	LCNN . . . . .	49
3.1.3	Parameterempfindlichkeit . . . . .	54
3.2	Fehlertoleranz . . . . .	58
3.2.1	Fehlerquellen . . . . .	58
3.2.2	Ausfallwahrscheinlichkeit . . . . .	64
3.2.3	Auswirkungen auf neuronale Netze . . . . .	68
3.3	Ausfallwahrscheinlichkeit . . . . .	75
3.3.1	Statische Defekte . . . . .	75
3.3.2	Dynamische Fehler . . . . .	77
3.4	Zusammenfassung und Diskussion . . . . .	85

<b>4</b>	<b>Methodologie zur Fehlertoleranz und Robustheit</b>	<b>87</b>
4.1	Erhöhen der Fehlertoleranz . . . . .	87
4.1.1	Bedeutung von Neuronen bei einem RBF . . . . .	88
4.1.2	Bedeutung der Neuronen beim LCNN . . . . .	99
4.1.3	Kontrolle der Modellkomplexität . . . . .	104
4.2	Erhöhen der Robustheit . . . . .	116
4.2.1	Empfindliche Parameter . . . . .	116
4.2.2	Empfindliche Neuronen . . . . .	120
4.3	Robustes und fehlertolerantes Lernen . . . . .	124
4.3.1	Robustheit als Zielfunktion . . . . .	125
4.3.2	Fehlertoleranz als Zielfunktion . . . . .	135
4.3.3	Signifikanz als Randbedingung . . . . .	140
4.4	Zusammenfassung und Diskussion . . . . .	143
<b>5</b>	<b>Technische Umsetzung</b>	<b>145</b>
5.1	Numerische Betrachtungen . . . . .	145
5.1.1	Konditionierung der Gaußfunktionen . . . . .	145
5.1.2	Polynomdarstellung und Horner-Schema . . . . .	147
5.1.3	Numerische Stabilität . . . . .	149
5.2	Entwurf neuronaler Systeme . . . . .	154
5.2.1	Digitaler Entwurf . . . . .	155
5.2.2	Analoger Entwurf . . . . .	170
5.3	Auswirkungen und Grenzen bedingt durch die Technologie . . . . .	173
5.3.1	Genauigkeit . . . . .	173
5.3.2	Synthesergebnisse . . . . .	176
5.3.3	Rauschen . . . . .	181
5.3.4	Vergleich der digitalen und analogen Realisierungen . . . . .	194
5.4	Zusammenfassung und Diskussion . . . . .	198
<b>6</b>	<b>Zusammenfassung</b>	<b>201</b>
	<b>Glossar</b>	<b>205</b>

<b>Abbildungsverzeichnis</b>	<b>212</b>
<b>Tabellenverzeichnis</b>	<b>214</b>
<b>Literaturverzeichnis</b>	<b>215</b>



# Einleitung

Fehler passieren uns jeden Tag. Jedoch besitzen wir die Fähigkeit, aus diesen zu lernen und sie beim nächsten Mal zu vermeiden. Auch können wir mit unserer Erfahrung uns nicht vertraute Situationen bewältigen und so zusätzliche Erkenntnisse sammeln. Diese verschaffen uns einen Gewinn an Wissen, um in Zukunft für unbekannte Situationen gewappnet zu sein. Auch der stetige Lernprozess, der mit unserer Geburt begonnen hat, hat uns über die Jahre mit einzigartigen Fähigkeiten ausgestattet, die uns in der Bewältigung des alltäglichen Lebens und neuen Situationen zur Seite stehen.

Alltägliche Beispiele zeigen unsere Stärken durch diesen Lernprozess. Beim Golfspiel wird versucht, einen Ball (Durchmesser 42,67 mm) mit möglichst wenigen Schlägen in ein mehrere hundert Meter entferntes Loch (Durchmesser 10,8 cm) zu spielen. Dafür gibt es für jede Spielbahn eine Vorgabe (PAR), die für einen guten Golfer die durchschnittliche Schlagzahl angibt. Für das Einhalten der Vorgabe bleibt kaum Raum für Fehler. Bei einem 500 m langem PAR 5 schlägt ein guter Spieler den Ball ca. 200 m weit. Um mit dem zweiten Schlag eine gute Ausgangsposition zu haben, darf bei dem ersten Schlag der Ball nicht mehr als 25 m seitlich von der idealen Bahn abweichen, da er sonst durch Hindernisse blockiert wird oder im hohen Gras liegt. Nach einfacher Trigonometrie darf der Ball beim Starten höchstens eine Winkeldifferenz von  $7,13^\circ$  aufweisen. Trotz dieser hohen Anforderungen gelingt es vielen Spielern dennoch, diese Genauigkeit einzuhalten und den Ball auf dem Fairway zu platzieren.

Die Aussagen aus dem alltäglichen Leben lassen sich ebenfalls auf technische Systeme übertragen. Auch hier wird der Raum für Fehler und Ungenauigkeiten immer enger, so dass katastrophale Folgen entstehen können, wie z.B. der Absturz einer Ariane 5 [Gle96] oder das Versagen einer Patriot-Rakete [Gen92]. Aber nicht nur bei komplexer Software spielen Fehler eine große Rolle. Aufgrund der extrem gestiegenen Komplexität über die letzten Jahre können Fehler innerhalb einer Systemarchitektur eines Prozessors nicht mehr vollständig ausgeschlossen werden. Gerade durch die fortgeschrittene Miniaturisierung werden sehr komplexe Systeme auf einem einzigen integrierten Baustein ermöglicht. Strukturgrößen im Nanometerbereich erlauben eine höhere Anzahl von Bauelementen auf derselben Fläche eines integrierten Bausteins als noch vor ein paar Jahren. Durch neue Materialien und Techniken lassen sich Systeme auf Basis von Molekülen herstellen, welche die

elektrische Funktionalität selbst übernehmen. Dies führt zu einer enorm hohen Anzahl von verfügbaren, identischen Basiselementen. Andererseits gelangt man bei der Fertigung an die Grenzen der Silizium-Technologie, so dass für eine hohe Zuverlässigkeit einzelner Bauelemente entweder hohe Kosten aufgrund einer komplexen und exakten Fertigung entstehen oder die Fehleranfälligkeit der Bauelemente ansteigen werden. Zudem wirken sich bei Strukturgrößen im Nanometerbereich thermische Störungen oder Hintergrundrauschen verstärkt auf die Genauigkeit der Parameter eines Bauelementes aus.

Diese Umstände erfordern neue Konzepte bei dem Entwurf von Systemen; sei es durch neuartige Werkzeuge, die bei der Entwicklung unterstützend eingreifen, oder neuartige Architekturen, welche auf geänderte Randbedingungen reagieren. Aufgrund der beobachteten Fähigkeiten des Menschen kann die Biologie Hilfestellungen für neuartige Entwurfsmethoden bieten. Durch Erkenntnisse im Bereich der Wahrnehmung oder Empfindung können so Prinzipien und Lernverfahren genutzt werden, welche den Aufbau von adaptiven Architekturen ermöglichen. Mit Hilfe von Wahrnehmungen können so Fehler von dem System selbstständig erkannt und mit Hilfe von Lernverfahren darauf angemessen reagiert werden. Dadurch kann die Toleranz des Systems gegenüber Fehlern erhöht werden, und ebenfalls wird der Entwurf komplexer Systeme vereinfacht, weil aufgrund der Adaptivität Veränderungen vom System selbst vorgenommen werden können.

Weiter weisen der Aufbau des Gehirns und die Möglichkeiten neuartiger Technologien mehrere Gemeinsamkeiten auf, so dass eine Übertragung biologischer Prinzipien auf technische Systeme denkbar scheint. Das Gehirn besteht aus einer großen Anzahl von Neuronen, die auf vielfältige Art und Weise über Axone bzw. Synapsen untereinander verbunden sind. So wird eine adaptive, parallele und energieeffiziente Verarbeitung von Informationen gewährleistet [Arb02, VBTG05]. Durch neuste Erkenntnisse auf dem Gebiet biologischer, neuronaler Netze und kognitiver Systeme lassen sich so gänzlich neuartige Architekturen zur Informationsverarbeitung einsetzen, die zudem aufgrund ihrer Fähigkeiten, wie Adaptivität und Wahrnehmung, Lösungen für die Probleme der fortgeschrittenen Miniaturisierung liefern.

Zielsetzung dieser Arbeit ist die Untersuchung von Fehlertoleranz und Robustheit künstlicher neuronaler Netze. So soll die Eignung dieser Netze für innovative Lösungskonzepte für die Probleme der zunehmenden Miniaturisierung in der Halbleiterfertigung näher betrachtet werden. In Anlehnung daran werden die Auswirkungen der technischen Umsetzung auf die Funktionsweise der neuronalen Netze anhand analoger und digitaler Architekturen untersucht. Bei diesen Analysen stehen besonders statische Neuronenmodelle, wie die radialen Basisfunktionen und das Local Cluster Neural Network, im Fokus der Untersuchungen. Als Anwendung wird die Approximation von Funktionen durch diese Netze betrachtet, da viele Anwendungsgebiete neuronaler Netze auf diese Form zurückgeführt werden können. Weiter

ist die Darstellung boolescher Funktionen für die Schaltungstechnik besonders interessant, da sich so neue Konzepte beim Logikentwurf ergeben.

Die Untersuchungen konzentrieren sich auf drei zentrale Fragestellungen

- Auswirkungen von Fehlern und Störungen auf die Eigenschaften der Approximation durch neuronale Netze
- Methoden zur Verbesserung dieser Eigenschaften der neuronalen Netze
- Einflüsse der Technologieentwicklung,

in denen die Eignung der neuronalen Netze für den Einsatz in der Nanoelektronik näher bestimmt wird. So wird in Kapitel 2 in die notwendigen Grundlagen zum besseren Verständnis der Analysen eingeführt. Ausgehend von den mathematischen Beschreibungen der Funktionsapproximation werden die Zusammenhänge zu künstlichen, neuronalen Netzen aufgezeigt. Weiter werden die Rahmenbedingungen der Arbeit beschrieben, welche sich durch Charakterisierung der Technologieentwicklung ergeben. Die Auswirkungen dieser Rahmenbedingungen auf neuronale Netze werden in Kapitel 3 untersucht. Dafür werden die Effekte, welche sich aufgrund von Störungen und fehleranfälligen Elementen ergeben, auf die Approximationseigenschaften der Netze abgeschätzt. Kapitel 4 stellt in Anlehnung daran mögliche Vorgehensweisen vor, die zur Verbesserung der Robustheit und Fehlertoleranz neuronaler Netze eingesetzt werden können. Speziell wurde darauf geachtet, dass diese Methoden unabhängig vom verwendeten neuronalen Netz und dessen Trainingsverfahren eingesetzt werden können. Die Umsetzung neuronaler Verfahren auf digitale und analoge Schaltungstechniken wird in Kapitel 5 vorgestellt. Im Besonderen wird auf die Handhabbarkeit und die Wiederverwendbarkeit der Systeme geachtet und deren Skalierung auf Basis der *Complementary Metal Oxide Semiconductor*, CMOS, untersucht. Die Arbeit endet mit der obligatorischen Zusammenfassung in Kapitel 6.



# Funktionsapproximation durch neuronale Netze

In diesem Kapitel wird auf die Approximation von Funktionen eingegangen. Zunächst werden die mathematischen Grundlagen der Approximationstheorie vorgestellt, wobei die verwendeten Begrifflichkeiten und Definitionen eingeführt werden. Im Fokus dieser Arbeit stehen für diesen Zweck die neuronalen Netze. In Bezug auf die Funktionsapproximation wird auf die Eigenschaften neuronaler Netze näher eingegangen, und die verwendeten Topologien der verschiedenen Netze dieser Arbeit werden vorgestellt.

Die verwendeten Modelle werden kurz erläutert, wobei diese Arbeit ausschließlich statische Neuronenmodelle aufgrund ihrer mathematischen Darstellungsform verwendet. Hierbei wird besonders auf die Multilayer Perceptrons und die Basisfunktionen eingegangen, wozu als Besonderheit neben dem Multilayer Perceptron das *Local Cluster Neural Network* betrachtet wird. Dabei wird kurz auf die Eigenschaften der entsprechenden Netze und mögliche Trainingsverfahren eingegangen. Weitere und ausführliche Informationen zu diesem Themengebiet finden sich in der Literatur [Hay99, Bis95b, Arb02].

## 2.1 Funktionsapproximation

Bei der Approximation einer Funktion soll eine Funktion  $v$  durch andere Funktionen genähert werden. Im Allgemeinen ist die Funktion  $v$  ein Element eines normierten Raumes  $X$  mit der Norm  $\|\cdot\|$ , wobei jene Funktion durch Funktionen aus dem Teilraum  $P \subset X$  dargestellt werden soll [Sau01].

Bei der Approximation einer Funktion mittels anderer Funktionen existiert eine Vielzahl von mathematischen Fragen hinsichtlich qualitativer Aussagen über den Fehler der Approximation, die Existenz einer *Bestapproximation* oder Aussagen über die Dichte der Funktionenklassen, beschrieben durch die einfachen Objekte. Dabei wird bei der Approximation eine Beschreibung so bestimmt, dass eine Norm der Differenz zwischen der gesuchten Funktion  $f$  und der zu approximierenden Funktion  $v$  minimal wird. Daneben kann eine Funktion auch so konstruiert werden,

indem eine Menge von Punkten<sup>1</sup>  $\{(\mathbf{x}^i, y^i) | v(\mathbf{x}^i) = y^i, \forall i\}$  mit der Funktion an den angegebenen Stellen übereinstimmt.

Der Fehler bei einer Approximation der Funktion<sup>2</sup>  $v(x)$  durch die Folge von Funktionen  $f_n(x)$  wird durch

$$F_n(v) = \inf_{f \in P_n} \|v - f\| \quad (2.1)$$

mit  $P_0 \subseteq P_1 \subseteq P_2 \subseteq \dots \subseteq P_n$  beschrieben, wobei für den Grenzwert gelten sollte

$$\lim_{n \rightarrow \infty} F_n(v) = 0 \quad (2.2)$$

und die Funktion  $v$  somit approximiert werden kann.  $P_n$  ist eine Folge von Approximationsräumen, so z.B. der Vektorraum der trigonometrischen Funktionen

$$P_n = \text{span}\{1, \cos x, \sin x, \dots, \cos nx, \sin nx\}. \quad (2.3)$$

Dabei gilt für eine Norm  $\|\cdot\|$  für ein Element  $v \in X$  [BSM05, Pla04]

1.  $\|v\|_X \geq 0 \quad \forall v \in X$
2.  $\|v\|_X = 0 \Leftrightarrow v = 0$
3.  $\|\alpha v\|_X = |\alpha| \|v\|_X \quad \forall v \in X, \alpha \in \mathbb{R}$
4.  $\|v + f\|_X \leq \|v\|_X + \|f\|_X \quad \forall v, f \in X$  (Dreiecksungleichung).

Dabei sind typische Räume, in denen solche Approximationsprobleme betrachtet werden, durch

$$L_p(X) = \left\{ f : X \rightarrow \mathbb{R} : \int_X |f|^p dt < \infty \right\} \quad (2.4)$$

beschrieben, wobei als Norm die Integralnormen

$$\|f\|_p = \left( \int_X |f(t)|^p dt \right)^{\frac{1}{p}} \quad 1 \leq p \leq \infty \quad (2.5)$$

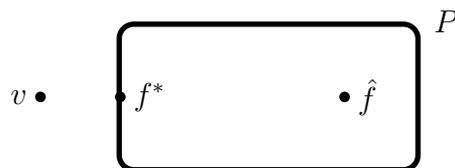
verwendet werden.

Eine wichtige Eigenschaft bei der Approximation ist die Dichte eines Unterraumes. Mit Hilfe dieser Eigenschaft kann jede Funktion beliebig genau approximiert werden, und die Definition lautet [Tka03]

**Definition 1** (dicht). *Eine Menge  $A \subset C(X)$  heißt gleichmäßig dicht in  $C(X)$ , dem metrischen Raum aller stetigen Funktionen, falls für jedes  $v \in C(X)$  und jedes  $\epsilon > 0$  ein  $f \in A$  existiert, so dass  $|v(x) - f(x)| < \epsilon$  für alle  $x \in X$ .*

<sup>1</sup> Der Exponent bezeichnet das jeweilige Element aus der Menge. Der Index bezeichnet dagegen den jeweiligen Eintrag eines Vektors.

<sup>2</sup> Die Problematik wird hier in einer Dimension betrachtet und lässt sich auf mehrere Dimensionen übertragen.



**Abbildung 2.1:** Verdeutlichung der Best-Approximation

Für eine Funktion  $f$  aus der Menge  $A$  gilt somit

$$\forall \epsilon > 0 \quad \forall v \in C(X) \quad \exists f \in A : \|f - v\|_\infty < \epsilon. \quad (2.6)$$

Als Konsequenz aus (2.6) bedeutet dies, dass jede Funktion  $f$  aus dem Unterraum  $A$  die Funktion  $v$  beliebig genau approximieren kann. Neben der Konvergenz des Fehlers existiert in jedem normierten Raum  $X$  eine *gleichmäßig beste Approximation*, die wie folgt definiert ist [Sau01]

**Definition 2** (Best-Approximation). *Für eine Teilmenge  $\emptyset \neq P \subset X$  eines normierten Raumes  $(X, \|\cdot\|)$  und ein Element  $v \in X$  wird  $f^*$  die Best-Approximierende zu  $v$  genannt, falls gilt*

$$f^* \in P, \quad \|f^* - v\| = \underbrace{\inf_{f \in P} \|f - v\|}_{=: F_v(P)}. \quad (2.7)$$

$F_v(P)$  bezeichnet dabei die minimale Distanz des Elements  $f$  zu der Teilmenge  $P$ .

Abbildung 2.1 veranschaulicht die Definition der Best-Approximierenden. Die Funktion  $v$  kann nicht exakt durch die Menge von Funktionen aus dem Unterraum  $P$  approximiert werden, wobei die Funktion  $f^*$  hinsichtlich einer Norm den minimalen Abstand zur Funktion  $v$  besitzt. Somit stellt  $f^*$  die optimale Approximation dar. Dagegen bildet die Funktion  $\hat{f}$  ebenfalls eine Approximation zur Funktion  $v$ , wobei allerdings ein größerer Abstand angenommen wird. Die Approximation ist somit nicht optimal. Hierbei ist besonders die Existenz und die Eindeutigkeit der Best-Approximierenden von Interesse sowie die Bestimmung der Funktion  $f^*$  oder die Abschätzung hinsichtlich der minimalen Distanz.

Antworten auf derartige Fragen sind in der Mathematik bewiesen worden, und es kann gezeigt werden, dass für jede kompakte Untermenge eines normierten Raumes für jedes Element dieses Raumes eine Best-Approximierende existiert. Die Eindeutigkeit dieser optimalen Approximation ist für streng konvexe Mengen erfüllt [Pla04]. Existiert in dem Unterraum  $P$  genau eine Best-Approximierende, so heißt  $P$  *Existenzmenge* und besitzt die Eigenschaft der Best-Approximation. Für weitere Eigenschaften bezüglich der streng konvexen Menge oder der Best-Approximierenden sei auf die Literatur verwiesen [Sau01, Pla04].

Einer der wichtigsten Sätze der Approximationstheorie ist der Satz von Weierstraß [Ber12, Bis61], welcher die Approximation einer Funktion mittels Polynome behandelt. Dieser besagt, dass in jedem kompakten Intervall  $I \subset \mathbb{R}$  zu einer stetigen Funktion  $v$  und jedem  $\epsilon > 0$  ein Polynom  $p$  existiert, so dass gilt

$$\|v - p\| < \epsilon. \quad (2.8)$$

Dies bedeutet, dass der Untervektorraum, welcher durch Polynome  $n$ -ten Grades aufgespannt wird, die Dichte-eigenschaft besitzt und so jede Funktion beliebig genau approximieren kann. In der Literatur existieren eine Vielzahl von Beweisen und Erweiterungen für den Satz von Weierstraß [Sau01].

Die Aussagen über die Approximationseigenschaften, wie z.B. die Best-Approximierende, hängen von dem jeweiligen Unterraum ab, der durch die Menge von Funktionen aufgespannt wird. Neben dem Einsatz von Polynomen lassen sich auch weitere Unterräume finden, welche ebenfalls die Dichte-eigenschaft besitzen und so jede Funktion beliebig genau approximieren können. Neuronale Netze spannen z.B. einen solchen Untervektorraum auf.

*Beispiel.* Die Sachverhalte zur Approximation und der weiteren Kapitel sollen anhand einer Funktion beispielhaft verdeutlicht werden. Als Funktion wird die si-Funktion verwendet, die wie folgt definiert ist

$$\text{si}(x) = \begin{cases} \frac{\sin(x)}{x}, & \text{für } x \neq 0 \\ 1, & \text{für } x = 0 \end{cases}. \quad (2.9)$$

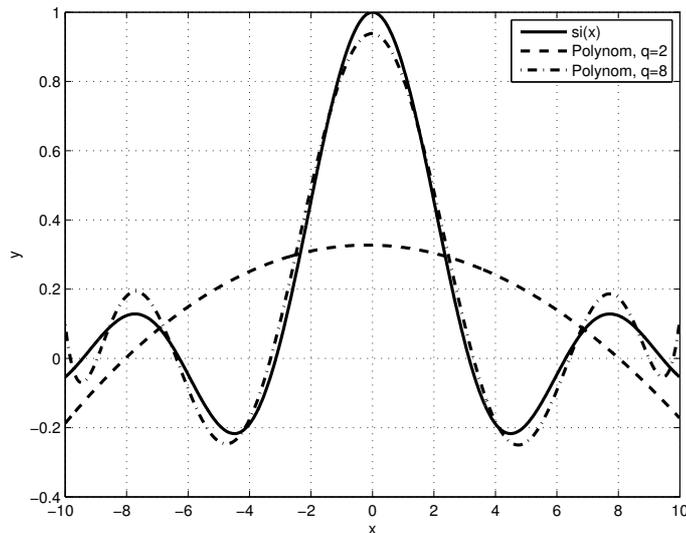
Dabei wird für alle Darstellungen die Funktion im Intervall  $[-10; 10]$  betrachtet, und eine Menge von 1000 gleichverteilten Testpunkten  $\{x^i; y^i = \text{si}(x^i)\}$  erzeugt. Abbildung 2.2 zeigt die Approximation der si-Funktion durch zwei Polynome, wobei die Eigenschaft ausgenutzt wurde, dass es sich bei der si-Funktion um eine gerade Funktion handelt. So ergeben sich für ein Polynom zweiten Grades

$$p(x) = a_2x^2 + a_0$$

insgesamt zwei freie Parameter und für ein Polynom achten Grades

$$p(x) = a_8x^8 + a_6x^6 + a_4x^4 + a_2x^2 + a_0$$

fünf freie Parameter. Für das Polynom zweiten Grades erhält man einen absoluten Fehler  $\Delta y = 0,6564$ , und für das Polynom achten Grades  $\Delta y = 0,1553$ . Da i.A. der funktionale Zusammenhang nicht bekannt ist, kann der Grad des Polynoms nicht eindeutig gewählt werden. Ein zu kleiner Grad erzeugt zu große Fehler, wobei hohe Grade unnötig komplexe Modelle liefern und stark oszillieren können.



**Abbildung 2.2:** Approximation der si-Funktion durch Polynome

Weiter kann man bei der Approximation von Funktionen zwischen zwei verschiedenen Modellen unterscheiden. Hier kann entweder ein parametrisches oder ein nicht-parametrisches Modell für die Darstellung der Funktion  $v$  genutzt werden [Bis95b]. Ein Ansatz, um die Funktion  $v(x)$  zu beschreiben, ist das parametrische Modell. Bei diesem Vorgehen wird *a priori* festgelegt, welche spezielle funktionale Form  $v(x)$  besitzen soll. Zudem wird davon ausgegangen, dass die Beziehung durch eine endliche Anzahl von Parametern beschrieben werden kann, wobei der Zusammenhang explizit oder implizit dargestellt werden kann.

Eine Alternative zu diesem Ansatz stellt das nicht-parametrische Modell dar. Dies bedeutet, dass die funktionale Form der Beziehung nicht festgelegt wird und dadurch eine flexible Form besitzt. Aufgrund der flexiblen Form lassen sich für den Zusammenhang mehrere Möglichkeiten im Gegensatz zu den parametrischen Modellen finden, wobei eine unbeschränkte Anzahl von Parametern zur Approximation zugelassen wird. Zusätzlich müssen keine Voraussagen über die Beobachtungen wie bei dem parametrischen Modell gemacht werden, was ohne zusätzliches Wissen über das Problem schwierig ist. Weitere Vorteile eines nicht-parametrischen Modells sind die Filterung der Daten und die Interpolation. So werden die Einflüsse von Ausreißern in den Daten abgeschwächt, und es erfolgt eine Interpolation von fehlenden Daten. Allerdings wird der Zusammenhang zwischen der Approximation und (einigen) essentiellen Parametern bei diesem Modell nicht berücksichtigt [Bis95b].

Das parametrische Modell setzt somit die Kenntnis des funktionalen Zusammenhangs voraus, wobei als Aufgabe das Finden der Parameter des Modells bestehen

bleibt. Das nicht-parametrische Modell dagegen blendet diesen Zusammenhang vollständig aus und versucht nur, die Funktion zu approximieren.

Bei der Approximation von Funktionen sind diese nur in den wenigstens Fällen explizit bekannt. So sind z.B. bei der Aufzeichnung von Sensordaten oder Prozessdaten nur einzelne Punkte der Funktion bekannt, so dass die Aufgabe in der Rekonstruktion des Zusammenhangs zwischen den Eingabevektoren und den Ausgabedaten liegt. Damit können unbekannte Werte zwischen diesen Testpunkten mit Hilfe des funktionalen Zusammenhangs rekonstruiert werden. Weiter lassen sich auch Werte außerhalb des durch die Testpunkte bekannten Bereiches ermitteln.

In diesem Fall wird der funktionale Zusammenhang durch eine Menge von Testdaten beschrieben, womit allgemein folgt

$$y^i = v(\mathbf{x}^i) + \epsilon^i \quad i = 1, \dots, M \quad (2.10)$$

bzw. charakterisiert als eine Menge von Punkten

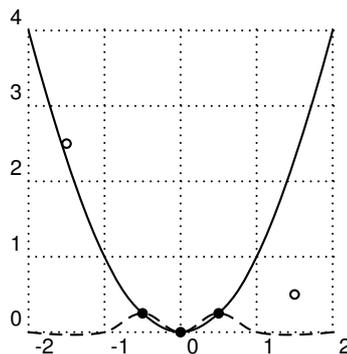
$$d = \left\{ (\mathbf{x}^i, y^i) \in \mathbb{R}^n \times \mathbb{R} \right\}_{i=1}^M. \quad (2.11)$$

Meistens sind die Testdaten durch Messungenauigkeiten oder anderweitigen Ungenauigkeiten mit Störungen  $\epsilon^i$  überlagert. Wie man an (2.11) erkennen kann, ist die Verwendung eines parametrischen Modells zur Rekonstruktion ohne Kenntnisse des funktionalen Zusammenhangs schwer durchführbar.

Dabei liegen die Testdaten zumeist nur innerhalb eines bestimmten Bereiches vor, z.B. innerhalb des Intervalls  $[a; b]$ , so dass für Werte außerhalb dieses Intervalls,  $x < a \vee x > b$ , keine Informationen bereitstehen. Hier gibt es zwei unterschiedliche Charakterisierungen, welche für diesen unbekanntem Bereich unterschiedliche Werte liefern. Beiden Methoden gemeinsam ist jedoch ihr Verhalten innerhalb des Intervalls  $[a; b]$ . Hier werden die Daten interpoliert bzw. approximiert, wobei durch das Modell Werte für die Punkte zwischen den Testpunkten geliefert werden.

Zuerst kann die Funktion  $v$  durch eine globale Approximation  $f$  dargestellt werden. Hier wird jene durch die Funktion  $y = f(\mathbf{x})$  angenähert, wobei außerhalb des durch Testpunkte bekannten Bereiches Vorhersagen für mögliche Funktionswerte gemacht werden. So existieren außerhalb des Intervalls  $[a; b]$  Funktionswerte, wobei die Approximationsgüte vom Verlauf und der Gutartigkeit der Funktion  $v$  abhängig ist. Als Konsequenz aus diesem Vorgehen wird der Verlauf der Funktion nicht nur innerhalb des Intervalls  $[a; b]$ , sondern auch außerhalb bei der Variation eines Parameters beeinflusst.

Daneben kann zur Darstellung der Funktion  $v$  die lokale Approximation verwendet werden. Innerhalb des Intervalls  $[a; b]$  werden wie zuvor die Testdaten approximiert, während im Gegensatz zur globalen Approximation außerhalb des Bereiches keine



**Abbildung 2.3:** Vergleich lokale (- -) und globale (-) Approximation

Vorhersagen bezüglich möglicher Funktionswerte gemacht werden. Die Funktion  $y = f(\mathbf{x})$  nimmt in diesem Bereich den Wert Null an.

Abbildung 2.3 veranschaulicht den Unterschied zwischen einer lokalen und einer globalen Approximation einer Funktion  $v$ , welche in diesem Beispiel durch drei Punkte

$$\left\{ (-0,5; 0,25), (0; 0), (0,5; 0,25) \right\}$$

beschrieben wird. Außerhalb des durch Testpunkte abgedeckten Intervalls liefert die lokale Funktion keine von Null verschiedenen Werte, während die globale Funktion versucht, Aussagen über den möglichen Verlauf der Funktion zu erstellen. Allerdings können diese Aussagen ungenügend sein. Das Verhalten der Approximation außerhalb des bekannten Intervalls wird auch als Generalisierungseigenschaft bezeichnet [Hay99, Bis95b].

Des Weiteren kann die Funktion  $v$ , welche durch die Testpunkte aus (2.11) beschrieben wird, nicht nur approximiert, sondern die Punkte auch interpoliert werden. Der Unterschied zwischen beiden Verfahren liegt in der Verwendung der Testpunkte als Randbedingungen. Bei der Interpolation muss die Funktion aus dem Unterraum  $P$  die Funktionsgleichung  $y^i = v(\mathbf{x}^i) = f(\mathbf{x}^i)$  erfüllen, während bei der Approximation die Norm der Differenz zwischen beiden Funktionen minimiert wird. Hierbei muss die Funktion  $f$  nicht durch die vorgegebenen Punkte laufen, vgl. z.B. eine Ausgleichsgerade [BSM05]. Beide Verfahren liefern Funktionswerte für die zwischen den Testpunkten liegenden Stellen.

Die Bestimmung der Parameter für die Interpolation kann auf unterschiedliche Art und Weise erfolgen und hängt von dem gewählten Verfahren ab. So können beispielsweise bei der Interpolation mittels eines Polynoms die Testpunkte als Stützstellen verwendet werden, womit sich ein Polynom vom Grad  $M - 1$  (vgl.(2.11)) ergibt. Als mögliche Methoden seien hier die Lösung eines linearen Gleichungssystems sowie die Interpolation nach Lagrange, Neville oder Newton angegeben [Pla04]. Bei einer hohen Anzahl von Punkten erhält man ein Polynom mit hohem Grad, was für die Darstellung der Funktion zu komplex sein kann [Bis95b]. Weiter wirken sich Stö-

rungen in den Daten (vgl. (2.10)) negativ auf die Interpolation aus, da die Funktion  $f$  durch diese Punkte verlaufen muss.

Bei der Approximation wird die Norm der Differenz zwischen der Funktion  $v$  und  $f$  minimiert. So erhält man ein Optimierungsproblem, welches zusätzlich durch Randbedingungen, z.B.  $y^i = f(\mathbf{x}^i)$  für die Interpolation, erweitert werden kann. Zur Lösung dieses Problem können bekannte mathematische Verfahren, wie das Gradientenverfahren oder Sekantenverfahren, genutzt werden [HUL93]. Wird die Funktion  $f$  durch eine lineare Überlagerung von Funktionen dargestellt, z.B. durch Kerne [Aro50], wobei die einzigen Parameter durch die Gewichtung der Funktionen bestimmt werden, kann die Minimierung der euklidischen Distanz auf die Lösung eines linearen Gleichungssystems (LGS) zurückgeführt werden [Hay99].

## 2.2 Neuronale Netze

In diesem Abschnitt wird der Zusammenhang zwischen den biologischen Modellen neuronaler Netze und ihren künstlichen Pendanten erläutert. Das fundamentale Interesse der Forschung über neuronale Netze besteht im Wesentlichen auf dem Aspekt, die Funktionsweise des Gehirns zu verstehen. Künstliche, neuronale Netze sind hierbei mathematische Konstrukte, die durch das Wissen und tiefere Erkenntnisse aus der Biologie beeinflusst werden. Da an dieser Fragestellung Wissenschaftler unterschiedlicher Fachrichtungen arbeiten, existiert eine Vielzahl von Blickrichtungen, Ansätzen und Modellen, wodurch sich dieses Themengebiet zu einem interdisziplinären Forschungsbereich ausgeweitet hat. Ausgehend von den biologischen Grundlagen werden die für diese Arbeit relevanten künstlichen Neuronenmodelle vorgestellt und ihr Zusammenhang mit den mathematischen Sichtweisen aus dem vorangegangenen Abschnitt erläutert. Für tiefer führende Erläuterungen besonders im Hinblick auf den Aufbau und die (biologische) Funktionsweise von Nervenzellen oder des Nervensystems sei auf die Literatur verwiesen [Arb02, Seg98, RT97, Sch97a]. Die hier verwendeten Bezeichnungen für die Modelle und deren Eigenschaften sind an [Hay99] angelehnt.

### 2.2.1 Multilayer Perceptron

Ein *Multilayer Perceptron* (MLP) stellt die Verallgemeinerung des *Singlelayer Perceptrons* (SLP) aus [Ros58] dar, wobei sich nur Unterschiede in der Topologie ergeben und nicht in der Verarbeitungsweise. Dabei besteht das künstliche Netzwerk in Anlehnung zu seinem biologischem Vorbild aus elementaren Berechnungseinheiten<sup>3</sup> (Neuronen), die durch Verbindungen (Synapsen) Informationen untereinander

---

<sup>3</sup> Hier werden nur vereinfachte Modelle betrachtet, wobei auch durch die Verbindungen eine Verarbeitung der Informationen geschehen kann.

austauschen können, wobei die Signalerzeugung biologischer Neuronen durch die künstliche Umsetzung simuliert werden soll.

So besteht das SLP aus einem einzigen Neuron, welches über gewichtete Kanäle Signale von den Eingängen in Analogie zu dem Empfang elektrischer Signale in der Biologie erhält. Zudem kann zusätzlich ein konstanter Eingang für die Weiterleitung von Informationen berücksichtigt werden. Eine wichtige Eigenschaft des Neurons ist, dass diese Gewichtung der Eingänge verändert werden kann und so ein Lernprozess oder im technischen Sinne eine Adaption existiert. Die gewichteten Eingänge werden im Neuron summiert und auf dieses Ergebnis wird die Aktivierungsfunktion des Neurons angewendet, wodurch die Stärke des Signals am Ausgang bestimmt wird. Wichtige Eigenschaften dieser Aktivierungsfunktion sind der nichtlineare Charakter der Funktion und die Begrenzung des Ausgangssignals.

Die mathematische Formulierung dieses Neuronenmodells ergibt sich wie folgt. Durch die Gewichtung  $w_{kj}$  der Eingänge  $x_j, \forall j = 1, \dots, n$  und deren Summation ergibt sich ein Empfangssignal des  $k$ -ten Neurons

$$u_k = \sum_{j=1}^n w_{kj}x_j + b_k = \sum_{j=0}^n w_{kj}x_j, \quad (2.12)$$

wobei  $k$  die Anzahl von parallelen Neuronen insgesamt bezeichnet. Bei dem SLP können mehrere Neuronen parallel arbeiten. Jedes Neuron stellt eine eigene Ausgabe dar, die nicht wie einem MLP in einer weiteren Ebene gewichtet überlagert wird. In (2.12) wurde ein konstanter Term vorgesehen, der jedoch durch eine mathematische Manipulation als zusätzlicher, konstanter Eingang verstanden werden kann. Aus dem Ergebnis der Summation wird die Stärke des Ausgangssignals mit Hilfe der Aktivierungsfunktion bestimmt

$$y_k = \sigma(u_k) = \sigma \left( \sum_{j=0}^n w_{kj}x_j \right). \quad (2.13)$$

Für die Aktivierungsfunktion  $\sigma(x)$  aus (2.13) können unterschiedliche Funktionen verwendet werden. Als Pionierarbeit auf diesem Gebiet verwendet das McCulloch-Pitts-Modell [McC43] die Heaviside-Funktion, so dass die Ausgabe auf Null und Eins begrenzt ist. Weiter werden in der Literatur bevorzugt sigmoide<sup>4</sup> (S-förmige) Funktionen, wie die logistische oder die tanh-Funktion, verwendet. So ergibt sich als Aktivierungsfunktion des Neurons

$$\sigma(\iota, x) = \frac{1}{1 + \exp(-\iota \cdot x)} \quad \text{oder} \quad (2.14)$$

$$\sigma(\iota, x) = \tanh(\iota \cdot x), \quad (2.15)$$

<sup>4</sup> Im Weiteren wird die sigmoide Funktion als Synonym für die logistische Funktion verwendet.

womit der Parameter  $\iota$  die Steilheit der Funktion beeinflusst und so als zusätzlicher Parameter<sup>5</sup> des Netzes angenommen werden kann. Die Wahl sigmoider Aktivierungsfunktionen hat unterschiedliche Gründe. Zum einen bieten sie eine gute Balance zwischen linearem und nichtlinearem Verhalten [MMMR96], zum anderen sind sie biologisch motiviert. Wird die mittlere Ausgangspulsrate eines biologischen Neurons, also der Ausgangswert des künstlichen Neurons, gegenüber seinem Eingangsstrom, vergleichbar mit den summierten Eingängen aus (2.12), aufgetragen, so ergibt sich ein sigmoides Verhalten [Arb02]. Abbildung 2.4(b) zeigt die sigmoiden Aktivierungsfunktionen für das SLP. Daneben können für die Funktion  $\sigma(x)$  auch stückweise lineare oder komplett lineare Funktionen verwendet werden, wodurch sich für den letzten Fall die Theorie der Netze dann auf die lineare Algebra reduziert.

Die Unterschiede des MLPs zu einem SLP bestehen einzig in der Topologie des Netzes. In Abbildung 2.4 ist die Topologie eines MLPs mit den Aktivierungsfunktionen dargestellt. Der Aufbau des Netzes besteht aus mehreren Schichten

1. Eingangsschicht
2. versteckte Schichten (mit variabler Anzahl)
3. Ausgangsschicht

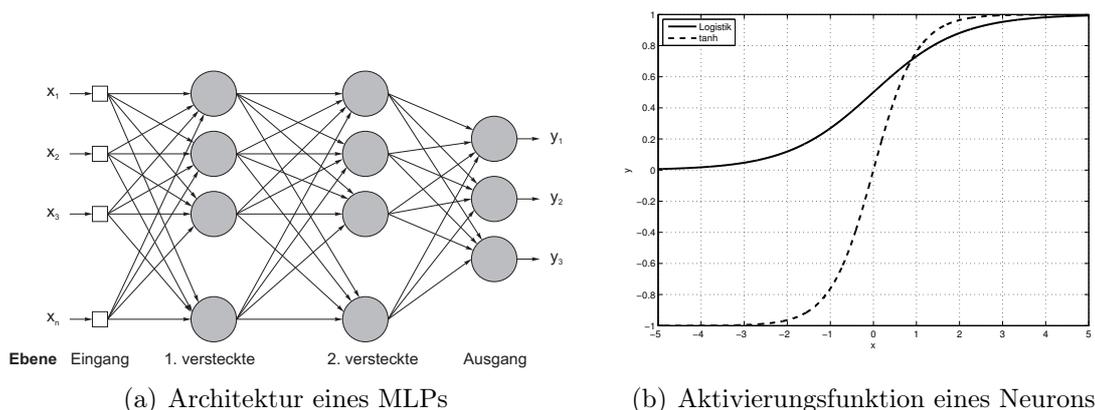
Das SLP besitzt nur eine Eingangsschicht und eine Ausgangsschicht, während beim MLP mindestens eine versteckte Schicht existiert. Dabei ist die Anzahl von Neuronen in den jeweiligen Schichten, anders als die Art der Verbindungen, nicht festgelegt. Für das MLP bzw. das SLP existieren nur Verbindungen zwischen zwei benachbarten Ebenen (vgl. Abbildung 2.4(a)), was bedeutet, dass in diesem Netz keine Rückkopplungen oder Vernetzungen innerhalb einer Ebene existieren, und dieser Typ eines neuronalen Netzes zu den vorwärtsgerichteten, neuronalen Netzen gehört.

Damit kann der funktionale Zusammenhang zwischen den Eingängen und Ausgängen eines MLPs aus Abbildung 2.4(a) mathematisch beschrieben werden. Allerdings wird sich hier auf ein einziges Neuron im Ausgang beschränkt. Ein MLP mit einer versteckten Ebene<sup>6</sup> und  $m$  Neuronen in dieser versteckten Ebene beschreibt aus mathematischer Sichtweise eine Menge von Funktionen bzw. einen Unterraum

$$\mathcal{S} = \left\{ y(\mathbf{x}) \mid y(\mathbf{x}) = \sum_{i=1}^m \alpha_i \sigma \left( \sum_{j=0}^n w_{ij} x_j \right); \alpha_i, w_{ij} \in \mathbb{R}, \sigma : \mathbb{R} \rightarrow \mathbb{R}; m, n \in \mathbb{N} \right\}, \quad (2.16)$$

<sup>5</sup> Die meisten Veröffentlichungen und Modelle verwendet einen festen Wert  $\iota$  für die Steilheit. Deshalb wird im Weiteren  $\sigma(\iota, x) = \sigma(x)$  verwendet. Eine Variation dieses Parameters bildet zusätzlich die bei biologischen Experimenten beobachtete *Intrinsic Plasticity* eines Neurons ab [Tri05].

<sup>6</sup> Aufgrund der Übersichtlichkeit wird auf eine Darstellung von mehreren versteckten Schichten verzichtet. Eine Ausweitung von (2.16) auf diesen Fall ist leicht möglich.



**Abbildung 2.4:** Funktionsweise eines MLPs

wobei  $\alpha_i$  die Gewichtung der Verbindungen von den Neuronen der versteckten Ebene zu dem Ausgang bezeichnet.

(2.16) beschreibt mathematisch gesehen eine Menge von Funktionen, welche für die Approximation einer Funktion  $v$  genutzt werden kann. So besteht in der Formulierung des Problems aus mathematischer und neuronaler Sicht kein Unterschied. Die einzige Differenz für die Blickrichtungen kann in der Bestimmung der Parameter liegen. Für das MLP werden dessen Gewichte zumeist über das Backpropagation-Verfahren [RHW86b] bestimmt, was im Wesentlichen ein Gradientenverfahren darstellt. Hier wird üblicherweise die Leistung des Differenzsignals, welches sich zwischen der Ausgabe des Netzes und der Geforderten bildet, minimiert, indem der Fehler durch das Netz zurückgeführt wird und die Parameter in Richtung des stärksten Abfalls verändert werden. Das MLP kann somit auch als reines mathematisches Konstrukt aufgefasst werden, obwohl Verknüpfungen zur Biologie existieren.

Aufgrund dieses Zusammenhangs war mathematisch die Fragestellung nach der Dichteigenschaft des aufgespannten Raumes aus (2.16) interessant, also ob jede Funktion  $v : \mathbb{R}^n \rightarrow \mathbb{R}$  beliebig genau approximiert werden kann, so dass eine Vielzahl von Veröffentlichungen sich mit dem Beweis dieser Eigenschaft beschäftigte. Ein Teil der Antwort auf diese Frage konnte durch den Beweis von Kolmogorov [Kol57] auf das 13. Problem<sup>7</sup> von Hilbert [Hil00] gefunden werden. Allerdings gibt es hier noch einige Einschränkungen zu beachten, und dieses Theorem wurde auf die neuronalen Netze von Hecht-Nielsen übertragen [HN87]. Weitere Beweise zu der Approximationseigenschaft eines MLPs, teilweise mit Bedingungen an die Aktivierungsfunktion, finden sich in [HSW89, Fun89].

Einige Einschränkungen müssen beim MLP hinsichtlich der Best-Approximierenden hingenommen werden. So wurde die Existenz der Best-Approximierenden für Netze

<sup>7</sup> Im Jahre 1900 hielt David Hilbert beim internationalen Mathematiker-Kongress in Paris eine Rede, in der er 23 mathematische Probleme vorstellte.

mit McCulloch-Pitts-Neuronen in [KKV00] gezeigt, so dass eine globale minimale Distanz zur approximierenden Funktion  $v$  vorhanden ist. Allerdings sind die Eigenschaften der Projektion nicht eindeutig und stetig, so dass keine oberen Abschätzungen für den Fehler gemacht werden können. Weiter wurde in [GP90] bewiesen, dass ein MLP mit sigmoider Aktivierungsfunktion unabhängig von der Wahl der Norm bei der Approximation keine Existenzmenge darstellt. Somit besitzt dieser Netzwerktyp nicht die Eigenschaft der Best-Approximation.

Aufgrund der Struktur des Unterraumes, welcher durch das MLP aufgespannt wird, handelt es sich bei diesem neuronalen Netz um eine globale Approximation [Koi95]. Somit werden außerhalb des durch Testpunkte abgedeckten Bereiches Vorhersagen über die Funktionswerte getroffen, wobei diese Generalisierungseigenschaften bewertet werden können [Hay99].

## 2.2.2 Local Cluster Neuronal Network

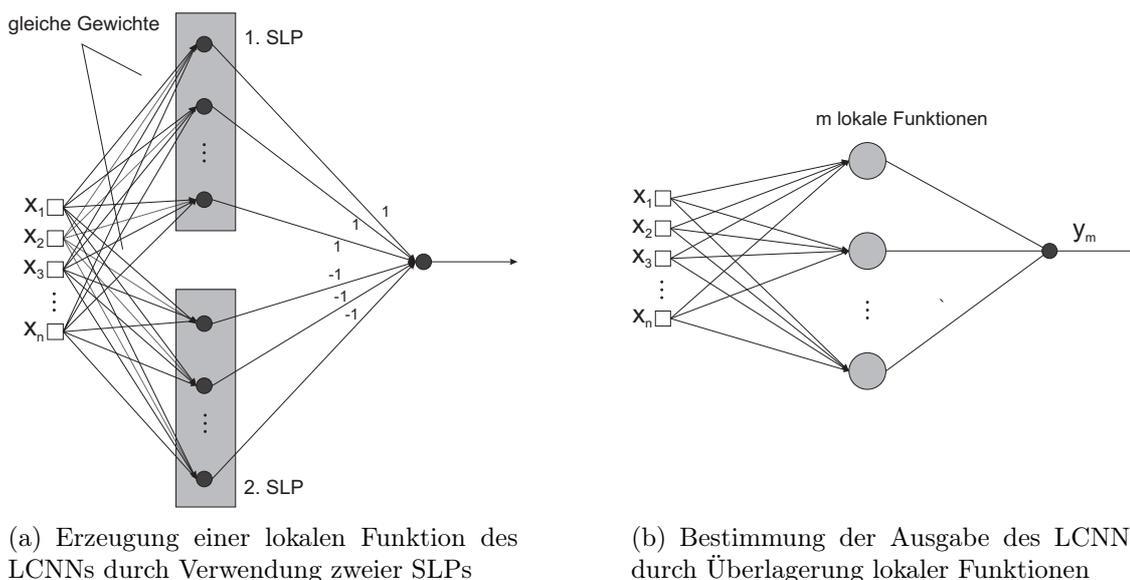
Das Local Cluster Neural Network (LCNN) wurde als eine neuronale Architektur von Geva und Sitte [GS93, GS92] vorgestellt. Die Intention dieser Netzwerkarchitektur war zum einen eine lokal begrenzte Antwort auf Eingaben, ähnlich einem radialen Basisfunktionsnetzwerk (vgl. Abschnitt 2.2.3), zu erhalten, und zum anderen einen einfacheren Entwurf in analoger Hardware als ein Basisfunktionsnetzwerk zu ermöglichen. Im mathematischen Sinne stellt das LCNN ein MLP mit besonderen Eigenschaften für die Parameter dar, so dass es innerhalb dieses Abschnitts vorgestellt wird.

Das LCNN kann als ein MLP mit insgesamt zwei versteckten Schichten<sup>8</sup>, in denen die Gewichte und Verbindungen speziellen Randbedingungen unterliegen, dargestellt werden. Eine wichtige Eigenschaft des LCNNs ist die Verbindungsstruktur zwischen den beiden versteckten Schichten, wobei nur bestimmte Verbindungen zwischen Neuronen erlaubt sind. So stellt das LCNN kein vollständig verbundenes Netz dar. Ziel durch diese besondere Struktur der Gewichtmatrix<sup>9</sup> ist, nur Ausgaben für einen lokal begrenzten Bereich des Eingangsraums zu erhalten. Abbildung 2.5 zeigt den schematischen Aufbau des LCNNs in Analogie zum MLP.

Abweichend von der ursprünglichen mathematischen Herleitung kann das LCNN als ein MLP mit zwei verborgenen Schichten aufgefasst werden. Abbildung 2.5 zeigt die allgemeine Funktionsweise des Netzes durch Überlagerung von lokalen Funktionen. Jede Verbindung ist dabei auch als eine Gewichtung zu verstehen. Die lokalen Funktionen werden durch ein MLP mit zwei versteckten Ebenen erzeugt (Abbildung 2.5(a)). Dafür kann die Formulierung des LCNNs so umgeformt werden, dass dieses

<sup>8</sup> Hier werden die Bezeichnungen für die (versteckten) Schichten aus [Hay99] verwendet. Die Eingänge und die Ausgänge stellen separate Schichten dar.

<sup>9</sup> Die Berechnungen aus (2.16) können auch als Matrixoperation mit einer Gewichtsmatrix dargestellt werden.



(a) Erzeugung einer lokalen Funktion des LCNNs durch Verwendung zweier SLPs

(b) Bestimmung der Ausgabe des LCNNs durch Überlagerung lokaler Funktionen

**Abbildung 2.5:** Struktur des LCNNs durch den Aufbau mittels SLPs

MLP durch zwei parallel arbeitende SLPs entsteht, wobei beide Netze die gleichen Gewichte vom Eingang zu der ersten Schicht besitzen müssen. Weiter handelt es sich bei den Parametern zwischen den beiden verborgenen Ebenen um Konstanten. Man erkennt, dass es sich nicht um ein vollständig verknüpftes Netz handelt.

Die ursprüngliche Herleitung des LCNNs startet nicht mit dem Vergleich zu einem SLP, sondern erfasst die Darstellung der lokalen Funktion. Bei dem SLP berechnet jedes Neuron die gewichtete Summe der Eingänge und wendet darauf die Aktivierungsfunktion an. Die Idee des LCNNs ist, die Aktivierungsfunktionen zweier Neuronen so zu kombinieren, dass eine lokale Funktion entsteht. Dazu wird eine *Ridge-Funktion* gebildet, die aus zwei verschobenen Sigmoiden erzeugt wird. Die Ridge-Funktion wird durch

$$l(\mathbf{w}, \mathbf{r}, k_1, \mathbf{x}, d) = \sigma(k_1, \mathbf{w}^T(\mathbf{x} - \mathbf{r}) + d) - \sigma(k_1, \mathbf{w}^T(\mathbf{x} - \mathbf{r}) - d) \quad (2.17)$$

dargestellt, wobei der Parameter  $d$  zur Verschiebung der Sigmoiden o.b.d.A. zu Eins gewählt wird [GS92]. Als Aktivierungsfunktion für die Neuronen wird die Logistikkfunktion aus (2.14) verwendet. Die Steilheit  $\iota$  der Funktion wird in (2.17) über den Parameter  $k_1$  gesteuert, und die Lage der Ridge-Funktion im Raum wird durch den Parametervektor  $\mathbf{w}$ , dessen Länge und die Position des Zentrums  $\mathbf{r}$  bestimmt.

Die lokale Funktion wird durch die Überlagerung mehrerer Ridge-Funktionen erreicht, welche unterschiedliche Orientierungen besitzen, jedoch die gleiche Position im Raum, d.h. das gleiche Zentrum. So ergibt sich eine Hügel-Funktion

$$f(\mathbf{W}, \mathbf{r}, k_1, \mathbf{x}) = \sum_{i=1}^n l(\mathbf{w}_i, \mathbf{r}, k_1, \mathbf{x}), \quad (2.18)$$

wobei  $\mathbf{W}$  eine  $n \times n$  Matrix<sup>10</sup> darstellt, welche durch die  $n$  Spaltenvektoren  $\mathbf{w}_i$  der Ridge-Funktionen gebildet wird. Diese Funktion besitzt eine Erhebung mit dem Zentrum  $\mathbf{r}$  sowie auslaufenden Ästen. Aus dieser Funktion kann eine lokale Funktion gebildet werden, indem die Äste entsprechend entfernt werden und nur die Erhebung übrig bleibt. Eine wiederholte Anwendung einer geeignet skalierten Logistikkfunktion  $\sigma_o(x) = 2 \cdot \sigma(k_2, x)$  auf die Funktion (2.18) entfernt die Äste und erzeugt eine *lokale Funktion*

$$\mathcal{L}(\mathbf{W}, \mathbf{r}, k_2, \mathbf{x}) = \sigma_o(f(\mathbf{W}, \mathbf{r}, k_1, \mathbf{x}) - b). \quad (2.19)$$

Zur Bestimmung der Ausgabe werden mehrere lokale Funktionen gewichtet überlagert, so dass sich der Ausgabewert des Netzes durch

$$y_m(\mathbf{x}) = \sum_{\nu=1}^m \alpha_\nu \mathcal{L}(\mathbf{W}_\nu, \mathbf{r}_\nu, \mathbf{x}) \quad (2.20)$$

berechnet. Somit beschreibt das LCNN ebenfalls einen Unterraum, welcher durch die Menge von Funktionen

$$\mathcal{S} = \left\{ y_m(\mathbf{x}) \mid y_m(\mathbf{x}) = \sum_{\nu=1}^m \alpha_\nu \mathcal{L}(\mathbf{W}_\nu, \mathbf{r}_\nu, \mathbf{x}), m \in \mathbb{N}, \mathbf{r}_\nu \in \mathbb{R}^n, \mathbf{W}_\nu \in \mathbb{R}^{n \times n} \right\} \quad (2.21)$$

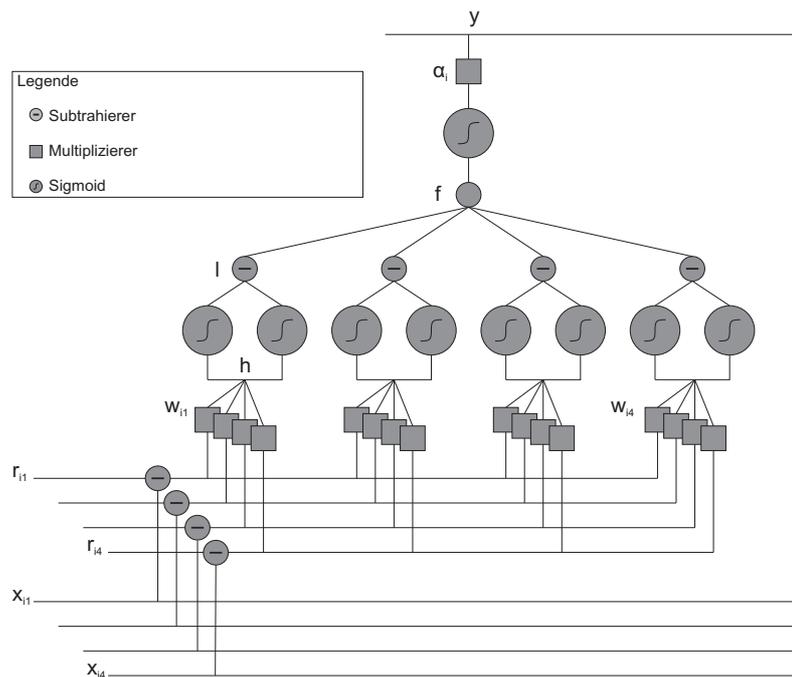
aufgespannt wird.

In Abbildung 2.6 ist der Datenfluss eines LCNNs nach der mathematischen Beschreibung für eine lokale Funktion mit vier Eingängen dargestellt. Die Ausgabe einer lokalen Funktion wird durch vier parallele Ridge-Funktionen bestimmt, welche selbst durch zwei gegeneinander verschobene Sigmoiden erzeugt werden. Der Vektor  $\mathbf{r}$  ermöglicht eine Verschiebung der lokalen Funktion im Eingangsraum.

Aufgrund der beschriebenen Vorgehensweise handelt es sich bei dem LCNN um eine lokale Approximation, wobei Erfahrungen zeigen, dass eine lokale Approximation kleinere Approximationsfehler als bei einem MLP hervorbringt und gleichzeitig ein geringerer Aufwand zur Bestimmung der Parameter benötigt wird [GS93, MD89]. Der geringere Aufwand für das Lernen begründet sich in der Gestalt der Sigmoiden beim MLP. Da diese für die Hälfte des Eingangsbereiches von Null verschiedene Werte annehmen, tragen im Mittel die Hälfte der Neuronen zu der Approximation bei. Somit müssen bei jedem Trainingsschritt die Parameter dieser Neuronen angepasst werden.

Hinsichtlich der Dichteigenschaft und somit der Approximierbarkeit des LCNNs können auf Aussagen für MLPs zurückgegriffen werden, da es sich bei dem LCNN von der Topologie her um ein MLP handelt. In [Kur92] wurde basierend auf dem Theorem von Kolmogorov bewiesen, dass jedes MLP mit zwei versteckten Ebenen

<sup>10</sup>Die Dimension der Matrix ist von der Dimension der Eingänge abhängig.



**Abbildung 2.6:** Verarbeitung der Eingangsdaten einer lokalen Funktion mit 4 Eingängen nach [GS92]

jede Funktion beliebig genau approximieren kann. Eine Besonderheit hierbei ist, dass bis auf die Gewichte der zweiten verborgenen Schicht zu den Ausgängen alle Parameter universell gewählt werden können. Nur für die Ausgangsgewichte müssen die speziellen Werte für die zu approximierende Funktion  $v$  bestimmt werden. Ebenso wurde in [MP99] gezeigt, dass ein MLP mit zwei versteckten Schichten jede stetige Funktion im Einheitswürfel beliebig genau approximieren kann. Darüber hinaus besitzt ein solches MLP mit zwei versteckten Ebenen im Gegensatz zu einem MLP mit einer verborgenen Schicht keine untere Grenze des Approximationsfehlers.

**Beispiel.** Die unterschiedlichen Vorgehensweisen bei der Approximation eines funktionalen Zusammenhangs sollen für ein MLP und das LCNN anhand der si-Funktion verdeutlicht werden. Abbildung 2.7 zeigt die Ausgaben beider Netze nach dem Training, wobei beide Netze mit jeweils 10 Neuronen bzw. lokalen Funktionen mit dem Backpropagation-Verfahren trainiert wurden. Aufgrund der globalen Approximation durch das MLP kann dieses die Nebenmaxima der Funktion mit dieser Anzahl von Neuronen nicht darstellen, und man kann den Vorteil des LCNNs erkennen, welches bedingt durch die lokale Vorgehensweise diese Maxima abbilden kann. Allerdings werden für die 10 lokalen Funktionen insgesamt 30 Sigmoide benötigt, so dass sich für das LCNN 30 freie Parameter und ebenfalls auch für das MLP ergeben. Aufgrund der Topologie lässt sich beim LCNN ein kleinerer Approximationsfehler  $\Delta y = 0,0539$  als beim MLP  $\Delta y = 0,1895$  erreichen.

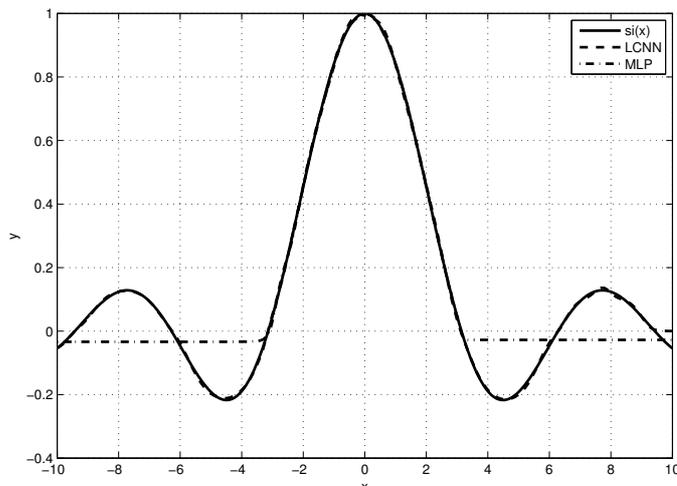


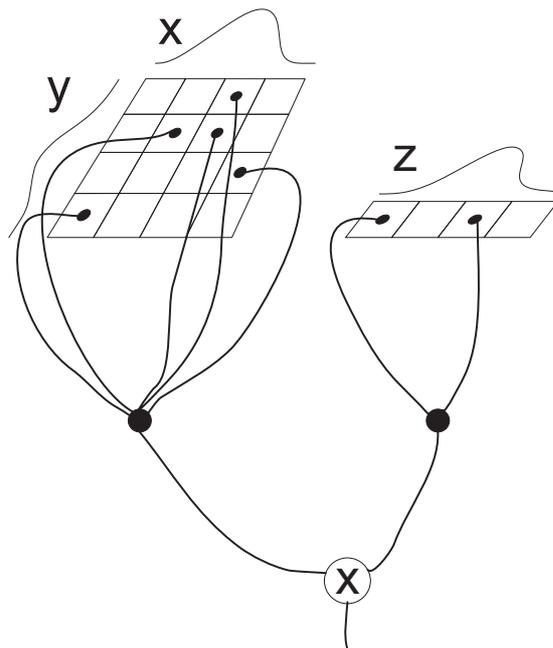
Abbildung 2.7: Approximation der si-Funktion durch ein MLP und ein LCNN

### 2.2.3 Basisfunktionen

Die biologische Motivation für radiale Basisfunktionen entstammt [PG90b]. Dabei ist der Ausgangspunkt eine Dreischichtenstruktur von Nervenzellen. Die Eingabeschicht verarbeitet dabei die Aktivitäten aus dem Merkmalsraum und bildet diese Eingabe auf eine Zwischenschicht ab, wobei die Aktivität mittels gewichteter Verbindungen propagiert wird. Diese sorgen dafür, dass die Aktivität eines Neurons aus der Zwischenschicht vom Abstand der Eingabe zum *virtuellen* Ort des Neurons abhängt. Dieser virtuelle Ort wird als Spezialisierung eines Neurons auf eine bestimmte Eingabe verstanden, und diese Gewichtung sorgt für eine monoton fallende Aktivierung des Neurons in der Zwischenschicht. Erfolgt diese Abnahme unabhängig von der Raumrichtung, so spricht man von radialen Basisfunktionen. Hierfür wird eine Verwendung einer gaußförmigen Aktivitätsfunktion vorgeschlagen [GP90]. Die Aktivierung der Neuronen in der Zwischenschicht wird anschließend durch eine lineare Signalübertragung der Ausgangsschicht zugeführt, welche die gewichtete Summe bildet.

Somit bilden radiale Basisfunktionen ein abstraktes Modell der biologischen Wirklichkeit. Für den exakten Zusammenhang zur Biologie sei auf die Literatur verwiesen [Pog90]. Abbildung 2.8 zeigt den beschriebenen Zusammenhang zwischen den drei Schichten. Hier dienen die Neuronen der Zwischenschicht zur Extraktion der Merkmale aus der Eingangsschicht, um so z.B. die motorischen Fähigkeiten zu steuern.

Neben diesem biologisch motivierten Ansatz der zu gaußförmigen, radialen Basisfunktionen führt, lässt sich dieser Typ neuronaler Netze auch über eine mathe-



**Abbildung 2.8:** Biologische Motivation eines RBFs

matisch fundierte Theorie herleiten. Im Mittelpunkt der Untersuchung steht die Rekonstruktion der Oberfläche, welche durch eine Menge von Testpunkten

$$d = \left\{ (\mathbf{x}^i, y^i) \in \mathbb{R}^n \times \mathbb{R} \right\}_{i=1}^M$$

(vgl. (2.11)) gegeben ist. Die Aufgabe besteht nun darin, die Oberfläche zwischen den Testpunkten abzuschätzen.

Dies ist jedoch ein schlecht gestelltes Problem, da eine Rekonstruktion der Oberfläche anhand der Informationen aus den Testpunkten nicht eindeutig lösbar ist (vgl. Abbildung 2.3). Dafür werden weitere Annahmen getroffen, um ein wohl gestelltes Problem zu erhalten. Eine erste Annahme ist, dass die Daten aus der Abbildung nicht völlig zufällig entstanden sind. Im Allgemeinen sind diese Zuordnungen glatt, was bedeutet, dass kleine Änderungen in den Parametern auch nur kleine Änderungen in den Ausgängen zur Folge haben. Dies ist eine schwache Forderung, welche eine Approximation ermöglicht. Daneben können natürlich weitere Kenntnisse über die Abbildung mit in die Rekonstruktion einfließen [PG89].

Eine Approximationstechnik, welche die Glattheit von Funktionen ausnutzt, ist die Regularisierung. Dabei wird die Rekonstruktion der Oberfläche als die Minimierung eines Funktionals bestehend aus zwei Termen ausgedrückt. Der erste Term erfasst

den Abstand der Funktion zu den bekannten Testpunkten, während der zweite Term die Glattheit der Funktion misst [PG89, PG90a]. Dies führt zu der Minimierung von

$$V(f) = \sum_{i=1}^M (\mathbf{y}^i - f(\mathbf{x}^i))^2 + \lambda \cdot \phi[f], \quad (2.22)$$

wobei  $\lambda$  die Gewichtung beider Terme vornimmt und  $\phi[f]$  ein Glattheitsfunktional, den sogenannten *Stabilisator*, darstellt. Über den Parameter  $\lambda$  kann die Gewichtung der beiden Terme aus (2.22) kontrolliert werden.

Für eine große Klasse von Funktionalen  $\phi[f]$  erhält man Lösungen der gleichen Form für (2.22) [GJP95]. Die Glattheit einer Funktion kann über die Oszillation gemessen werden und als Leistung einer hochpassgefilterten Funktion bestimmt werden. Die Glattheit kann so als

$$\phi[f] = \int_{\mathbb{R}^n} d\mathbf{s} \frac{|\tilde{f}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})} \quad (2.23)$$

definiert werden, wobei  $\tilde{f}, \tilde{G}$  die Fouriertransformierte der jeweiligen Funktion bezeichnen, und  $1/\tilde{G}(\mathbf{s})$  einen Hochpass beschreibt.

Unter der Randbedingung, dass die Fouriertransformierte  $\tilde{G}$  reell und symmetrisch ist, folgt für die Lösungsform von (2.22) [GP90, GPC90]

$$f(\mathbf{x}) = \sum_{j=1}^M \alpha_j g(\mathbf{x} - \mathbf{c}_j) + \sum_{i=1}^k \beta_i \psi_i(\mathbf{x}), \quad (2.24)$$

wobei der erste Term eine gewichtete Überlagerung von (Basis-) Funktionen und der Zweite einen Polynomanteil beschreibt.

Oft wird für den Stabilisator ein rotationsinvariantes Funktional gewählt, so dass dies auf radial symmetrische Funktionen  $g(\|\mathbf{x}\|)$  führt. Je nach Wahl des Funktionals  $\phi[f]$  erhält man für  $g$  unterschiedliche Funktionen. Zudem entfällt der Polynomanteil aus (2.24) für den Fall positiv definiter Funktionen  $g$  [Isk94].

### 2.2.3.1 Gaußförmige Basisfunktionen

Die Wahl des Funktionals beeinflusst die Form der Funktion  $g$ . Für den Stabilisator

$$\phi[f] = \int_{\mathbb{R}^n} d\mathbf{s} e^{\frac{\|\mathbf{s}\|^2}{\beta}} |\tilde{f}(\mathbf{s})|^2 \quad (2.25)$$

mit einem Skalierungsfaktor  $\beta$  erhält man die Gaußfunktion

$$g_j(z) = e^{-\frac{z^2}{2\sigma_j^2}}. \quad (2.26)$$

Da zudem die Gaußfunktion eine positive definite Funktion darstellt, ergibt sich die Rekonstruktion der Oberfläche zu

$$f(\mathbf{x}) = \sum_{j=1}^M \alpha_j e^{-\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{\sigma_j^2}}. \quad (2.27)$$

Die radialen Basisfunktionen (RBF)<sup>11</sup> mit Gaußfunktionen beschreiben somit ebenfalls eine Menge von Funktionen

$$\mathcal{S} = \left\{ f_m(\mathbf{x}) \mid f_m(\mathbf{x}) = \sum_{i=1}^m \alpha_i \exp\left(\frac{-\|\mathbf{x}-\mathbf{c}_i\|^2}{2\sigma_i^2}\right); m \in \mathbb{N}; \sigma_i \in \mathbb{R} \setminus \{0\}; \right. \\ \left. \alpha_i \in \mathbb{R}; \mathbf{c}_i \in \mathbb{R}^n \right\}. \quad (2.28)$$

Für den Fall  $\mathbf{x} = \mathbf{c}_i$  und  $\sigma_i = 0$  wäre die Funktionsmenge aus (2.28) nicht wohl definiert und würde nicht zu der Menge der stetigen Funktionen gehören. Somit wird dieser Fall durch die entsprechende Einschränkung von  $\sigma_i$  ausgeschlossen.

(2.28) stellt eine vereinfachte Form der verallgemeinerten, gaußförmigen, radialen Basisfunktionen dar. Anstelle der euklidischen Norm ist es in einigen Anwendungen sinnvoll, eine allgemeine Norm zu verwenden [PG90a]

$$\|\mathbf{x}\|_C^2 = (\mathbf{C}\mathbf{x})^T(\mathbf{C}\mathbf{x}) = \mathbf{x}^T \mathbf{C}^T \mathbf{C} \mathbf{x}, \quad (2.29)$$

wobei  $\mathbf{C}$  eine  $n \times n$ -Gewichtsmatrix darstellt.

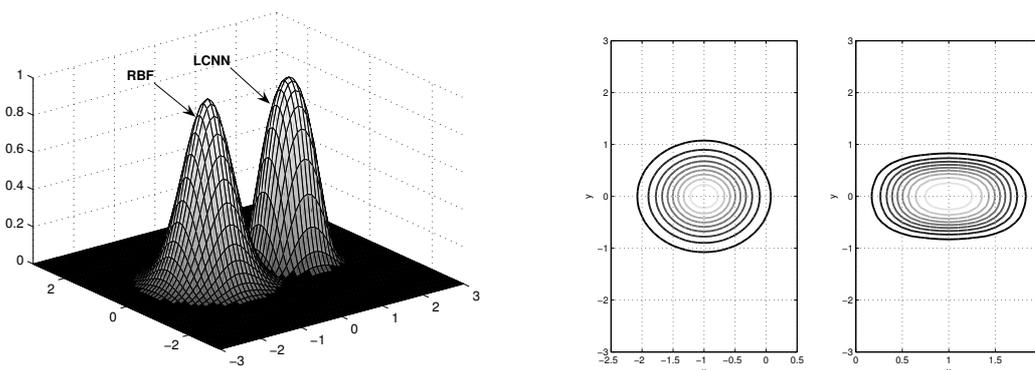
Ein verallgemeinertes RBF-Netzwerk kann durch

$$f_m(\mathbf{x}) = \sum_{i=1}^m \alpha_i \exp\left(-(\mathbf{x}-\mathbf{c}_i)^T \mathbf{C}_i^T \mathbf{C}_i (\mathbf{x}-\mathbf{c}_i)\right) \quad (2.30)$$

$$= \sum_{i=1}^m \alpha_i \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{c}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}-\mathbf{c}_i)\right) \quad (2.31)$$

ausgedrückt werden, wobei  $\boldsymbol{\Sigma}$  die Kovarianzmatrix darstellt. Ist  $\boldsymbol{\Sigma}$  eine positiv definite Matrix, so existiert ihre Inverse, und ist die Kovarianzmatrix zusätzlich noch eine Diagonalmatrix, so kann die Inverse sehr einfach durch Kehrwertbildung der Elemente der Hauptdiagonalen berechnet werden [BSM05]. Für identische Elemente der Hauptdiagonalen der Kovarianzmatrix  $\boldsymbol{\Sigma}$  ergibt sich die Form aus (2.28), wobei aufgrund der Bedingung der positiven definiten Matrix  $\boldsymbol{\Sigma}$  der Fall  $\sigma_i = 0$  ausgeschlossen wird.

<sup>11</sup> Im Folgenden wird der Ausdruck *radiale Basisfunktionen* synonym zu gaußförmige, radiale Basisfunktionen verwendet.



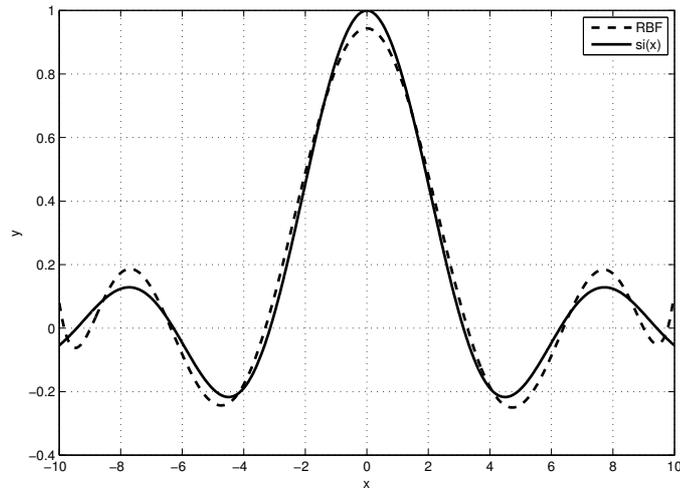
(a) lokale Funktionen durch RBF und LCNN (b) Kontur der beiden lokalen Funktionen, links RBF, rechts LCNN

**Abbildung 2.9:** Vergleich von RBF und LCNN

Aufgrund seiner Darstellung handelt es sich bei einem gaußförmigen, radialen Basisnetz ebenfalls wie das LCNN um eine lokale Approximationstechnik mit den angedeuteten Eigenschaften. Allerdings unterscheiden sich das LCNN und das RBF, obwohl beide Netze das gleiche Ziel besitzen. Die zugrunde liegende Theorie beider Netze ist gänzlich unterschiedlich. Es lassen sich keine Stabilisatoren finden, die zu der Logistikkfunktion für die Funktion  $g$  aus (2.24) führen [GJP95]. Abbildung 2.9 zeigt einen Vergleich zwischen dem LCNN und einem RBF. In Abbildung 2.9(a) sind die beiden lokalen Funktionen dargestellt, und man erkennt ähnliches Verhalten für beide Netztopologien. In Abbildung 2.9(b) zeigt sich dagegen der Unterschied zwischen beiden Netzen. Aufgrund ihrer mathematischen Struktur erreichen die RBF eine rotationsinvariante Darstellung, während ein radiales Verhalten für das LCNN nur angenähert wird, wie man an (2.20) überprüfen kann.

Bezüglich der Approximationsfähigkeiten der gaußförmigen RBF ist die Interpolation der Daten durch (2.22) für  $\lambda = 0$  sichergestellt, was allerdings keine guten Ergebnisse garantiert. In [GP90] wurde gezeigt, dass die RBF jede Funktion beliebig genau approximieren können und die Eigenschaft der Best-Approximierenden im Gegensatz zu einem MLP besitzen. Allerdings gilt diese Aussage für den Fall, dass so viele Gaußfunktionen wie Testpunkte genutzt werden, oder bei einer kleineren Anzahl von Basisfunktionen die Lage der Zentren festgelegt wird.

Der Bezug zu den biologischen Vorbildern ergibt sich aufgrund des gleichen Verhaltens von RBF-Netzen und kortikalen Neuronen [Pog90]. Zudem kann eine Gaußfunktion hoher Dimension durch Faktorisierung berechnet werden (vgl. Kapitel 4 und Abbildung 2.8). So können RBF das rezeptive Feld eines Sensorfeldes über die gewichteten Verbindungen und hoher Aktivität hinsichtlich bestimmter Merkmale abbilden. Für tiefere Erläuterungen des Zusammenhanges sei auf [PG89, Pog90] verwiesen.



**Abbildung 2.10:** Approximation der si-Funktion durch ein RBF-Netz

*Beispiel.* Das Vorgehen der gaußförmigen, radialen Basisfunktionen soll anhand der Approximation der si-Funktion verdeutlicht werden. Dafür wird ein neuronales Netz mit 10 Gaußfunktionen mit Hilfe von NETLAB [Nab02] trainiert, so dass insgesamt 30 freie Parameter durch das Modell bereitgestellt werden. Abbildung 2.10 zeigt die Antwort des RBF-Netzes. Auch hier werden durch die lokale Approximation der RBF-Netze die Nebenmaxima der Funktion korrekt dargestellt, wie bei einem LCNN. Der Approximationsfehler des Netzes liegt bei  $\Delta y = 0,1558$ , welcher größer als bei einem LCNN ist. Dies liegt an der Form des RBF-Netzes, da ein verallgemeinertes RBF-Netz aus (2.31) einem LCNN entspricht und zur Approximation ein vereinfachtes Netz nach (2.28) verwendet wurde.

### 2.2.3.2 Weitere Basisfunktionen

Neben der Gaußfunktion als Basisfunktion lassen sich durch den Stabilisator noch weitere Funktionen finden, welche u.U. besser in Hardware umsetzbar sind. Wählt man eine Produktform der Funktion  $\tilde{G}(\mathbf{s})$

$$\tilde{G}(\mathbf{s}) = \prod_{j=1}^M \tilde{g}(s_j), \quad (2.32)$$

so führt dies auf ein Funktional

$$\phi[f] = \int_{\mathbb{R}^n} d\mathbf{s} \frac{\tilde{f}(\mathbf{s})}{\prod_{j=1}^M \tilde{g}(s_j)} \quad (2.33)$$

und schließlich zu der Basisfunktion

$$g(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{c}\|_1}. \quad (2.34)$$

Man sieht an (2.34), dass bei dieser Basisfunktion anstatt der euklidischen Norm die  $L_1$ -Norm verwendet wird, was für den Fall einer Hardwareumsetzung dieses Approximationsschemas von Interesse ist. Jedoch ist dieser Stabilisator nicht immer wünschenswert, da bei einer Approximation mit der Manhattan-Norm diese nicht so glatt verläuft wie die Approximation mit dem Gaußkern [SSM98f].

Auch hier beschreibt dieses Basisfunktionsnetzwerk eine Menge von Funktionen

$$\mathcal{S} = \left\{ f_m(\mathbf{x}) \mid f_m(\mathbf{x}) = \sum_{i=1}^m \alpha_i e^{-\|\mathbf{x}-\mathbf{c}_i\|_1}, \alpha_i \in \mathbb{R}; \mathbf{c}_i \in \mathbb{R}^n; m \in \mathbb{N} \right\}. \quad (2.35)$$

Daneben existieren weitere Basisfunktionen, welche im Bereich der Approximationstheorie weniger in der Forschung neuronaler Netze untersucht wurden [Isk94]:

$$\text{Polynome} \quad \psi(x) = (-1)^{\lceil \nu/2 \rceil} \|x\|^\nu \quad \nu > 0, \nu \notin 2\mathbb{N} \quad (2.36)$$

$$\text{Thin plate Splines} \quad \psi(x) = (-1)^{k+1} \|x\|^{2k} \log \|x\| \quad k \in \mathbb{N} \quad (2.37)$$

$$\text{Mutiquadratics} \quad \psi(x) = (-1)^{\lceil \nu/2 \rceil} (c^2 + \|x\|^2)^{\nu/2} \quad \nu > 0, \nu \notin 2\mathbb{N}, c \neq 0 \quad (2.38)$$

$$\text{inverse Mutiquadratics} \quad \psi(x) = (c^2 + \|x\|^2)^{\nu/2} \quad -d < \nu < 0, c \neq 0. \quad (2.39)$$

## 2.3 Technologische Randbedingungen

Eines der am meist zitierten und diskutierten Veröffentlichungen in der Halbleiter- und Schaltungstechnik ist der Artikel von Gordon Moore [Moo65]. In diesem beschreibt er seine Beobachtung, dass die Anzahl von Komponenten pro integrierte Funktion sich jeweils innerhalb eines Jahres über die nächste Dekade verdoppeln wird. Von ihm selbst wurde diese Aussage 1975 korrigiert und der Zeitraum auf 24 Monate verlängert [Moo75]. Über die Jahre wurde durch die Halbleiterindustrie versucht, die Gültigkeit dieser Beobachtung einzuhalten [Sem05]; auch ist schon oft das Ende von dieser Beobachtung vorausgesagt worden.

Unbestritten ist jedoch die stark gestiegene Komplexität seit Moores Veröffentlichung, was die Entwickler von Systemen vor ständig wechselnde Probleme stellt. Auch sind die Strukturgrößen der Transistoren stark verkleinert worden, was neben einer höheren Integrationsdichte auch neue Herausforderungen beim Entwurf mit sich bringt. Zusammenfassend werden für den Entwurf neue Techniken benötigt. Im Folgenden soll auf den Trend der aktuellen Technologieentwicklung eingegangen

werden und mögliche Probleme sowie Grenzen der jeweiligen Prozesse aufgezeigt werden. Aufgrund der starken Anwendung und der hohen Bedeutung wird dabei besonders auf die *Complementary Metal Oxide Semiconductor* (CMOS) eingegangen.

### 2.3.1 Skalierung der CMOS-Technologie

Seit der Patentierung des grundlegenden Konzeptes des Feldeffekt-Transistors 1930 [Lil30] haben sich die geometrischen Abmessungen der Transistoren erheblich verkleinert, und die Entwicklung hat sich von diskreten zu integrierten Bauelementen vollzogen. Inspiriert durch die Beobachtungen von Moore hat die Halbleiterindustrie 1994 damit begonnen, diese Entwicklung vorherzusagen und sich Entwicklungsziele zu setzen [Sem05]. Die dabei getroffenen Aussagen betreffen die unterschiedlichen Bereiche der Fertigung, wie z.B. Lithografie, Ausbeute oder Modellierung, und reichen in der Version 2005 bis in das Jahr 2020, wobei zwischen kurzfristigen und langfristigen Entwicklungen unterschieden wird. Teilweise sind diese Prognosen nur als Entwicklungsziele zu verstehen, wobei eine Fertigung der Schaltungen und ein Betrieb beim Verfehlen dieses Ziels durchaus möglich sind. Als Konsequenz können jedoch bestimmte gewünschte Eigenschaften, wie z.B. die Ausbeute einer Halbleiterscheibe, das gewünschte dynamische Verhalten eines Transistors oder eine geringe Variation der Parameter über die Halbleiterscheibe, nicht eingehalten werden, so dass beim Entwurf auf diese Besonderheiten eingegangen werden muss.

Integrierte Schaltungen benutzen zu einem hohen Anteil die CMOS-Technologie. Dies liegt an mehreren Faktoren. So erreicht man mit dem *Metal Oxide Semiconductor Field Effect Transistor* (MOSFET) hohe Integrationsdichten und eine hohe Zuverlässigkeit. Weiter benötigen Schaltungen mit diesen Bauelementen eine geringe Leistung, und es entstehen geringere Kosten gegenüber der Bipolartechnik aufgrund geringerer Prozessschritte<sup>12</sup>. Abbildung 2.11 zeigt den stark vereinfachten Querschnitt eines n-Kanal-MOSFETs. Dieser besteht aus drei Anschlüssen: *Source*, *Drain* und *Gate*. Der Stromfluss zwischen Source und Drain wird bei einem MOSFET über die anliegende Spannung am Gate gesteuert, wobei der Entwickler auf die Eigenschaften des Transistors im Wesentlichen durch die geometrischen Abmessungen des Kanals besitzt.

Eine Miniaturisierung in der MOS-Technologie ist aufgrund der weitgehenden Unabhängigkeit der Kennlinie des Transistors von den geometrischen Abmessungen möglich. Die Idee der Skalierung der Bauelemente stammt aus [DGR<sup>+</sup>74], wobei hier eine *ideale* Skalierung vorgeschlagen wird. Alle geometrischen Abmessungen eines Transistors werden dabei um den Faktor  $\alpha_L$  reduziert. Senkt man zusätzlich noch alle Spannungen um denselben Faktor herab und erhöht alle Dotierung und

<sup>12</sup>Zumindest wurde am Anfang der Fertigung eine geringere Anzahl von Prozessschritten benötigt.

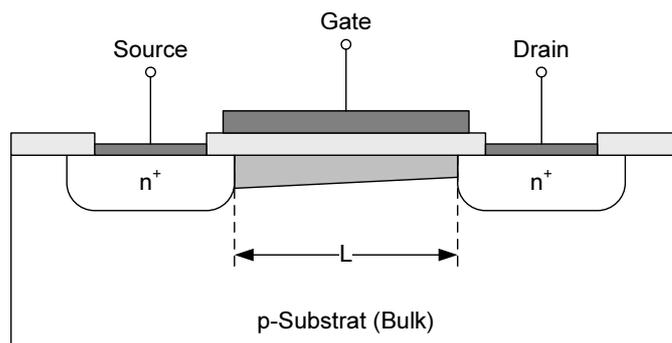


Abbildung 2.11: Querschnitt eines MOSFETs

Tabelle 2.1: Skalierungsregeln der CMOS-Technologie [FDN<sup>+</sup>01]

Physikalische Parameter	ideale Skalierung	allgemeine Skalierung	selektive Skalierung
Kanallänge	$1/\alpha_L$	$1/\alpha_L$	$1/\alpha_d$
Leitungsbreite	$1/\alpha_L$	$1/\alpha_L$	$1/\alpha_w$
elektrisches Feld	1	$\alpha_L/\alpha_U$	$\epsilon$
Spannung	$1/\alpha_L$	$1/\alpha_U$	$\epsilon/\alpha_d$
Drainstrom	$1/\alpha_L$	$1/\alpha_U$	$\epsilon/\alpha_w$
Fläche	$1/\alpha_L^2$	$1/\alpha_L^2$	$1/\alpha_w^2$
Kapazität	$1/\alpha_L$	$1/\alpha_L$	$1/\alpha_w$
Leistung	$1/\alpha_L^2$	$1/\alpha_U^2$	$\epsilon/(\alpha_w\alpha_d)$

Ladungsdichten ebenfalls um diesen Faktor, bleibt das elektrische Feld im Transistor konstant. Allerdings war diese Skalierungsform bei den Prozessen nicht zu beobachten, da die Spannungen nicht im gleichen Maße wie die Längen reduziert wurden. Dies begründet sich auf dem größeren elektrischen Feld bei höheren Spannungen, da sich so größere Ladungsträrgeschwindigkeiten<sup>13</sup> und höhere Arbeitsgeschwindigkeiten des Transistors erzielen lassen. Mittlerweile begrenzen die Leckströme die Skalierung der Spannungen.

Tabelle 2.1 zeigt die Skalierung der Geometrie und deren Auswirkungen auf die Eigenschaften des MOS-Transistors für verschiedene Skalierungsformen [FDN<sup>+</sup>01]. Für alle Skalierungsfaktoren gilt dabei  $\alpha_i > 1, \epsilon > 1$ . Werden die Spannungen und die geometrischen Abmessungen mit den gleichen Faktoren verkleinert, so erhält man ein konstantes, elektrisches Feld im Transistor. Aufgrund der angesprochenen Zusammenhänge ist die Spannung über die Technologiegenerationen nicht mit dem gleichen Faktor wie die geometrischen Dimensionen skaliert worden. Weiter ist es z.B. nicht mehr möglich, die Dicke des Gateoxids mit dem gleichen Faktor wie die

<sup>13</sup>Die höhere Geschwindigkeit hat mittlerweile ihre Sättigung gefunden.

Kanallänge und -breite zu reduzieren, da hier nur noch wenige Atomlagen<sup>14</sup> genutzt werden. Dies führt zu einer allgemeinen Skalierung, bei der die elektrische Feldstärke im Transistor ansteigt. Zusätzlich sind die geometrischen Abmessungen der Verbindungsleitungen mit einem weiteren Faktor reduziert worden, so dass sich eine dritte Form ergibt.

Allerdings skalieren die Potentialverläufe innerhalb des Transistors nicht ideal mit der vorgestellten Theorie. So müssen zur Bestimmung der minimalen Gatelänge die elektrischen Felder mit Hilfe von Simulationswerkzeugen untersucht werden. Weiter sind die Auswirkungen aus Tabelle 2.1 als Näherung zu verstehen. Auch besitzt die Skalierungstheorie ihre Grenzen, oder es können andere Skalierungsformen wünschenswert sein, da z.B. ein konstanter dynamischer Bereich oder eine konstante Anstiegsgeschwindigkeit bei geschalteten Kapazitäten erwünscht ist.

Die Grenzen der Skalierung sind vielfältig und hier wird deshalb nur ein kurzer Überblick gegeben. Mit kleineren Kanälen treten immer mehr Effekte auf, welche das Verhalten des MOS-Transistors gegenüber seiner ursprünglichen Beschreibung verfälschen. So treten vermehrt Einflüsse in der Schwellenspannung bei geringen Kanallängen auf. Zum einen kann die Schwellenspannung nicht beliebig herabgesetzt werden, da sonst hohe Leckströme im Betrieb unterhalb der Schwellenspannung auftreten. Zum anderen beeinflussen bei kleinen Dimensionen die Dotierungen von Source und Drain die Ladungen im Kanal und führen zu einer kleineren Schwellenspannung [Raz00]. Weiter wird bei hohen elektrischen Feldern zwischen dem Gate und dem Kanal die Beweglichkeit der Ladungsträger reduziert, da die freien Ladungsträger sich in einem schmalen Bereich unter dem Oxid konzentrieren und dieser Effekt auf die Beweglichkeit wirkt. Besonders bei einer Skalierungsform mit einem nicht konstanten Feld tritt dieser Effekt auf [Raz00].

Daneben kann die Skalierung der Gateoxidstärke nicht weiter vorgenommen werden, da zum einen bald eine minimale Dicke von einer Atomlage erreicht ist, und zum anderen der Tunnelstrom durch das Oxid einen immer größeren Anteil erhält, was zu einer Beschränkung führt [Tau02]. Weiter skaliert die Bandlücke von Silizium nicht, was nur durch die Veränderung des Halbleitermaterials erreicht werden kann. Das Potential verhindert ein Tunneln der Ladungsträger im Vergleich zur idealen Skalierung, so dass aufgrund des höheren elektrischen Feldes die Beweglichkeit abnimmt und die Verarmung des Kanals zunimmt. Dadurch sind kleine Schwellenspannungen schwierig zu realisieren. Eine Prognose zu den Grenzen der Skalierungsformen für bestimmte Anwendungsgebiete und eine genaue Beschreibung der Ursachen für diese Schranken finden sich in [FDN<sup>+</sup>01]. Neben diesen technisch bedingten Grenzen existieren auch Ökonomische, bei denen eine Deckung der Herstellungskosten durch den Erlös nicht mehr gewährleistet ist.

---

<sup>14</sup>Eine Dicke des Gateoxids von 3 – 5 nm entspricht ca. 6–10 Atomlagen.

Tabelle 2.2: Technologieparameter für zukünftige CMOS-Technologien [ZC06]

Gatelänge (nm)	130	90	65	45	32
$U_{\text{dd}}$ (V)	1,30	1,20	1,10	1,00	0,90
$T_{\text{ox}}$ (nm)	2,25	2,05	1,85	1,75	1,65
$U_{\text{th}}$ (V)	0,288	0,284	0,289	0,292	0,295
$I_{\text{on}}$ ( $\mu\text{A}/\mu\text{m}$ )	1020	1100	1185	1255	1300
$I_{\text{off}}$ (nA/ $\mu\text{m}$ )	30	50	70	100	150
$K$ ( $\mu\text{A}/\text{V}^2$ )	258,95	235,98	234,22	225,33	227,31

Tabelle 2.2 zeigt eine Zusammenfassung aus [ZC06] für die Prozesse verschiedener Halbleiterhersteller über mehrere Technologiegenerationen. Hier bezeichnen  $U_{\text{dd}}$  die Versorgungsspannung,  $T_{\text{ox}}$  die Dicke des Gateoxids,  $U_{\text{th}}$  die Schwellenspannung,  $I_{\text{on}}$  den Drainstrom im eingeschalteten Zustand,  $I_{\text{off}}$  den Drainstrom im ausgeschalteten Zustand (Verluststrom) und  $K$  den Transkonduktanz-Parameter. Weiter gibt es zu den Transistoren auch entsprechende Simulationsmodelle (BSIMv4), welche unter Spice zur Simulation eingesetzt werden können. Man erkennt die Abweichungen von der idealen Skalierung anhand der Werte für die Schwellenspannungen, welche im Gegensatz zur Theorie sogar ansteigen. Ein Grund dafür ist auch die stärkere Optimierung der Prozesse auf einen digitalen Entwurfsablauf. Weitere Innovationen, um die Miniaturisierung voranzutreiben, unterliegen aber einer stetigen Entwicklung [HND<sup>+</sup>06].

### 2.3.2 Alternativen zu CMOS

Da mittlerweile die konventionelle CMOS-Fertigung an ihre Skalierungsgrenzen stößt, wird verstärkt nach möglichen alternativen Technologien für integrierte Schaltungen gesucht. Für die angestrebten Ziele bei der CMOS-Technologie sind immer weniger Lösungen bekannt. Zurzeit werden im aktuellen Rahmenprogramm der Europäischen Union Projekte mit diesen Schwerpunkten (*Future and Emerging Technologies*) gefördert. Ebenfalls werden auch hier erste Prognosen in Analogie zur CMOS-Technologie gestellt [Com00]. In Anlehnung an diesen Fahrplan werden hier kurz mögliche Alternativen vorgestellt, wobei für eine detaillierte Einführung auf [GGML<sup>+</sup>97] verwiesen sei, an der sich dieser kurze Überblick orientiert.

Viele dieser neuartigen Technologien basieren auf Quanteneffekten und besitzen in ihrer Struktur als grundlegende Eigenschaft eine Insel aus einem Halbleitermaterial oder Metall<sup>15</sup>. Diese ist von Isolatoren oder einer Raumladungszone umgeben, so

<sup>15</sup> Aufgrund der Verwendung eines Halbleiters oder Metalls bleiben diese Technologien kompatibel zu der CMOS-Technologie.

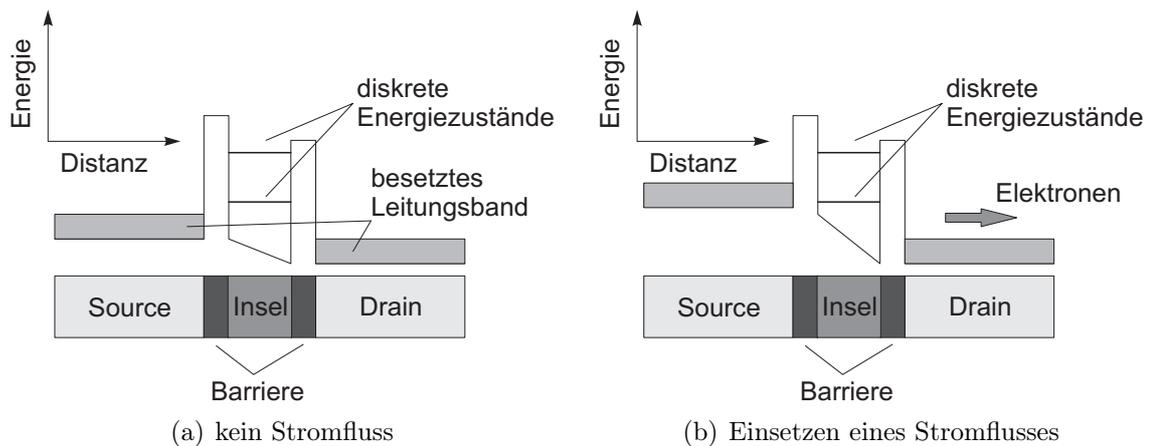


Abbildung 2.12: Funktionsweise der Resonant-Tunnelemente

dass sich Potentialbarrieren ergeben, welche den Ab- und Zufluss von Elektronen verhindern. Auf der Insel befindet sich eine bestimmte Anzahl von Elektronen, die nur diskrete Energiezustände annehmen können. Durch einen Isolator abgetrennt befinden sich an den Seiten der Insel weitere Source- und Draingebiete mit einem besetzten Leitungsband (vgl. Abbildung 2.12(a)). Elektronen können nun die Potentialbarrieren mit einer endlichen Wahrscheinlichkeit durchtunneln, wenn das Energieniveau des Leitungsbandes des Sourcegebietes mit einem (leeren) diskreten Energiezustand auf der Insel übereinstimmt. In Abbildung 2.12(a) stimmen die Energieniveaus nicht überein, und kein Strom kann fließen. Wird eine äußere Spannung angelegt, so werden die Potentiale verändert, und ein Stromfluss kann erzeugt werden (Abbildung 2.12(b)). Bei weiterem Erhöhen der Spannung stimmen die Energieniveaus nicht mehr überein, der Stromfluss sinkt wieder ab. Dadurch besitzen diese Bauelemente einen negativen, differentiellen Widerstand und mehrere Maxima bzw. Minima in der Strom-Spannungskennlinie, und sie werden aufgrund ihres Verhaltens *Resonant-Tunnelemente* genannt [GGML<sup>+</sup>97]. Mit Hilfe der vielen Maxima lässt sich im Gegensatz zur CMOS-Logik auch eine mehrwertige Logik nutzen.

Eng verwandt mit den Tunnelementen ist der *Einzelelektronentransistor* (SET), welcher sich hauptsächlich durch die Dimensionierung der Insel von den Tunnelementen abhebt. Weiter benötigt der Transistor drei Anschlüsse: Source, Gate, Drain, während die Tunnelemente auch mit zwei Anschlüssen betrieben werden können. Auf der Insel befinden sich viele bewegliche Elektronen, wobei ein Tunneln von Elektronen auf die Insel durch die Coulomb-Blockade verhindert wird. In Abhängigkeit der Spannung am Gate kann ein zusätzliches Elektron die Barriere überwinden und auf die Insel durchtunneln, welches dann durch die Barriere zum Draingebiet tunnelt und einen Stromfluss erzeugt. Wird dieser Prozess hochfrequent wiederholt, so kann ein kontinuierlicher Strom gemessen werden. Der Name *Einzelelektronentransistor* beruht auf der Funktionsweise durch das Tunneln eines einzelnen Elektrons.

Die bisher gefertigten Bauelemente werden bei sehr tiefen Temperaturen (40 K) betrieben, was sich aufgrund der Randbedingungen ergibt, dass die auf den Inseln gespeicherte Energie die thermische Energie ( $k \cdot T$ ) übersteigen muss. Bei einem Betrieb bei Raumtemperatur ergibt sich so bisher eine unzuverlässige Funktionalität. Eine weitere Beschränkung stellt die heisenbergsche Unschärferelation  $\Delta E \cdot \Delta t \geq h$  dar, wobei  $\Delta E$  die Energie darstellt, eine Information zu schreiben. Um bei der Speicherung Fehler zu vermeiden, darf ein Speicher mit diesen Elementen nicht zu nahe an der theoretischen Grenze arbeiten. So müssen die Energie und die Geschwindigkeit zum Schreiben einer Speicherzelle gleichzeitig erhöht werden. Weiter ist die Verarbeitungsgeschwindigkeit dieser Bauelemente durch die  $RC$ -Konstante beschränkt, was einen Logikentwurf auf Basis einer lokalen Architektur zum Nutzen dieses Vorteils notwendig macht.

Für die *Tunnelodiode* werden zwei Methoden für die Energiebarrieren genutzt: ein pin-Übergang oder eine große Bandlückenbarriere eingebettet in einer kleinen Bandlücke. Bei der ersten Ausführung, der Interband Tunnelodiode (ITD), tunneln die Ladungsträger am Übergang vom Valenzband in das Leitungsband aufgrund der hohen Dotierung. Die Breite der Energiebarriere hängt dabei u.A. von der Breite der undotierten Schicht, den Dotierungen und der angelegten Spannung ab. Bei der zweiten Variante, der Resonant-Tunnelodiode (RTD), besitzen diese Dioden zwei Tunnelbarrieren, welche nach dem in Abbildung 2.12 beschriebenen Prinzip arbeiten. Ein Problem dieser Bauelemente ist die Empfindlichkeit der Kennlinien gegenüber der Dicke der Tunnelbarrieren, wobei für eine hohe Zuverlässigkeit die Fertigung der Barrieren sehr genau ausgeführt werden muss. Für die Kontaktierung müssen große Flächen verwendet werden, und es fehlt im Moment aufgrund der Bauform eine mehrlagige Verbindungstechnik. Dies führt zu einer geringeren Integrationsdichte als bei konventionellen MOSFETs. Zusätzlich ist eine Integration von ITD auf Silizium aufgrund von Effekten bei der Abkühlung begrenzt.

Bei der *molekularen Nanoelektronik* werden gänzlich neue Materialien eingesetzt, welche eine identische Struktur besitzen, sehr klein und somit leicht zu fertigen sind. Dabei werden Moleküle für die elektrische Funktionalität eingesetzt, wobei als grundlegendes Konzept die Kontrolle der Transkonduktanz eines Moleküls basierend auf verschiedenen Effekten verwendet wird. So werden elektrische, mechanische, photorezeptive oder elektrochemische Effekte zur Umsetzung dieses Konzeptes genutzt. Eine Verbindung zu benachbarten Molekülen (und so der Aufbau komplexer Funktionen) wird mit Hilfe von selbstjustierenden Mechanismen vorgenommen. Schwierigkeiten bei diesem Vorgehen sind zum einen die (noch) ungenauen Modelle der elektrischen Vorgänge in den Molekülen und zum anderen eine hohe Ausbeute für die Schaltungen zu erzielen. Auch werden bisher nur geringe Integrationsdichten und hohe Schaltzeiten gegenüber der CMOS-Technologie erreicht.

Daneben existieren noch weitere Alternativen zur konventionellen CMOS-Technologie, wie z.B. die *Spin Devices*, welche auf der Veränderung des magnetischen Spins

der Elektronen aufgrund angelegter Spannungen basieren, oder der *Rapid Single Flux Quantum Logic*, welche den Josephson-Effekt bei zwei Supraleitern nutzt. Für diese weiteren Konzepte sei auf die Literatur verwiesen [Com00, GGML<sup>+</sup>97]

### 2.3.3 Störungen und Rauschen

Eine zentrale Rolle in dieser Arbeit stellen Störungen und Rauschen dar. Beide können mit Hilfe statistischer Methoden beschrieben werden. Für die Analyse der Robustheit neuronaler Netze werden die Störungen in den Eingängen und in den Parametern als Zufallsvariablen betrachtet. Die Störungen können die Eingänge, selbst Zufallsvariablen, additiv und multiplikativ überlagern. Für eine skalare Zufallsvariable  $X$  ergibt sich im additiven Fall

$$\tilde{X} = X + \mathcal{N} \quad (2.40)$$

und im multiplikativen Fall

$$\tilde{X} = \tau \cdot X. \quad (2.41)$$

Für den Fall eines Zufallsvektors ergibt sich analog für beide Arten der Überlagerung

$$\tilde{\mathbf{X}} = \mathbf{X} + \mathcal{N} = \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix} + \begin{pmatrix} \mathcal{N}_1 \\ \vdots \\ \mathcal{N}_n \end{pmatrix} \quad (2.42)$$

bzw.

$$\tilde{\mathbf{X}} = \begin{pmatrix} \tau_1 \cdot X_1 \\ \vdots \\ \tau_n \cdot X_n \end{pmatrix} = \tau \cdot \mathbf{X}, \quad (2.43)$$

wobei in (2.43) davon ausgegangen wurde, dass alle Einträge des gestörten Vektors durch die gleichartige Zufallsvariable gestört werden.

Weiter wird angenommen, dass alle Störungen voneinander und zu anderen Zufallsvariablen, wie z.B. den Eingängen, statistisch unabhängig sind, d.h., es gilt für die Wahrscheinlichkeitsdichte

$$p_{\mathcal{N}}(\boldsymbol{\eta}) = \prod_{i=1}^n p_{\mathcal{N}_i}(\eta_i) \quad (2.44)$$

und damit für die Erwartungswerte

$$E(\mathcal{N}_i \cdot \mathcal{N}_j) = E(\mathcal{N}_i) \cdot E(\mathcal{N}_j). \quad (2.45)$$

Wird nicht weiter darauf hingewiesen, so sind die Störungen normalverteilt mit Mittelwert  $\mu_i$  und Varianz<sup>16</sup>  $\phi_i^2$ . Da es sich um stochastisch unabhängige Verteilungen handelt, muss nicht auf die multivariate Gaußfunktion zurückgegriffen werden, sondern kann die skalare Dichtefunktion

$$p_{\mathcal{N}_i}(\eta_i) = \frac{1}{\sqrt{2\pi}\phi_i} e^{-\frac{(\eta_i - \mu_i)^2}{2\phi_i^2}} \quad (2.46)$$

verwendet werden.

Dabei wird für additives Rauschen

$$\mu = 0$$

und für multiplikative Störungen

$$\mu = 1$$

angenommen, so dass für die Erwartungswerte gestörter Zufallsvariablen gilt

$$E(X + \mathcal{N}) = E(X) + E(\mathcal{N}) = E(X) \quad (2.47)$$

bzw.

$$E(\tau \cdot X) = E(\tau) \cdot E(X) = E(X). \quad (2.48)$$

Somit folgt für die Differenz zweier Zufallsvariablen

$$\Delta X = \tilde{X} - X = X + \mathcal{N} - X = \mathcal{N} \quad (2.49)$$

sowie

$$\Delta X = \tilde{X} - X = \tau \cdot X - X = (\tau - 1)X, \quad (2.50)$$

wobei man analoge Ergebnisse für die Zufallsvektoren erhält.

Mit (2.49) und (2.50) folgt für das Moment zweiter Ordnung der Differenz

$$E(\Delta X^2) = E(\mathcal{N}^2) = N = \sigma_n^2, \quad (2.51)$$

wobei hier  $N$  die Leistung der Störungen und  $\sigma_n^2$  die Varianz bezeichnet sowie

$$E(\Delta X^2) = E((\tau - 1)^2 X^2) = E(X^2) (1 - 2E(\tau) + E(\tau^2)) = E(X^2) \sigma_n^2 \quad (2.52)$$

und  $E(X^2)$  die Eingangsleistung darstellt.

Das Rauschen in Kapitel 5 wird als stochastische Prozesse aufgefasst, die jeweils einen Erwartungswert von  $m_n(t) = 0$  besitzen. Bei dem Rauschen soll es sich außer-

---

<sup>16</sup> Zur besseren Übersicht wird hier anstelle von  $\sigma^2$  der griechische Buchstabe  $\phi^2$  oder  $\sigma_n^2$  verwendet, um Verwechslungen mit der Varianz der Gaußfunktionen aus den RBF-Netzen zu vermeiden.

dem um stationäre und ergodische Prozesse handeln, und die Kreuzkovarianzmatrix soll eine Diagonalform besitzen.

### 2.3.4 Fehler und Defekte

Der Fertigungsprozess zukünftiger Technologien wird im Gegensatz zu heutigen Generationen vermehrt Defekte und fehleranfällige Elemente hervorbringen [HJ03a, NSF02, Con03]. Hier können insgesamt drei verschiedene Arten von Fehlern betrachtet werden:

**permanente Fehler oder Defekte** Diese Fehlerquellen entstehen aufgrund irreversibler physikalischer Veränderungen.

**sporadische Fehler** Diese entstehen aufgrund von Veränderungen in den Betriebsbedingungen, wie z.B. durch veränderte Betriebsspannung oder Umgebungstemperatur.

**transiente Fehler** Diese Fehler resultieren aus vorübergehenden Betriebszuständen.

Der Unterschied zwischen sporadischen und transienten Fehlern ist dabei, dass sporadische Fehler wiederholt, an den gleichen Stellen und in einer großen Anzahl auftreten, während transiente Fehler aufgrund der hohen Integrationsdichte auftauchen (Interferenzen, elektrostatische Entladung, etc.). Dabei besitzen aktuelle CMOS-Technologien Fehlerraten aufgrund permanenter Fehler in der Größenordnung von  $10^{-7} \text{ h}^{-1}$  [Sem05].

Aufgrund der kleineren Strukturgröße in zukünftigen Technologieprozessen wird die Anfälligkeit der Systeme gegenüber sporadischen und transienten Fehlern ansteigen [Con03]. Die Anzahl der zufälligen Defekte pro Fläche bleibt dabei annähernd nach dem Technologiefahrplan [Sem05] konstant, um eine definierte Ausbeute der Halbleiterscheibe sicherzustellen.

Verschiedene Maße definieren die Eigenschaften einer Schaltung oder der Bauelemente, wie z.B. die mittlere Zeitspanne zwischen dem Auftauchen von Fehlern, die mittlere Zeit bis zu dem Entstehen eines Fehlers oder die Anzahl der Fehler innerhalb einer Zeitspanne [Por02]. Für die späteren Analysen wird angenommen, dass die jeweiligen Ausfallraten sowie die Dauer bis zum Auftreten eines Fehlers oder Defektes für die einzelnen Elemente bekannt sind.

Die Auswirkungen der einzelnen Fehlertypen können je nach Ort ihres Auftretens oder nach Art ihres Ursprunges unterschiedlich sein. So können etwa Verunreinigungen (mikroskopische, molekulare, alkalische oder atomare) während der Herstellung in Abhängigkeit ihres Auftretens und ihrer Art zu Abschattungseffekten, zu

Strukturveränderungen, zu Veränderungen in der Zeitstabilität oder zu einer Beeinflussung der Bauelementeparameter (Schwellenspannung etc.) führen [Hil04]. Dies wirkt sich negativ auf die Ausbeute an funktionstüchtigen Elementen aus.

Für die weitere Analyse wird angenommen, dass sich die unterschiedlichen Fehler auf das Verhalten der Schaltung auswirken. Um diese Effekte auf die Bauelemente zu erfassen, müssen die Fehler auf ein logisches<sup>17</sup> Fehlermodell abgebildet werden [Por02]. Als Fehlermodell wird dabei das in der Praxis bewährte *Stuck-At*-Fehlermodell angewendet [Smi97]. Dies bedeutet, dass sich alle Fehler auf einen logischen Fehler zurückführen lassen, so dass verschiedene Verbindungen, Eingänge oder Ausgänge eines Elementes auf einem logischen Wert ('0', SA0 oder '1', SA1) festsitzen. Da auch analoge Schaltungen mit kontinuierlichen Werten betrachtet werden, wird weiter angenommen, dass Ein- und Ausgänge auf einem beliebigen Wert  $\mu$  innerhalb eines Intervalls festsitzen können. Auch wird vorausgesetzt, dass nur ein einzelner Fehler oder Defekt auftaucht.

Die Auswirkungen von Fehlern und Defekten auf Parameter oder Eigenschaften, wie z.B. Verzögerungs- oder Laufzeiten, welche die Leistungsfähigkeit des Systems herabsetzen, wird in dieser Arbeit nicht näher betrachtet. Eine weitergehende Untersuchung dieser Aspekte ist im Hinblick auf den Einsatz in der Nanotechnologie und analogen Entwürfen sinnvoll.

**Bemerkung.** Neben dem *Stuck-At*-Fehlermodell existieren noch weitere Modelle in der Literatur, die Defekte und Fehler genauer modellieren können (*Stuck-On*, *Stuck-Off*, *Stuck-Open*) [Smi97]. Da die Analyse in dieser Arbeit möglichst unabhängig von der verwendeten Architektur geschehen soll, wurde ein Fehlermodell auf einer hohen Abstraktionsebene gewählt. Während sich bei der CMOS-Technologie gezeigt hat, dass ein Fehlermodell auf Logikebene ausreicht, um defekte Bauelemente während der Produktion zu erkennen [Smi97], so muss dies für die Nanotechnologie insbesondere beim Einsatz fremder Materialien verstärkt nachgewiesen werden.

Bei der Untersuchung in Kapitel 3 wird das Fehlermodell so verwendet, dass ein Parameter des neuronalen Netzes auf einem Wert festsetzt. Bei einer analogen Umsetzung ist diese Annahme gerechtfertigt, während bei der digitalen Implementierung nicht unbedingt alle zur Darstellung verwendeten binären Stellen durch einen Fehler betroffen sein müssen. Allerdings ist diese Annahme wiederum bei kleineren Strukturen der Nanotechnologie gerechtfertigt, da z.B. ein Partikel in der Größe eines Rechelementes dieses zu einem Ausfall bringen kann [Com00].

Bei einer binären Zahlendarstellung kann jede Stelle von einem *Stuck-At*-Fehler betroffen sein. Dieser Umstand kann als eine Zufallsvariable modelliert werden, so dass sich eine Störung, die mit Mittelwert und Varianz beschrieben werden kann, ergibt.

---

<sup>17</sup>Der Begriff wird hier als Element der booleschen Algebra verwendet und nicht im Sinne der Pragmatik.

# Analyse von Fehlertoleranz und Robustheit

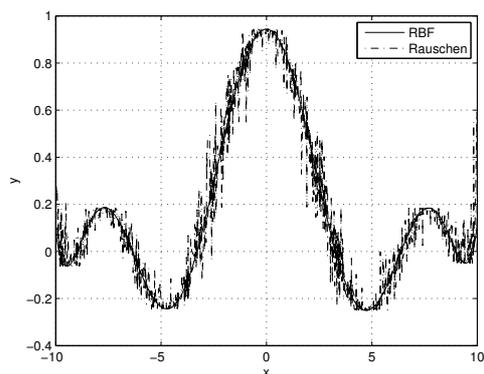
Die Begriffe Robustheit und Fehlertoleranz implizieren ein allgemeines Verständnis für den Sachverhalt. Konkrete Aussagen aber werden durch beide Ausdrücke vermieden. Dazu soll in diesem Kapitel eine mathematische Formulierung für die neuronalen Netze erstellt werden, welche qualitative Aussagen über die Eigenschaften trifft. So erlauben dann diese Analysen einen Vergleich von Netzen mit unterschiedlichen Topologien und lassen sich zusätzlich zur Verbesserung der Systemeigenschaften heranziehen.

## 3.1 Robustheit neuronaler Netze

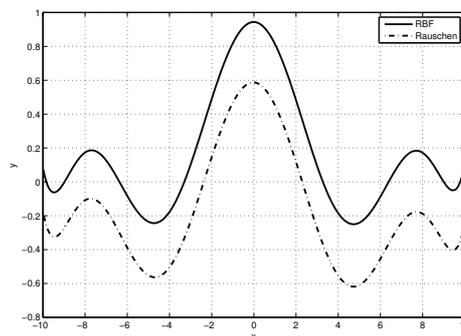
Die Robustheit neuronaler Netze bezieht sich auf das Verhalten des Systems gegenüber Parameterschwankungen und Rauschen in den Eingängen. Bedingt durch die technische Umsetzung der neuronalen Netze in Hard- oder Software ergeben sich Abweichungen von den trainierten Parametern. Die Ausgabe des Netzes soll unter diesen Umständen nur gering von der ursprünglichen Ausgabe abweichen.

Aber auch ohne Schwankungen von Parameter in den technischen Realisierungen muss ein neuronales Netz mit Rauschen in den Eingängen umgehen können. Nur selten sind für die Rekonstruktion der Funktion Testdaten ohne überlagertes Rauschen vorhanden, so dass auch hier eine definierte Ausgangsfunktion erreicht werden muss.

Das Ziel der Eigenschaft Robustheit ist es, eine stabile Approximation zu erreichen. Stabil bedeutet dabei, dass zwei sich leicht unterscheidende Eingänge nur kleine Abweichungen in der Ausgabe erzeugen. Dadurch kann eine definierte Ausgabe bei Schwankungen in den Parametern oder Eingängen garantiert werden. Diese Eigenschaft einer stabilen Approximation bei zwei sich gering unterscheidenden Eingängen wird mathematisch durch den Begriff der gleichgradigen Stetigkeit erfasst. Dabei ergibt sich mit der Definition der Stetigkeit [Heu03, Sim64, Rud76]:



(a) Rauschen in den Eingängen



(b) Rauschen in den Ausgangsgewichten

**Abbildung 3.1:** Auswirkungen des Rauschens auf die Approximation eines RBF-Netzes

**Definition 3** (Stetigkeit). *Die auf  $X$  definierte Funktion  $f$  heißt stetig in  $\xi$ , wenn zu jedem  $\epsilon > 0$  ein  $\delta = \delta(\epsilon)$  existiert, so dass für alle  $x \in X$  mit  $|x - \xi| < \delta$  folgt  $|f(x) - f(\xi)| < \epsilon$ .*

ebenfalls die Definition der gleichgradigen Stetigkeit einer Funktionsmenge [Ste04b]

**Definition 4** (gleichgradige Stetigkeit). *Sei  $E \subseteq \mathbb{R}^n, E \neq \emptyset$ .  $\mathcal{S}$  sei eine Menge (Familie) von reellwertigen Funktionen  $f : E \rightarrow \mathbb{R}^m$ .  $\mathcal{S}$  heißt gleichgradig stetig, wenn es zu jedem  $\epsilon > 0$  ein  $\delta > 0$  gibt, so dass für alle  $x, y \in E$  mit  $\|x - y\| < \delta$  folgt  $\|f(x) - f(y)\| < \epsilon$  für alle  $f \in \mathcal{S}$ .*

In den folgenden Kapiteln werden die verschiedenen Typen neuronaler Netze hinsichtlich der Eigenschaft der gleichgradigen Stetigkeit untersucht.

**Beispiel.** Zur Veranschaulichung der Auswirkungen des Rauschens auf die Approximationseigenschaften eines RBF-Netzes wird das Netz aus Abbildung 2.10 mit 10 Neuronen verwendet. Der Eingang  $x$  wird nun mit weißem Gaußrauschen, Varianz  $\sigma_n^2 = 0,1$  und Erwartungswert  $\mu = 0$ , überlagert, wobei für die Parameter des Netzes keine Störungen angenommen werden. In dem anderen Szenarium wurden die Eingänge als rauschfrei betrachtet und die Ausgangsgewichte mit dem gleichen Rauschen gestört. Abbildung 3.1 zeigt die Auswirkungen des Rauschens auf die Approximationseigenschaften des Netzes. Der Ausgabewert des RBF-Netzes in Abbildung 3.1(a) zeigt eine deutlich verrauschte si-Funktion. Der Approximationsfehler steigt auf  $\Delta y = 0,6448$  gegenüber  $\Delta y = 0,1558$  bei rauschfreien Eingängen. Die gestörten Ausgangsgewichte führen auf einen Fehler von  $\Delta y = 0,4257$ . Das Ziel eines robusten Netzes ist dabei, den Einfluss dieses Rauschens durch eine obere Schranke zu begrenzen, so dass bestimmte Auflösungen garantiert werden können.

### 3.1.1 Basisfunktionen

In diesem Abschnitt werden die Basisfunktionen aus Abschnitt 2.2.3 auf ihre gleichgradige Stetigkeit hin untersucht. Die Analysen konzentrieren sich dabei auf die gaußförmigen, radialen Basisfunktionen und die Netze mit Produkt-Stabilisatoren.

#### 3.1.1.1 Radiale Basisfunktionen

Die radialen Basisfunktionen mit Gaußfunktion lassen sich nach (2.28) als eine Menge von Funktionen darstellen. Zuerst wird die gleichgradige Stetigkeit des Basisfunktionsnetzwerkes bzgl. seines Eingangsraumes  $\mathbb{R}^n$  untersucht.

Für die Differenz im Ausgang für zwei unterschiedliche Eingangsvektoren  $\mathbf{x}$  und  $\mathbf{y}$  folgt:

$$|f_m(\mathbf{x}) - f_m(\mathbf{y})| = \left| \sum_{i=1}^m \alpha_i \exp\left(\frac{-\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) - \sum_{i=1}^m \alpha_i \exp\left(\frac{-\|\mathbf{y} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) \right|. \quad (3.1)$$

Da (3.1) stetig differenzierbar innerhalb des gesamten Eingangsraumes ist, kann auf diese Gleichung der Mittelwertsatz [Heu03] angewendet werden, womit folgt

$$|f_m(\mathbf{x}) - f_m(\mathbf{y})| = |\nabla_x f_m(\boldsymbol{\xi})(\mathbf{x} - \mathbf{y})|, \quad (3.2)$$

wobei  $\boldsymbol{\xi} \in \mathbb{R}^n$  einen geeigneten Punkt auf der Verbindungsstrecke von  $\mathbf{x}$  nach  $\mathbf{y}$  mit

$$\boldsymbol{\xi} = \mathbf{x} + \lambda(\mathbf{y} - \mathbf{x}) \quad \text{mit} \quad 0 \leq \lambda \leq 1 \quad (3.3)$$

darstellt, und  $\nabla_x$  den Gradienten bzgl. der Eingänge  $x$  bezeichnet.

Mit Hilfe der Cauchy-Schwarzschen Ungleichung sowie der Äquivalenz der Normen [Heu03]

$$\|\cdot\|_\infty \leq \|\cdot\|_2 \leq \|\cdot\|_1 \quad (3.4)$$

kann (3.2) weiter vereinfacht werden

$$|\nabla_x f_m(\boldsymbol{\xi})(\mathbf{x} - \mathbf{y})| \leq \|\nabla_x f_m(\boldsymbol{\xi})\|_2 \|\mathbf{x} - \mathbf{y}\|_2 \leq \|\nabla_x f_m(\boldsymbol{\xi})\|_1 \|\mathbf{x} - \mathbf{y}\|_2. \quad (3.5)$$

Somit folgt mit zur Hilfenahme der Dreiecksungleichung

$$|f_m(\mathbf{x}) - f_m(\mathbf{y})| = |\nabla_x f_m(\boldsymbol{\xi}) \cdot (\mathbf{x} - \mathbf{y})| \quad (3.6)$$

$$\begin{aligned} &\leq \sum_{k=1}^n \left| \sum_{i=1}^m \alpha_i \exp\left(-\frac{\|\boldsymbol{\xi} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) \cdot \frac{(-2)(\xi_k - c_{ik})}{2\sigma_i^2} \right| d(\mathbf{x}, \mathbf{y}) \\ &\leq \sum_{k=1}^n \sum_{i=1}^m \left| \alpha_i \exp\left(-\frac{\|\boldsymbol{\xi} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) \cdot \frac{-(\xi_k - c_{ik})}{\sigma_i^2} \right| d(\mathbf{x}, \mathbf{y}) \quad (3.7) \end{aligned}$$

$$= \sum_{k=1}^n \sum_{i=1}^m |\alpha_i| \left| \exp\left(-\frac{\|\boldsymbol{\xi} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) \cdot \frac{-(\xi_k - c_{ik})}{\sigma_i^2} \right| d(\mathbf{x}, \mathbf{y}). \quad (3.8)$$

Für weitere Abschätzungen von (3.8) wird das Maximum der ersten Ableitung der Gaußfunktion bestimmt. Dabei gilt für die Grenzwerte  $|x_k| \rightarrow \infty$

$$\lim_{|x_k| \rightarrow \infty} \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) \cdot \frac{x_k - c_{ik}}{\sigma_i^2} = 0. \quad (3.9)$$

Damit folgt für die Extremwerte der ersten Ableitung

$$\frac{\partial \left( e^{-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}} \frac{(x_k - c_{ik})}{\sigma_i^2} \right)}{\partial x_k} = -\exp(\cdot) \frac{(x_k - c_{ik})^2}{\sigma_i^4} + \exp(\cdot) \frac{1}{\sigma_i^2} = 0 \quad (3.10)$$

$$\Rightarrow (x_k - c_{ik})^2 = \sigma_i^2 \quad (3.11)$$

$$x_k = \pm\sigma_i + c_{ik} \quad (3.12)$$

und führt zu dem Maximum der ersten Ableitung von

$$\left| \exp(\cdot) \frac{\pm\sigma_i + c_{ik} - c_{ik}}{\sigma_i^2} \right| = \left| \exp(\cdot) \frac{\pm\sigma_i}{\sigma_i^2} \right| = \underbrace{|\exp(\cdot)|}_{\leq 1} \left| \frac{\pm\sigma_i}{\sigma_i^2} \right| \quad (3.13)$$

$$\leq \frac{1}{\sigma_i}. \quad (3.14)$$

Mit Hilfe von (3.13) kann (3.8) weiter abgeschätzt werden, und es gilt

$$\|\nabla_x f(\boldsymbol{\xi})\|_1 d(\mathbf{x}, \mathbf{y}) \leq \sum_{k=1}^n \sum_{i=1}^m |\alpha_i| \left| \frac{1}{\sigma_i} \right| d(\mathbf{x}, \mathbf{y}) \quad (3.15)$$

$$= n \cdot \sum_{i=1}^m \left| \frac{\alpha_i}{\sigma_i} \right| d(\mathbf{x}, \mathbf{y}) = \epsilon. \quad (3.16)$$

Mit den Ergebnissen aus (3.16) kann  $\delta$  zu

$$\delta = \frac{\epsilon}{n \cdot \sum_{i=1}^m \left| \frac{\alpha_i}{\sigma_i} \right|} \quad (3.17)$$

gewählt werden.

Aus (3.17) kann gefolgert werden, dass  $\epsilon$  und  $\delta$  nicht unabhängig von der Funktion gewählt werden können und als Konsequenz ein radiales Basisfunktionsnetzwerk nicht die Eigenschaft der gleichgradigen Stetigkeit im Eingangsraum besitzt.

Damit (3.16) für alle Funktionen  $f_m(\mathbf{x})$  aus (2.28) in Einklang mit Definition 4 steht, müssen  $\epsilon$  und  $\delta$  unabhängig von allen Funktionen gewählt werden können. Nach (3.17) jedoch sind beide Parameter von den Gewichten  $\alpha_i$  bzw. den Varianzen  $\sigma_i$  abhängig. Diese wiederum hängen von der zu approximierenden Funktion ab, so dass  $\epsilon$  und  $\delta$  von der Funktion abhängen. Damit besitzt ein RBF nicht die Eigenschaft der gleichgradigen Stetigkeit im Eingangsraum.

Um jedoch für ein RBF-Netzwerk die gleichgradige Stetigkeit im Eingangsraum zu erreichen, müssen die Gewichte und Varianzen einige Randbedingungen erfüllen. Damit  $\epsilon$  und  $\delta$  in (3.17) unabhängig von der Funktion gewählt werden können, muss Folgendes erfüllt sein

$$\begin{aligned} \Rightarrow \quad & \sum_{i=1}^m \left| \frac{\alpha_i}{\sigma_i} \right| \text{ muss begrenzt sein} \\ \Rightarrow \quad & |\alpha_i| \leq B \wedge \left| \frac{1}{\sigma_i} \right| \leq T \end{aligned} \quad (3.18)$$

$$\Rightarrow \quad \sum_{i=1}^m \left| \frac{\alpha_i}{\sigma_i} \right| \leq \sum_{i=1}^m B \cdot T = m \cdot B \cdot T, \quad (3.19)$$

wobei  $B$  die obere Grenze der Ausgangsgewichte und  $1/T$  die untere Grenze der Varianzen bezeichnen.

Mit den Randbedingungen der Parameter aus (3.18) erhält das RBF die gleichgradige Stetigkeit, und es folgt

$$|f_m(\mathbf{x}) - f_m(\mathbf{y})| \leq \|\nabla_x f(\xi)\|_1 d(\mathbf{x}, \mathbf{y}) \quad (3.20)$$

$$\leq n \cdot \sum_{i=1}^m |\alpha_i| \left| \frac{1}{\sigma_i} \right| d(\mathbf{x}, \mathbf{y}) \quad (3.21)$$

$$\leq n \cdot \sum_{i=1}^m B \cdot T = n \cdot m \cdot B \cdot T = \epsilon \quad (3.22)$$

und

$$\delta = \frac{\epsilon}{n \cdot m \cdot B \cdot T}. \quad (3.23)$$

Damit können beide Parameter unabhängig von der Funktion gewählt werden, und die RBF sind gleichgradig stetig.

Für Rauschen in den Eingängen kann der sich einstellende mittlere quadratische Fehler berechnet werden. Dafür wird angenommen, dass es sich bei dem Rauschen um eine Gaußverteilung mit dem Mittelwert  $E(X) = 0$  und endlicher Varianz  $\sigma_n^2$  handelt.

Weiter seien alle Eingänge durch gleichartige Störungen überlagert, wobei diese Störungen voneinander stochastisch unabhängig sind, d. h.

$$E(X_i \cdot X_j) = 0 \quad \forall i \neq j. \quad (3.24)$$

Somit können insgesamt  $n$  Eingänge zu dem mittleren quadratischen Fehler beitragen, und es ergibt sich

$$\text{mse} \leq n^2 \left( \sum_{i=1}^m \left| \frac{\alpha_i}{\sigma_i} \right| \right)^2 n \cdot \sigma_n^2. \quad (3.25)$$

Aus (3.25) ergibt sich, dass der mittlere quadratische Fehler nicht nach oben beschränkt ist, da die Gewichte und Varianzen keine obere bzw. untere Grenze besitzen. Mit den Randbedingungen aus (3.18) kann eine obere Grenze für den Fehler angegeben werden

$$\text{mse} \leq n^3 m^2 B^2 T^2 \sigma_n^2. \quad (3.26)$$

**Bemerkung.** Es sei hier darauf hingewiesen, dass der mittlere quadratische Fehler aus (3.26) nur aufgrund des Rauschens zustande kommt. Zusätzlich muss der mittlere quadratische Fehler aufgrund der Approximation durch das Modell noch einbezogen werden, um den gesamten mittleren quadratischen Fehler zu erhalten. Ähnliche Aussagen gelten auch bei den weiteren Abschätzungen.

Ähnliche Überlegungen können auch für den Gewichtsraum angestellt werden. Dafür werde angenommen, dass  $\mathcal{W}_1$  und  $\mathcal{W}_2$  zwei Punkte innerhalb dieses Raumes darstellen und  $d(\mathcal{W}_1, \mathcal{W}_2)$  eine Metrik bezeichnet. Damit ergibt sich in ähnlicher Weise für den Gewichtsraum, wenn sich zwei Parametervektoren leicht unterscheiden

$$|f_m(\mathcal{W}_1) - f_m(\mathcal{W}_2)| = \left| \sum_{i=1}^m \alpha_i^1 \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i^1\|^2}{2\sigma_{i1}^2}\right) - \sum_{i=1}^m \alpha_i^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i^2\|^2}{2\sigma_{i2}^2}\right) \right|. \quad (3.27)$$

Mit Hilfe des Mittelwertsatzes kann (3.27) weiter abgeschätzt werden

$$|f_m(\mathcal{W}_1) - f_m(\mathcal{W}_2)| \leq (\|\nabla_{\alpha} f(\xi)\|_1 + \|\nabla_{\mathbf{c}_i} f(\xi)\|_1 + \|\nabla_{\sigma} f(\xi)\|_1) d(\mathcal{W}_1, \mathcal{W}_2), \quad (3.28)$$

wobei  $\nabla_\alpha$ ,  $\nabla_{\mathbf{c}_i}$  und  $\nabla_\sigma$  die Gradienten in Bezug auf den entsprechenden Parameter bezeichnen.

Die einzelnen Gradienten können durch

$$\|\nabla_\alpha f(\xi)\|_1 = \sum_{i=1}^m \left| \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) \right| \leq \sum_{i=1}^m |1| = m \quad (3.29)$$

$$\|\nabla_{\mathbf{c}_i} f(\xi)\|_1 = \sum_{k=1}^n \left| \sum_{i=1}^m \alpha_i \exp\left(-\frac{\|\mathbf{x} - \xi_i\|^2}{2\sigma_i^2}\right) \frac{x_k - \xi_{ik}}{\sigma_i^2} \right| \quad (3.30)$$

$$\leq \sum_{k=1}^n \sum_{i=1}^m \left| \alpha_i \exp\left(-\frac{\|\mathbf{x} - \xi_i\|^2}{2\sigma_i^2}\right) \frac{x_k - \xi_{ik}}{\sigma_i^2} \right| \quad (3.31)$$

$$\leq \sum_{k=1}^n \sum_{i=1}^m \left| \frac{\alpha_i}{\sigma_i} \right| = n \cdot \sum_{i=1}^m \left| \frac{\alpha_i}{\sigma_i} \right| \quad (3.32)$$

$$\begin{aligned} \|\nabla_\sigma f(\xi)\|_1 &= \sum_{i=1}^m \left| \alpha_i \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\xi^2}\right) \frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\xi^3} \right| \\ &\leq \sum_{i=1}^m |\alpha_i| \left| \frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\xi^3} \right| \end{aligned} \quad (3.33)$$

abgeschätzt werden.

In (3.32) kann in gleicher Weise wie beim Eingangsraum die Summe mit dem Maximum der ersten Ableitung und der Dreiecksungleichung abgeschätzt werden (vgl. Gleichung (3.14)). Die Abschätzung in (3.33) erhält man mit dem Maximum der Gaußfunktion

$$h(z) = \exp\left(\frac{-z^2}{2\sigma^2}\right) \leq 1. \quad (3.34)$$

Damit ergibt sich für den Gewichtsraum

$$\begin{aligned} \|\nabla_{\mathcal{W}} f(\xi)\|_1 d(\mathcal{W}_1, \mathcal{W}_2) &\leq \left( m + n \cdot \sum_{i=1}^m \left| \frac{\alpha_i}{\sigma_i} \right| + \sum_{i=1}^m \left| \alpha_i \frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\xi^3} \right| \right) d(\mathcal{W}_1, \mathcal{W}_2) \\ &= \epsilon \\ \Rightarrow \delta &= \frac{\epsilon}{m + n \cdot \sum_{i=1}^m \left| \frac{\alpha_i}{\sigma_i} \right| + \sum_{i=1}^m \left| \alpha_i \frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\xi^3} \right|}. \end{aligned} \quad (3.35)$$

(3.35) zeigt ähnlich wie schon beim Eingangsraum die Abhängigkeit der Parameter  $\epsilon$  und  $\delta$  von den Gewichten  $\alpha_i$ , den Varianzen  $\sigma_i$  und der euklidischen Distanz zwischen dem Eingangsvektor und den Zentren  $\|\mathbf{x} - \mathbf{c}_i\|$ . Diese Abhängigkeit besteht somit auch zu der approximierenden Funktion, so dass die Parameter  $\epsilon$  und  $\delta$  nicht unabhängig von der Funktion gewählt werden können. Das RBF-Netz besitzt somit ebenfalls nicht die Eigenschaft der gleichgradigen Stetigkeit im Gewichtsraum.

Die gleichgradige Stetigkeit kann jedoch erreicht werden, wenn man zusätzlich zu den Randbedingungen der Gewichte und Varianzen aus (3.18) die euklidische Distanz zwischen Eingang und den Zentren durch eine obere Schranke begrenzt

$$\|\mathbf{x} - \mathbf{c}_i\| \leq C. \quad (3.36)$$

**Bemerkung.** Durch das Beschränken der euklidischen Distanz zwischen dem Eingangsvektor  $\mathbf{x}$  und den Zentren  $\mathbf{c}_i$  des RBF-Netzes müssen die Eingangsvektoren innerhalb einer (Hyper-)Sphäre liegen. Durch dieses Begrenzen des Radius wird somit auch die Vapnik-Chervonenkis-Dimension (VC-Dimension) [Vap98] der Hyperebenen mit Marge  $\Delta$  zu den Testdaten bei einer Klassifizierung begrenzt. Dadurch wird ebenfalls die Wahrscheinlichkeit für eine falsche Klassifikation von Testdaten nach oben beschränkt und hängt somit nicht mehr von der Dimension des transformierten Raums ab (Kernel-Trick) [Vap98, Vap95b]. Es gilt hierbei die VC-Dimension  $h$

$$h \leq \min \left\{ \frac{R^2}{\Delta^2}, n \right\} + 1, \quad (3.37)$$

wobei  $R$  den Radius,  $\Delta$  die Marge und  $n$  die Eingangsdimension bezeichnet. Somit ist für große Margen und kleine Radien die VC-Dimension viel kleiner als die Anzahl der freien Parameter. Dies ist für die Stützvektormaschine (Support Vector Machine) hilfreich [SS02], da dort der Eingangsvektor in einen hochdimensionalen Raum transformiert werden kann, ohne dass die VC-Dimension und damit die Anzahl freier Parameter ansteigt. Somit können gute Generalisierungseigenschaften sichergestellt werden, und man erhält eine Kapazitätskontrolle. Die Randbedingung in (3.36) steht damit in enger Beziehung zu den Ergebnissen der statistischen Lerntheorie.

Für die Abschätzung aus (3.35) ergibt sich somit

$$\sum_{i=1}^m \left| \frac{\alpha_i}{\sigma_i} \right| \leq \sum_{i=1}^m B \cdot T = mBT \quad (3.38)$$

$$\sum_{i=1}^m \left| \alpha_i \frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\xi^3} \right| \leq \sum_{i=1}^m B \cdot C^2 \cdot T^3 = mBC^2T^3 \quad (3.39)$$

und damit

$$\begin{aligned} |f_m(\mathcal{W}_1) - f_m(\mathcal{W}_2)| &\leq \|\nabla_{\mathcal{W}} f(\xi)\|_1 d(\mathcal{W}_1, \mathcal{W}_2) \\ &\leq (m + n \cdot m \cdot B \cdot T + m \cdot B \cdot C^2 \cdot T^3) d(\mathcal{W}_1, \mathcal{W}_2) \\ &= \epsilon. \end{aligned} \quad (3.40)$$

Der Parameter  $\delta$  kann dann zu

$$\delta = \frac{\epsilon}{m + nmBT + mBC^2T^3} \quad (3.41)$$

gesetzt werden. Somit können die Parameter  $\epsilon$  und  $\delta$  unabhängig von der Funktion gewählt werden, und es ergibt sich mit den Randbedingungen aus (3.18) und (3.36) die gleichgradige Stetigkeit des RBF-Netzes im Gewichtsraum.

Weiter sei nun angenommen, dass die Parameter durch Rauschen überlagert werden. Auch hier wird gaußsches Rauschen mit Mittelwert  $E(X) = 0$  und endlicher Varianz  $\sigma_n^2$  angenommen, wobei alle Rauschprozesse voneinander stochastisch unabhängig sind. Wird jedes Gewicht, jedes Zentrum und jede Varianz des Netzwerks von Rauschen überlagert, so tragen insgesamt  $m + m \cdot n + m = m(n + 2)$  Parameter zu dem Rauschen im Ausgang bei. Der mittlere quadratische Fehler kann mit (3.35) abgeschätzt werden zu

$$\text{mse} \leq \left( m + n \cdot \sum_{i=1}^m \left| \frac{\alpha_i}{\sigma_i} \right| + \sum_{i=1}^m \left| \alpha_i \frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\xi^3} \right| \right)^2 m(n + 2)\sigma_n^2. \quad (3.42)$$

Man erkennt, dass ohne Restriktionen der Gewichte dieser Fehler beliebig groß werden kann und eine garantierte Ausgabe nicht mehr gewährleistet werden kann. Um nun eine zuverlässige Ausgabe zu erhalten, führen die eingeführten Randbedingungen auf eine obere Schranke des mittleren quadratischen Fehlers

$$\text{mse} \leq (m + nmBT + mBC^2T^3)^2 m(n + 2)\sigma_n^2. \quad (3.43)$$

Der mittlere quadratische Fehler ist sowohl im Eingangsraum (3.26) als auch im Ausgangsraum (3.43) von der Varianz des Rausches abhängig. Diese wird maßgeblich von der jeweiligen Realisierung bestimmt, ob es sich um eine analoge oder digitale Realisierung handelt. Auf den Einfluss dieser Realisierungsform wird näher in Kapitel 5 eingegangen.

Mit den Abschätzungen des mittleren quadratischen Fehlers in (3.16) oder (3.35) kann die garantierte Auflösung bestimmt werden, die durch das neuronale Netz im Ausgang erreicht werden kann. Aufgrund der Abschätzung der oberen Grenze für die Rauschleistung im Ausgang kann mit diesem Zusammenhang nur die Auflösung für den schlechtesten Fall bestimmt werden. Hierfür muss die Leistung des Ausgangssignals oder des Eingangssignals bekannt sein. Ist nur die Eingangsleistung bekannt, so kann mit einer analogen Abschätzung die Leistung im Ausgang abgeschätzt werden. Mit Hilfe der Informationstheorie [Sha48, Sar98] kann eine garantierte Auflösung des Netzes bestimmt werden.

**Beispiel.** Für ein RBF-Netz mit  $m = 20$  Neuronen und insgesamt  $n = 4$  Eingängen ergibt sich bei einer Beschränkung der Parameter  $B = 5, T = 10$ , einer

Rauschleistung von  $N = 0,001$  und einer Signalleistung von  $P_s = 1000$  im Ausgang im schlechtesten Fall eine Auflösung von

$$k \approx \frac{1}{2} \text{ld} \left( 1 + \frac{P_s}{N} \right) \geq 0,0112 \text{ bit.}$$

Diese Abschätzungen für den schlechtesten Fall können zur Klassifizierung der RBF-Netze hinsichtlich ihrer Rauscheigenschaften verwendet werden. So kann für das hier verwendete RBF-Netz durch eine Reduktion der Eingänge von vier auf drei (z.B. durch die Hauptkomponentenanalyse [Pea01b, Hay99]) eine Auflösung von 0,0262 bit garantiert werden. So zeigt sich auch, dass sich eine Verringerung der Eingangsdimension positiver auswirkt als eine Verkleinerung des Netzes von 20 auf 15 Neuronen. Hier ergibt sich eine garantierte Auflösung von 0,0198 bit.

### 3.1.1.2 Basisfunktionen mit Produkt-Stabilisatoren

Ebenso wie bei den radialen Basisfunktionen kann auch bei den Basisfunktionsnetzwerken mit Produkt-Stabilisatoren nach der Robustheit dieser Basisfunktionen gegenüber Rauschen in den Eingängen und Gewichten gefragt werden. Die Übertragungsfunktion eines solchen Netzwerkes ergibt sich aus Abschnitt 2.2.3.2 zu

$$y = f_m(\mathbf{x}) = \sum_{i=1}^m \alpha_i e^{-\|\mathbf{x}-\mathbf{c}_i\|_1}. \quad (3.44)$$

Im Gegensatz zu den radialen Basisfunktionen mit Gaußfunktionen kann (3.44) nicht mittels Mittelwertsatz analysiert werden, da die Übertragungsfunktion des Netzes nicht auf dem ganzen  $\mathbb{R}^n$  aufgrund der  $L_1$ -Norm  $\|\cdot\|_1$  stetig differenzierbar ist.

Um hier die gleichgradige Stetigkeit im Eingangsraum nachzuweisen, reicht es aus, die Lipschitz-Stetigkeit von (3.44) zu zeigen. Denn nach [Heu03, BSM05] umfasst die Lipschitz-Stetigkeit ebenso die gleichmäßige Stetigkeit bei einer Funktion. Für die Lipschitz-Stetigkeit gilt [BSM05]:

**Definition 5** (Lipschitz-Stetigkeit). *Eine Funktion  $f : X \rightarrow Y$  heißt Lipschitz-stetig, wenn es ein  $K \in \mathbb{R}$  gibt mit  $|f(x) - f(y)| \leq K \cdot |x - y|$  für alle  $x, y \in X$ .*

Die Erfüllung der Lipschitz-Bedingung ist stets gegeben, wenn  $f(x)$  in dem betrachteten Gebiet eine beschränkte Ableitung besitzt.

Für das Basisfunktionsnetzwerk mit Produkt-Stabilisatoren folgt, dass die Funktion  $g(x) = e^{-|x|}$  Lipschitz-stetig mit einer Lipschitz-Konstante  $K = 1$  ist. Aufgrund der Lipschitz-Stetigkeit von  $e^{-x}$  mit  $K = 1$  kann gefolgert werden, dass

$$|e^{-a} - e^{-b}| \leq |a - b| = ||x| - |y|| \leq |x - y| \quad \text{mit } a = |x|, b = |y| \quad (3.45)$$

gilt.

Als Konsequenz aus ihrer Lipschitz-Stetigkeit besitzt die Funktion  $e^{-|x|}$  die gleichmäßige Stetigkeit, die mit der Lipschitz-Stetigkeit eingeschlossen wird. Da die Funktion  $e^{-|x|}$  Lipschitz-stetig ist, ist auch jede verschobene Funktion  $e^{-|x-a|}$  ebenfalls Lipschitz-stetig [BSM05].

Zunächst soll auch hier die gleichgradige Stetigkeit für den Eingangsraum untersucht werden. Für die gleichgradige Stetigkeit muss die Differenz der Ausgänge für zwei verschiedene Eingänge berechnet werden

$$|f_m(\mathbf{x}) - f_m(\mathbf{y})| = \left| \sum_{i=1}^m \alpha_i e^{-\|\mathbf{x}-\mathbf{c}_i\|_1} - \sum_{i=1}^m \alpha_i e^{-\|\mathbf{y}-\mathbf{c}_i\|_1} \right|. \quad (3.46)$$

Somit kann (3.46) vereinfacht werden

$$|f_m(\mathbf{x}) - f_m(\mathbf{y})| = \left| \sum_{i=1}^m \alpha_i (e^{-\|\mathbf{x}-\mathbf{c}_i\|_1} - e^{-\|\mathbf{y}-\mathbf{c}_i\|_1}) \right| \quad (3.47)$$

$$\leq \sum_{i=1}^m |\alpha_i| \|\mathbf{x} - \mathbf{y}\|_1 \quad (3.48)$$

$$= \sum_{i=1}^m |\alpha_i| d(\mathbf{x}, \mathbf{y}) = \epsilon, \quad (3.49)$$

wobei  $d(\mathbf{x}, \mathbf{y})$  eine Metrik auf dem Eingangsraum beschreibt.

Somit kann  $\delta$  zu

$$\delta = \frac{\epsilon}{\sum_{i=1}^m |\alpha_i|} \quad (3.50)$$

gewählt werden.

Bedingt durch (3.50) besitzt das Basisfunktionsnetzwerk mit Produkt-Stabilisatoren ebenfalls nicht die Eigenschaft der gleichgradigen Stetigkeit im Eingangsraum, da  $\delta$  und  $\epsilon$  von den Ausgangsgewichten der Neuronen  $\alpha_i$  abhängig sind. Diese wiederum sind von der zu approximierenden Funktion abhängig, und somit können  $\delta$  und  $\epsilon$  nicht unabhängig von den Funktionen aus der Funktionsmenge (2.35) gewählt werden. Somit lässt sich hier ohne Einschränkungen keine gleichgradige Stetigkeit beweisen.

Um auch hier ein neuronales Netz mit der Eigenschaft der gleichgradigen Stetigkeit im Eingangsraum zu erhalten, müssen Randbedingungen eingeführt werden. Mit den Randbedingungen aus (3.18) folgt auch hier für (3.49)

$$|f_m(\mathbf{x}) - f_m(\mathbf{y})| \leq \sum_{i=1}^m |\alpha_i| d(\mathbf{x}, \mathbf{y}) \leq \sum_{i=1}^m B d(\mathbf{x}, \mathbf{y}) = mB d(\mathbf{x}, \mathbf{y}) = \epsilon \quad (3.51)$$

$$\Rightarrow \delta = \frac{\epsilon}{mB}. \quad (3.52)$$

Mit dieser Randbedingung der oberen Grenze der Gewichte sind  $\delta$  und  $\epsilon$  unabhängig von der zu approximierenden Funktion, und somit kann dem Netzwerk die Eigenschaft der gleichgradigen Stetigkeit im Eingangsraum zugeordnet werden.

Analoge Betrachtungen können für den Parameterraum durchgeführt werden. Da die Funktionsmenge des Basisfunktionsnetzes bzgl. der Ausgangsgewichte stetig differenzierbar ist, kann der Mittelwertsatz angewendet werden, und es folgt

$$\begin{aligned} |f_m(\alpha^1) - f_m(\alpha^2)| &\leq \|\nabla_{\alpha} f_m(\boldsymbol{\xi})\|_1 d_{\alpha}(\alpha^1, \alpha^2) \\ &= \sum_{i=1}^m \underbrace{|e^{-\|\mathbf{x} - \mathbf{c}_i\|_1}|}_{\leq 1} d_{\alpha}(\alpha^1, \alpha^2) \end{aligned}$$

$$\leq m \cdot d_{\alpha}(\alpha^1, \alpha^2) = \epsilon \quad (3.53)$$

$$\Rightarrow \delta = \frac{\epsilon}{m}. \quad (3.54)$$

Als Konsequenz aus (3.54) folgt, dass die Funktionsmenge gleichgradig stetig bzgl. ihrer Ausgangsgewichte ist, da  $\delta$  und  $\epsilon$  unabhängig von der Funktion gewählt werden können.

Hinsichtlich der Zentren kann auf die Lipschitz-Stetigkeit zurückgegriffen werden, und man erhält

$$\begin{aligned} |f(\mathbf{c}_i^1) - f(\mathbf{c}_i^2)| &= \left| \sum_{i=1}^m \alpha_i e^{-\|\mathbf{x} - \mathbf{c}_i^1\|_1} - \sum_{i=1}^m \alpha_i e^{-\|\mathbf{x} - \mathbf{c}_i^2\|_1} \right| \\ &\leq \sum_{i=1}^m |\alpha_i| \|\mathbf{c}_i^1 - \mathbf{c}_i^2\|_1 \end{aligned} \quad (3.55)$$

$$\begin{aligned} &\leq \max_{i=1, \dots, m} \|\mathbf{c}_i^1 - \mathbf{c}_i^2\|_1 \sum_{i=1}^m |\alpha_i| \\ &= d_{\mathbf{c}_i}(\mathbf{c}_{\xi}^1, \mathbf{c}_{\xi}^2) \sum_{i=1}^m |\alpha_i| = \epsilon. \end{aligned} \quad (3.56)$$

Somit kann  $\delta$  mit Hilfe von (3.56) zu

$$\delta = \frac{\epsilon}{\sum_{i=1}^m |\alpha_i|} \quad (3.57)$$

gewählt werden. Man erkennt hier, dass ähnlich wie bei den RBF  $\delta$  und  $\epsilon$  von den Gewichten der Funktion abhängig sind und somit nicht unabhängig von dieser gewählt werden können. Als Folge ist die Funktionsmenge nicht gleichgradig stetig. Allerdings mit den Randbedingungen aus (3.18) kann auch hier die gleichgradige Stetigkeit erreicht und der mittlere quadratische Fehler nach oben abgeschätzt werden. Diese Ergebnisse finden sich in [ER05a].

### 3.1.2 LCNN

Für die gleichgradige Stetigkeit des LCNNs können die gleichen Berechnungen aus Abschnitt 3.1.1.1 angewendet werden. Auch hier muss für die gleichgradige Stetigkeit die Differenz des Ausgangs für zwei verschiedene Eingangsvektoren berechnet werden.

Für den Eingangsraum des LCNNs folgt

$$|y_m(\mathbf{x}) - y_m(\mathbf{y})| = |\nabla y_m(\boldsymbol{\xi}) \cdot (\mathbf{x} - \mathbf{y})|, \quad (3.58)$$

wobei  $y_m(\mathbf{x})$  die Übertragungsfunktion des LCNNs mit  $m$  lokalen Funktionen bezeichnet.

Für den Gradienten zur Anwendung des Mittelwertsatzes folgt

$$\begin{aligned} \frac{\partial y_m(\mathbf{x})}{\partial x_n} &= \sum_{i=1}^m \alpha_i \frac{\partial \mathcal{L}_i(\mathbf{x})}{\partial x_n} = \sum_{i=1}^m \alpha_i \sigma'_o(f(\mathbf{x}) - b) \frac{\partial f(\mathbf{x})}{\partial x_n} \\ &= \sum_{i=1}^m \alpha_i \sigma'_o(\cdot) \sum_{j=1}^n \frac{\partial l_{ij}}{\partial x_n} = \sum_{i=1}^m \alpha_i \sigma'_o(\cdot) \sum_{j=1}^n w_{ijn} (\sigma'(\cdot) - \sigma'(\cdot)). \end{aligned} \quad (3.59)$$

Dabei gilt für die Ableitung der Sigmoid

$$\sigma'(x) = \frac{\partial \sigma(x)}{\partial x} = \frac{k_1 \cdot e^{-k_1 x}}{(1 + e^{-k_1 x})^2} = k_1 \cdot \sigma(x)(1 - \sigma(x)) \quad (3.60)$$

$$\sigma'_o(x) = \frac{\partial \sigma_o(x)}{\partial x} = 2 \frac{k_2 \cdot e^{-k_2 x}}{(1 + e^{-k_2 x})^2} = k_2 \cdot \sigma_o(x) \left(1 - \frac{\sigma_o(x)}{2}\right). \quad (3.61)$$

Somit kann das Maximum der Ableitungen für beide Sigmoide bestimmt werden

$$\lim_{x \rightarrow 0} \sigma'(x) = \frac{1}{4}k_1 \quad (3.62)$$

$$\lim_{x \rightarrow 0} \sigma'_o(x) = \frac{1}{2}k_2. \quad (3.63)$$

Damit lässt sich (3.58) mit Hilfe des Mittelwertsatzes, der Dreiecksungleichung und dem Maximum der Sigmoide vereinfachen, und es folgt

$$|y_m(\mathbf{x}) - y_m(\mathbf{y})| \leq \sum_{\nu=1}^n \left| \sum_{i=1}^m \alpha_i \sigma'_o(\cdot) \sum_{j=1}^n w_{ij\nu} (\sigma'(\cdot) - \sigma'(\cdot)) \right| d(\mathbf{x}, \mathbf{y}) \quad (3.64)$$

$$\leq \sum_{\nu=1}^n \left| \sum_{i=1}^m \frac{k_2 \alpha_i}{2} \sum_{j=1}^n \frac{k_1}{4} w_{ij\nu} \right| d(\mathbf{x}, \mathbf{y}) \quad (3.65)$$

$$\leq \frac{k_1 k_2}{8} \sum_{\nu=1}^n \sum_{i=1}^m \sum_{j=1}^n |\alpha_i w_{ij\nu}| d(\mathbf{x}, \mathbf{y}) = \epsilon \quad (3.66)$$

und für  $\delta$  gilt somit

$$\delta = \frac{8\epsilon}{k_1 k_2 \sum_{\nu=1}^n \sum_{i=1}^m \sum_{j=1}^n |\alpha_i w_{ij\nu}|}. \quad (3.67)$$

Nach (3.66) sind  $\epsilon$  und  $\delta$  von den Gewichten  $w_{ij\nu}$  abhängig, die wiederum von der zu approximierenden Funktion abhängen. Damit können  $\epsilon$  und  $\delta$  nicht unabhängig von der Funktion gewählt werden, und somit besitzt das LCNNs ebenfalls nicht die Eigenschaft der gleichgradigen Stetigkeit im Eingangsraum.

Werden jedoch die Parameter des LCNNs begrenzt, so dass für diese gilt

$$|w_{ij\nu}| \leq T \quad (3.68)$$

$$|\alpha_\nu| \leq B, \quad (3.69)$$

kann (3.66) weiter vereinfacht werden

$$|y_m(\mathbf{x}) - y_m(\mathbf{y})| \leq \frac{k_1 k_2}{8} \sum_{\nu=1}^n \sum_{i=1}^m \sum_{j=1}^n B \cdot T \cdot d(\mathbf{x}, \mathbf{y}) \quad (3.70)$$

$$= \frac{k_1 k_2}{8} n^2 m B T = \epsilon. \quad (3.71)$$

Damit kann  $\delta$  zu

$$\delta = \frac{8\epsilon}{k_1 k_2 n^2 m B T} \quad (3.72)$$

gewählt werden, und somit besitzt das LCNNs für Einschränkungen der Parameter nach (3.68) und (3.69) die Eigenschaft der gleichgradigen Stetigkeit.

Der mittlere quadratische Fehler bei rauschenden Eingängen kann mit den gleichen Randbedingungen wie beim RBF nach oben abgeschätzt werden, und es ergibt sich für diesen

$$\text{mse} \leq \frac{k_1^2 k_2^2}{64} \left( \sum_{\nu=1}^n \sum_{i=1}^m \sum_{j=1}^n |\alpha_i w_{ij\nu}| \right)^2 n \sigma_n^2. \quad (3.73)$$

Bei der zusätzlichen Einschränkung der Gewichte ergibt sich

$$\text{mse} \leq \frac{k_1^2 k_2^2}{64} n^4 m^2 B^2 T^2 n \sigma_n^2 = \frac{k_1^2 k_2^2}{64} n^5 m^2 B^2 T^2 \sigma_n^2. \quad (3.74)$$

Weiter soll die Eigenschaft im Gewichtsraum betrachtet werden. Es seien  $\mathcal{W}^1$  und  $\mathcal{W}^2$  zwei Punkte im Parameterraum, und  $d(\mathcal{W}^1, \mathcal{W}^2)$  bezeichne eine Metrik zwischen den beiden Punkten. Somit gilt für zwei Punkte im Parameterraum bzgl. der Differenz im Ausgang

$$|y_m(\mathcal{W}^1) - y_m(\mathcal{W}^2)| \leq (\|\nabla_w y_m(\boldsymbol{\xi})\|_1 + \|\nabla_{\mathbf{r}} y_m(\boldsymbol{\xi})\|_1 + \|\nabla_{\alpha} y_m(\boldsymbol{\xi})\|_1) d(\mathcal{W}^1, \mathcal{W}^2), \quad (3.75)$$

wobei  $\nabla_w$  den Gradient bzgl. der Gewichte,  $\nabla_{\mathbf{r}}$  den Gradient bzgl. der Zentren und  $\nabla_{\alpha}$  bzgl. der Gewichte bezeichnen.

Somit folgt für die einzelnen Gradienten

$$\|\nabla_w y_m(\boldsymbol{\xi})\|_1 = \sum_{\mu=1}^m \sum_{i=1}^n \sum_{j=1}^n \left| \alpha_{\mu} \frac{\partial \mathcal{L}_{\mu}}{\partial w_{\mu ij}} \right| \quad (3.76)$$

$$= \sum_{\mu=1}^m \sum_{i=1}^n \sum_{j=1}^n |\alpha_{\mu} \sigma'_o(\cdot) (\sigma'(\cdot) - \sigma'(\cdot))(x_j - r_{\mu j})| \quad (3.77)$$

$$\leq \frac{k_1 k_2}{8} \sum_{\mu=1}^m \sum_{i=1}^n \sum_{j=1}^n |\alpha_{\mu}| |x_j - r_{\mu j}| \quad (3.78)$$

$$= \frac{k_1 k_2}{8} \sum_{\mu=1}^m n |\alpha_{\mu}| \|\mathbf{x} - \mathbf{r}_{\mu}\|_1 \quad (3.79)$$

sowie

$$\|\nabla_{\mathbf{r}} y_m(\boldsymbol{\xi})\|_1 \leq \sum_{\mu=1}^m \sum_{j=1}^n |\alpha_\mu| \left| \frac{\partial \mathcal{L}_\mu}{\partial r_{\mu j}} \right| \quad (3.80)$$

$$= \sum_{\mu=1}^m \sum_{j=1}^n |\alpha_\mu| \left| \sigma'_o(\cdot) \sum_{i=1}^n (-w_{\mu ij}) (\sigma'(\cdot) - \sigma(\cdot)) \right| \quad (3.81)$$

$$\leq \sum_{\mu=1}^m \sum_{j=1}^n \frac{|\alpha_\mu| k_1 k_2}{8} \sum_{i=1}^n |w_{\mu ij}| \quad (3.82)$$

$$= \frac{k_1 k_2}{8} \sum_{\mu=1}^m |\alpha_\mu| \sum_{j=1}^n \sum_{i=1}^n |w_{\mu ij}| \quad (3.83)$$

und

$$\|\nabla_{\alpha} y_m(\boldsymbol{\xi})\|_1 = \sum_{\mu=1}^m |\mathcal{L}_\mu| \leq \sum_{\mu=1}^m 1 = m. \quad (3.84)$$

Mit (3.79), (3.83) und (3.84) kann (3.75) weiter abgeschätzt werden, und es folgt

$$\begin{aligned} |y_m(\mathcal{W}^1) - y_m(\mathcal{W}^2)| &\leq \left( \frac{k_1 k_2}{8} \sum_{\mu=1}^m n |\alpha_\mu| \|\mathbf{x} - \mathbf{r}_\mu\|_1 + \right. \\ &\quad \left. \frac{k_1 k_2}{8} \sum_{\mu=1}^m |\alpha_\mu| \sum_{j=1}^n \sum_{i=1}^n |w_{\mu ij}| + m \right) d(\mathcal{W}^1, \mathcal{W}^2) = \epsilon \end{aligned} \quad (3.85)$$

sowie für  $\delta$  gilt

$$\delta = \frac{\epsilon}{\frac{k_1 k_2}{8} \sum_{\mu=1}^m n |\alpha_\mu| \|\mathbf{x} - \mathbf{r}_\mu\|_1 + \frac{k_1 k_2}{8} \sum_{\mu=1}^m |\alpha_\mu| \sum_{j=1}^n \sum_{i=1}^n |w_{\mu ij}| + m}. \quad (3.86)$$

Somit kann aus (3.86) gefolgert werden, dass auch im Parameterraum das LCNN keine gleichgradig stetige Funktionsmenge beschreibt, da  $\epsilon$  und  $\delta$  nicht unabhängig von der Funktion gewählt werden können. Diese hängen von den Parametern  $\alpha_\mu$  bzw.  $w_{\mu ij}$  der Funktion des LCNNs ab.

**Bemerkung.** In (3.75) wurde die Cauchy-Schwarzsche Ungleichung und anschließend die Äquivalenz der  $L_p$ -Normen aus (3.4) angewendet, wobei vom Betrag in (3.76) ausgegangen wurde. Wird nach der Cauchy-Schwarzschen Ungleichung die Äquivalenz der Normen nur teilweise auf die einzelnen Terme angewendet, so ergibt sich in (3.79) die euklidische Norm.

Mit den Randbedingungen aus (3.68) und (3.69) ergibt sich noch nicht ein gleichgradig stetiges Netz, da zusätzlich noch die Manhattan-Norm begrenzt sein muss

$$\|\mathbf{x} - \mathbf{r}_\mu\|_1 \leq C. \quad (3.87)$$

**Bemerkung.** Dabei ist schon ausreichend, wenn anstatt der Manhattan-Distanz zwischen Eingangsvektor und Zentren die euklidische Distanz nach oben begrenzt ist. Es kann folgendes Lemma bewiesen werden

*Lemma 3.1.1.* Die euklidische Norm sei durch eine obere Schranke  $B$  begrenzt. Dann ist auch die Manhattan-Norm durch eine Schranke  $\hat{B}$  nach oben beschränkt, und es gilt

$$\sum_{i=1}^n |x_i|^2 \leq B \Rightarrow \sum_{i=1}^n |x_i| \leq \hat{B}. \quad (3.88)$$

*Beweis.* Der Beweis erfolgt durch vollständige Induktion. Für  $n = 1$  gilt:

$$|x_1|^2 \leq B \Rightarrow |x_1| \leq \hat{B}, \quad (3.89)$$

was mit den Eigenschaften von Ungleichungen [BSM05] bewiesen werden kann.

Nun gelte

$$\sum_{i=1}^n |x_i|^2 \leq B \Rightarrow \sum_{i=1}^n |x_i| \leq \hat{B} \quad (3.90)$$

für ein beliebiges  $n \in \mathbb{N}$  und somit folgt für  $n + 1$

$$\sum_{i=1}^{n+1} |x_i|^2 = \sum_{i=1}^n |x_i|^2 + |x_{n+1}|^2 \leq B_n + |x_{n+1}|^2 \leq B_{n+1}. \quad (3.91)$$

Damit folgt, dass  $|x_{n+1}|^2$  und somit auch  $|x_{n+1}|$  begrenzt sein muss. Damit folgt

$$\sum_{i=1}^n |x_i| + |x_{n+1}| \leq \hat{B}_n + |x_{n+1}| \leq \hat{B}_{n+1}, \quad (3.92)$$

da  $|x_{n+1}|$  beschränkt ist. □

Damit ergibt sich ein gleichgradig stetiges Netz auch im Parameterraum und hierfür gilt

$$|y_m(\mathcal{W}^1) - y_m(\mathcal{W}^2)| \leq \left( \frac{k_1 k_2}{8} m \cdot n \cdot B \cdot C + \frac{k_1 k_2}{8} m \cdot n^2 \cdot B \cdot T + m \right) d(\mathcal{W}^1, \mathcal{W}^2) \quad (3.93)$$

bzw.

$$\delta = \frac{\epsilon}{\frac{k_1 k_2}{8} m \cdot n \cdot B \cdot C + \frac{k_1 k_2}{8} m \cdot n^2 \cdot B \cdot T + m}. \quad (3.94)$$

Somit können  $\epsilon$  und  $\delta$  unabhängig von der Funktion gewählt werden.

Der mittlere quadratische Fehler kann mit Hilfe von (3.85) abgeschätzt werden, und es ergibt sich

$$\text{mse} \leq \left( \frac{k_1 k_2}{8} \sum_{\mu=1}^m n |\alpha_\mu| \|\mathbf{x} - \mathbf{r}_\mu\|_1 + \frac{k_1 k_2}{8} \sum_{\mu=1}^m |\alpha_\mu| \sum_{j=1}^n \sum_{i=1}^n |w_{\mu ij}| + m \right)^2 \times m(n^2 + n + 1)\sigma_n^2. \quad (3.95)$$

Insgesamt benötigt das LCNN zur Darstellung einer einzigen lokalen Funktion  $n^2 + n + 1$  Parameter, so dass bei insgesamt  $m$  lokalen Funktionen sich  $m \cdot (n^2 + n + 1)$  mögliche Rauschquellen ergeben. Ohne die Randbedingungen kann hier keine obere Grenze bzgl. des mittleren quadratischen Fehlers angegeben werden. Mit den Randbedingungen jedoch folgt

$$\text{mse} \leq \left( \frac{k_1 k_2}{8} m \cdot n \cdot B \cdot C + \frac{k_1 k_2}{8} m \cdot n^2 \cdot B \cdot T + m \right)^2 m(n^2 + n + 1)\sigma_n^2. \quad (3.96)$$

### 3.1.3 Parameterempfindlichkeit

Aufgrund von Rauschen oder begrenzter Genauigkeit zur Speicherung der Gewichte ergeben sich Fehler in der Aktivierung eines Neurons und somit ebenfalls im Ausgang des neuronalen Netzes. Der Einfluss jeden Parameters auf die Störung kann bestimmt werden, indem seine Empfindlichkeit ermittelt wird, ähnlich der Koeffizientenempfindlichkeit bei digitalen Filtern. Für diese Empfindlichkeit folgt [Fra03]

$$S_w^y = \frac{\partial y}{\partial w} \frac{w}{y}, \quad (3.97)$$

wobei  $w$  den untersuchten Parameter bezeichnet. Die Empfindlichkeit wird somit durch den normalisierten Gradienten ausgedrückt.

Da im Ausgang alle Basisfunktionen linear überlagert werden und so Abweichungen der einzelnen Neuronen sich linear mischen, wird die Empfindlichkeit der Parameter nur für eine isolierte Gaußfunktion bestimmt.

So ergibt sich eine Empfindlichkeit hinsichtlich des Ausgangsgewichtes der Gaußfunktion von

$$S_\alpha^y = \frac{\alpha e^{-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{2\sigma^2}}}{\alpha e^{-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{2\sigma^2}}} = 1 \quad (3.98)$$

sowie für einen Eintrag des Zentrums

$$S_c^y = \frac{c_i(x_i - c_i)}{\sigma^2} \quad (3.99)$$

und für die Varianz der Gaußfunktion

$$S_\sigma^y = \frac{\|\mathbf{x} - \mathbf{c}\|^2}{\sigma^2}. \quad (3.100)$$

Wie man an den Gleichungen ablesen kann, ist die Empfindlichkeit des Ausgangsgewichtes bei einer Gaußfunktion konstant, während die Empfindlichkeit der anderen Parameter besonders für kleine Varianzen sehr hoch ist. Anschaulich ist dies mit der schmalen Breite der Gaußfunktionen zu erklären, so dass bei kleinen Störungen in den Parametern eine große Abweichung im Ausgang entsteht.

Ebenfalls für große Differenzen zwischen dem Eingang und dem Zentrum besitzt eine Basisfunktion eine hohe Empfindlichkeit bezüglich Störungen in den Zentren und den Varianzen. Aufgrund begrenzter Auflösung und Speicherkapazität können die Gewichte nicht mit ihrem exakten Wert gespeichert werden, was zu einer Störung des Ausgangs und zu dieser Empfindlichkeit führt. Daneben resultiert die begrenzte Auflösung in obere und untere Grenzen für die Parameter (vgl. Kapitel 3.1.1). Als Folge ergibt sich auch für die Parameterempfindlichkeit eine obere Grenze.

Mit den oberen Schranken aus dem vorherigen Abschnitt folgt direkt

$$S_c^y \leq \frac{K}{T^2} \quad (3.101)$$

$$S_\sigma^y \leq \frac{C^2}{T^3}, \quad (3.102)$$

wobei die Konstante  $K$  sich aus der größten darstellbaren Zahl für die Zentren und aus der Differenz zwischen größter und kleinster darstellbarer Zahl für die Eingänge und die Zentren bestimmt.

Aufgrund der hohen Empfindlichkeit der Parameter für die Zentren und die Varianzen können hohe relative Fehler entstehen. Wegen der Form der Gaußfunktion ist dies besonders in den Ästen zu beobachten, allerdings sind für die Approximation von Funktionen diese Bereiche eher uninteressant, da hier eine Gaußfunktion nur wenig zur Approximation beiträgt. So ist der maximale absolute Fehler wesentlich interessanter. Dieser ergibt sich, wenn die Empfindlichkeit mit der Aktivierung der Neuronen multipliziert wird, und für den absoluten Fehler folgt

$$\Delta y_\alpha = e^{-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{2\sigma^2}} \cdot \Delta\alpha \leq 1 \cdot \Delta\alpha \quad (3.103)$$

$$\Delta y_c = \frac{\alpha(x_i - c_i)}{\sigma^2} e^{-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{2\sigma^2}} \cdot \Delta c_i \leq \frac{\alpha e^{-\frac{1}{2}}}{\sigma} \cdot \Delta c_i \quad (3.104)$$

$$\Delta y_\sigma = \frac{\alpha \|\mathbf{x} - \mathbf{c}\|^2}{\sigma^3} e^{-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{2\sigma^2}} \cdot \Delta\sigma \leq \frac{2\alpha e^{-1}}{\sigma} \cdot \Delta\sigma. \quad (3.105)$$

Da für alle Abweichungen die gleichen Parameter verwendet werden, hängt bei identischen Abweichungen der Parameter der absoluten Fehler besonders von den

Ausgangsgewichten und der Breite der Gaußfunktion ab. Die gleichen Abweichungen ergeben sich z.B. bei Verwendung der gleichen Auflösung für alle Parameter. Die maximale Abweichung kann für den Definitionsbereich der Eingänge bestimmt werden, so dass sich die Abschätzungen in (3.103), (3.104) und (3.105) ergeben.

Wie man an diesen Gleichungen erkennen kann, sollten für eine möglichst robuste Approximation kleine Ausgangsgewichte und große Varianzen verwendet werden. So kann die Abweichung der Gaußfunktionen gering gehalten werden, was analog zu den Ergebnissen aus Abschnitt 3.1.1 gilt.

Nach dem Training sind die Parameter des Netzes festgelegt, so dass für eine Umsetzung des Netzes in Hardware die benötigte Auflösung für die Gewichte bestimmt werden kann<sup>1</sup>. Mit Hilfe der aktuellen Parameter kann ermittelt werden, für welche Gewichte die höchste Auflösung verwendet werden muss, damit sich ein kleiner absoluter Fehler im Ausgang ergibt. Die Ausgangsgewichte sollten dabei mit einer hohen Auflösung umgesetzt werden, da so ein kleiner Approximationsfehler erreicht werden kann. Zum einen wird der absolute Fehler aufgrund der kleineren Quantisierung der Ausgangsgewichte abgesenkt, und zum anderen kann durch ein wiederholtes (simples) Trainingsverfahren der absolute Fehler aufgrund begrenzter Auflösung der Varianzen oder Zentren verringert werden.

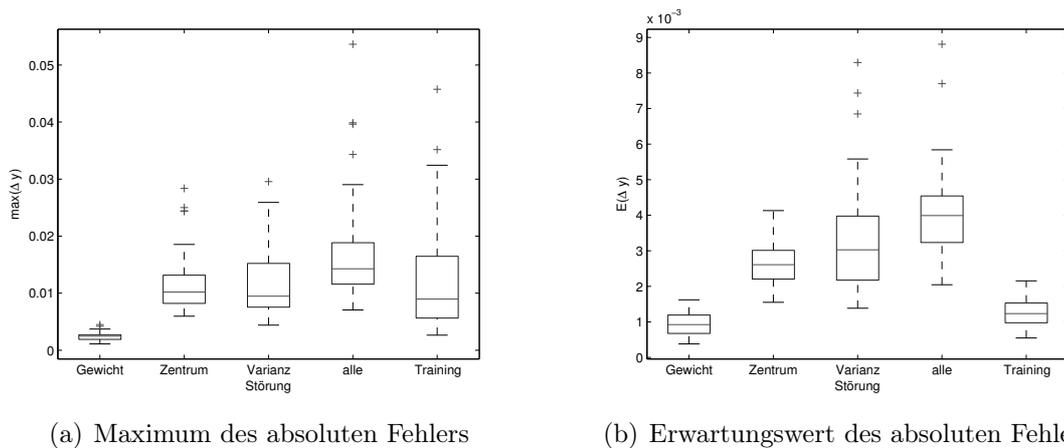
Hierfür wird mit den Trainingsvektoren ein überbestimmtes, lineares Gleichungssystem für die Ausgangsgewichte aufgestellt, wobei die Lösung mit Hilfe der Pseudoinversen die beste Lösung im Sinne minimaler Fehlerquadrate darstellt<sup>2</sup> [GLL96, Hay99].

Abbildung 3.2 zeigt den maximalen absoluten Fehler und seinen Erwartungswert bei Störungen in den Parametern eines RBF-Netzes. Dazu wurde mit insgesamt 50 Durchläufen ein neuronales Netz mit 50 Gaußfunktionen trainiert, und im Anschluss an das Lernverfahren sind die Parameter des Netzes mit einer Auflösung von 9 bit für die Nachkommastellen quantisiert worden. So entsteht ein maximaler Quantisierungsfehler von  $1/1024$ . Abbildung 3.2 zeigt dabei die Effekte der einzelnen Quantisierungen für die Ausgangsgewichte, die Zentren, die Varianzen und für Störungen in allen Parametern gleichzeitig. Zusätzlich wurden nach Quantisierung aller Parameter die Ausgangsgewichte mit Hilfe des linearen Gleichungssystems erneut bestimmt, und diese neuen Werte anschließend quantisiert.

Wie man an Abbildung 3.2(a) und Darstellung 3.2(b) erkennt, erzeugen Störungen in den Varianzen und den Zentren einen größeren absoluten Fehler als die Quantisierung der Ausgangsgewichte. Die Effekte aufgrund der Störungen in den Parametern

<sup>1</sup> Für den Fall, dass das neuronale Netz auf einen DSP oder einem FPGA umgesetzt wird, kann die Auflösung an das Problem angepasst werden, um so eine effiziente Realisierung hinsichtlich des Speicherbedarfs zu erreichen. Bei einem Einsatz auf statischer Hardware sollte die maximal verfügbare Auflösung verwendet werden.

<sup>2</sup> Ein neuronales Netz kann bei zufälliger Initialisierung der Parameter in der versteckten Ebene jede beliebige Funktion approximieren, wenn nur die Ausgangsgewichte an das aktuelle Problem angepasst werden (siehe Abschnitt 4.3 oder z.B. [HZM<sup>+</sup>06, HCS06]).



(a) Maximum des absoluten Fehlers

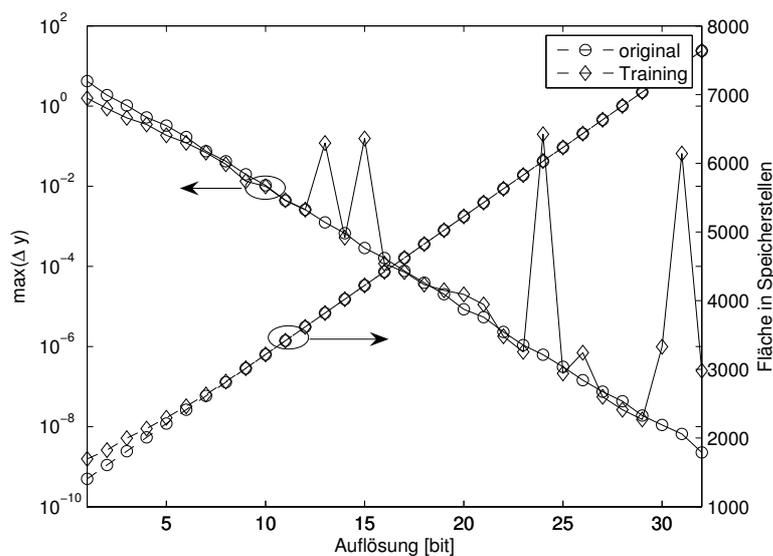
(b) Erwartungswert des absoluten Fehlers

**Abbildung 3.2:** Maximaler Fehler und Erwartungswert der Abweichungen aufgrund Störungen in den Parametern eines RBF-Netzes.

können zu einem Teil kompensiert werden, indem ein erneutes Training mit der veränderten Aktivierung der Neuronen durchgeführt wird. Hier lässt sich eine genauere Approximation erreichen, wobei negative Auswirkungen auf den Flächenbedarf entstehen können. Das erneute Trainingsverfahren kann in Abhängigkeit der aktuellen Aufgabe zu größeren Gewichten führen, was den geringeren Flächenbedarf aufgrund kleiner Auflösung der Varianzen und Zentren wieder kompensiert.

Abbildung 3.3 zeigt den Zusammenhang zwischen Approximationsgenauigkeit und der benötigten Fläche zur Speicherung der Parameter für das gleiche Experiment. Die Fläche wurde als linear abhängig zu der verwendeten Auflösung angenommen und wird in Anzahl von benötigten Speicherstellen des gesamten Netzes angegeben, wobei für alle Parameter eine Festpunktdarstellung mit identischer Auflösung des Bruchanteils angenommen wurde. Für den Vergleich wurden zum einen die Auswirkungen der Quantisierung mit und ohne zusätzlichen Trainingsschritt, wie zuvor erläutert, betrachtet. Abbildung 3.3 zeigt jeweils den Scharmittelwert aus den Experimenten.

Das erneute Trainieren des Netzes, um die Quantisierungsfehler zu beseitigen, wirkt sich nur bei niedriger Auflösung der Signale positiv auf die Approximationseigenschaften aus. Hier lässt sich der maximale absolute Fehler kompensieren, allerdings führt diese bessere Eigenschaft zu einem größeren System. Diese ergibt sich aufgrund einer höheren Auflösung des ganzzahligen Anteils, da größere Werte gespeichert werden müssen. Bei hohen Auflösungen dagegen bietet ein erneutes Training keinen Vorteil, da weder geringere Fläche noch bessere Eigenschaften des Netzes erreicht werden. Die Ausreißer in den Approximationseigenschaften ergeben sich aufgrund der Randbedingungen des Experimentes und dem Trainingsverfahren. Zum einen wurde zur Bestimmung des Fehlers die Übertragungsfunktion des Netzes als Zielfunktion verwendet, und zum anderen liefert die Lösung des LGS das Minimum



**Abbildung 3.3:** Maximaler absoluter Fehler und Flächenbedarf bei der Quantisierung der Parameter

bzgl. des mittleren quadratischen Fehlers, welches nicht notwendigerweise das Minimum des absoluten Fehlers sein muss.

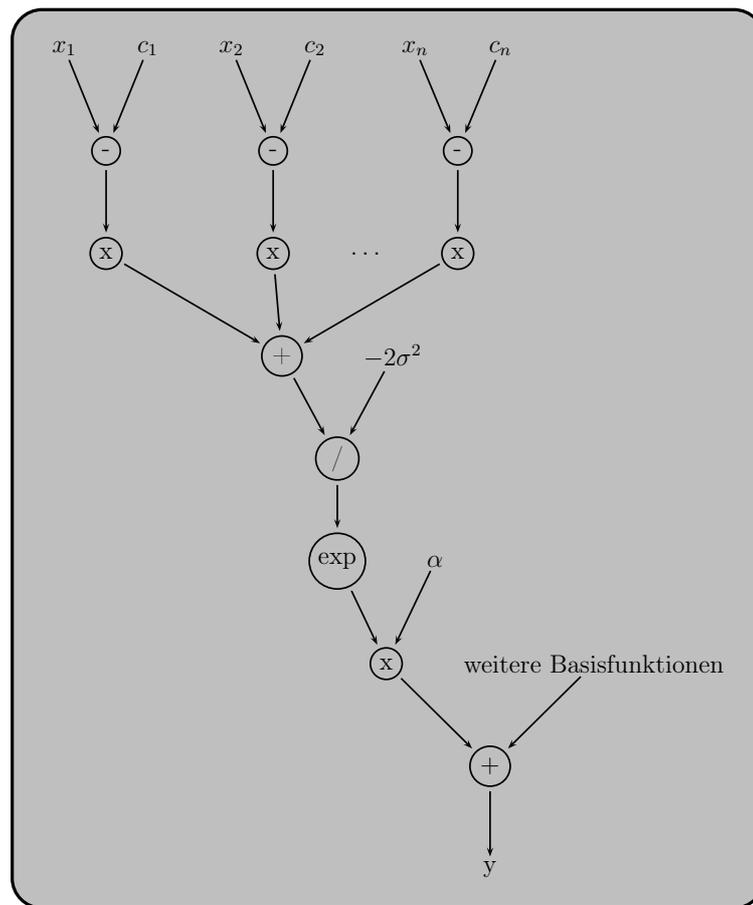
## 3.2 Fehlertoleranz

Neben der Robustheit ist auch die Toleranz gegenüber Fehlern eine gewünschte Eigenschaft für nanoelektronische Systeme. In diesem Abschnitt werden mathematische Modelle vorgestellt, mit denen die Auswirkungen von Defekten und Fehler während der Laufzeit auf neuronale Netze bestimmt werden können. Man erhält so eine abstrakte Beschreibung, die sich auf verschiedene Technologien übertragen lässt.

### 3.2.1 Fehlerquellen

Um die jeweiligen Netze in ihrer Komplexität vergleichen und später Aussagen über die Lebensdauer treffen zu können, wird davon ausgegangen, dass die elementaren Operationen, wie Multiplikation und Addition, jeweils mit zwei Operanden durchgeführt werden. So werden zur Berechnung eines Produktes oder einer Summe mit insgesamt  $n$  Faktoren oder Summanden  $n - 1$  Recheneinheiten<sup>3</sup> mit zwei Eingängen benötigt. So lassen sich die Netze in ihrer Komplexität vergleichen, da auf Grundeinheiten zurückgegriffen wird.

<sup>3</sup> Im Folgenden werden alle Rechenoperationen sofern nicht anderweitig erwähnt mit zwei Eingängen durchgeführt.



**Abbildung 3.4:** Datenflussgraph eines RBF-Netzes, Variante I

Fehler bei der Berechnung können auf einen logischen Stuck-At-Fehler (SA-Fehler) in der jeweiligen Zahlendarstellung zurückgeführt und als zusätzliches Rauschen aufgefasst werden (vgl. Abschnitt 2.3.4). Weiter können die Parameter des Netzes bedingt durch Fehler nicht mehr verändert werden, und besonders bei kleinen Strukturgrößen kann sich ein Fehler oder Defekt auf das globale Verhalten der Architektur auswirken, so dass ein SA-Fehler diesen Effekt erfasst.

Für die Berechnung der Ausgabe eines RBF-Netzes werden drei elementare, mathematische Funktionen benötigt:

- Multiplizierer/Dividierer
- Addierer/Subtrahierer
- exp-Funktion.

Allerdings kann die Ausgabe auf unterschiedliche Art und Weise berechnet werden, je nachdem welche Operation aus (2.28) zuerst ausgeführt wird. In dieser Arbeit werden zwei Vorgehensweisen betrachtet. Die erste Möglichkeit, Variante I, folgt

der mathematischen Beschreibung der radialen Basisfunktionen und bestimmt zunächst die euklidische Distanz zwischen dem Eingangsvektor und einem Zentrum. Die Ausgabe wird dann mit Hilfe der Varianz, der exp-Funktion und dem Ausgangsgewicht bestimmt, was in Abbildung 3.4 dargestellt ist. Für dieses Vorgehen werden zur Bestimmung der Aktivierung einer einzelnen Gaußfunktion

$$\nu_e = 1 \quad \nu_A = 2n - 1 \quad \nu_M = n + 2 \quad (3.106)$$

Elemente benötigt, wobei  $\nu_e$  die Anzahl von exp-Funktionen,  $\nu_A$  die Anzahl von Addierern und  $\nu_M$  die Anzahl von Multiplizierern bezeichnet.

Eine multivariate Gaußfunktion kann auch mittels eindimensionaler Gaußfunktionen berechnet werden, wobei die Ergebnisse der eindimensionalen Funktion durch eine Multiplikation das Ergebnis bestimmen (Variante II). Dies entspricht dem biologisch motivierten Vorgehen aus Abbildung 2.8, bei der das rezeptive Feld aus einzelnen Merkmalsdetektoren zusammengesetzt wird. Diese Methode ist in Abbildung 3.5 dargestellt und insgesamt werden zur Berechnung der Aktivierung

$$\nu_e = n \quad \nu_A = n \quad \nu_M = 3n \quad (3.107)$$

Elemente benötigt.

Analog zum RBF-Netz werden bei einem LCNN ebenfalls drei elementare Rechenoperationen benötigt:

- Multiplizierer
- Addierer/Subtrahierer
- Sigmoide

und für die Bestimmung eines Ausgabewertes einer lokalen Funktion müssen insgesamt

$$\nu_\sigma = 2n + 1 \quad (3.108)$$

$$\nu_A = n^2 + 3n \quad (3.109)$$

$$\nu_M = n^2 + 1 \quad (3.110)$$

Elemente des jeweiligen Typs eingesetzt werden.

Die Anzahl der Elemente wurde dabei minimiert, so dass die Verfahrensweise des LCNNs nicht exakt nach Kapitel 2.2.2 erfolgt. Zur Berechnung der lokalen Funktion wird das Vorgehen aus Abbildung 3.6 verwendet. Für die Darstellung der Ridge-Funktion werden zwei verschobene Sigmoide benötigt, deren Eingangsargumente sich durch

$$\mathbf{w}^T(\mathbf{x} - \mathbf{r}) \pm d = \mathbf{w}^T \mathbf{x} \underbrace{- \mathbf{w}^T \mathbf{r}}_{\Theta_{1,2}} \pm d \quad (3.111)$$

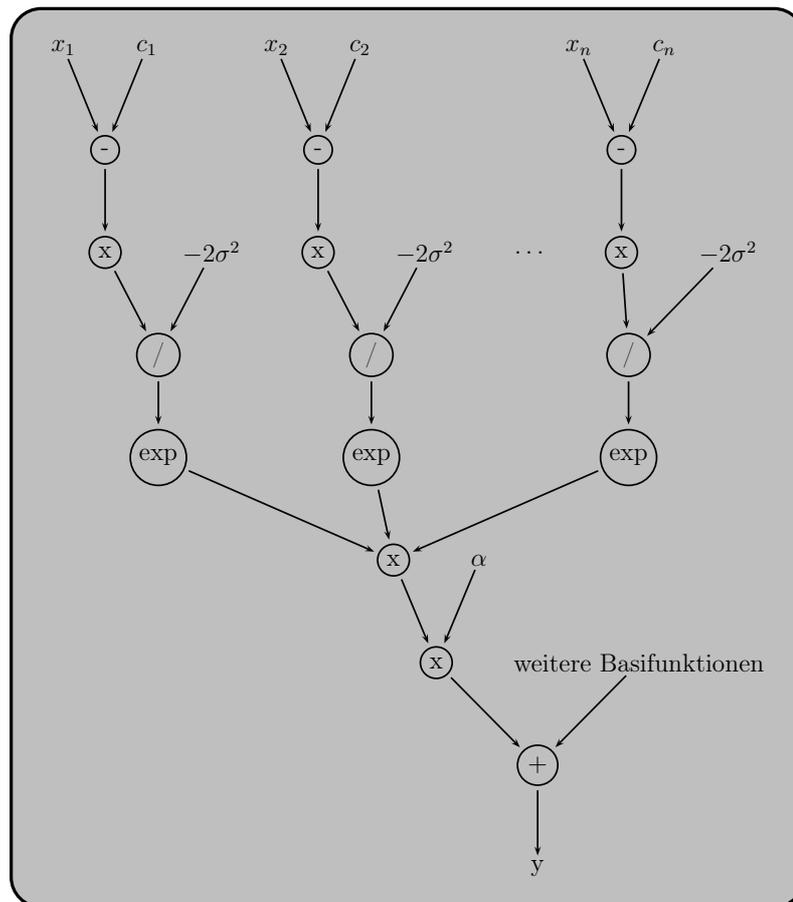


Abbildung 3.5: Datenflussgraph eines RBF-Netzes, Variante II

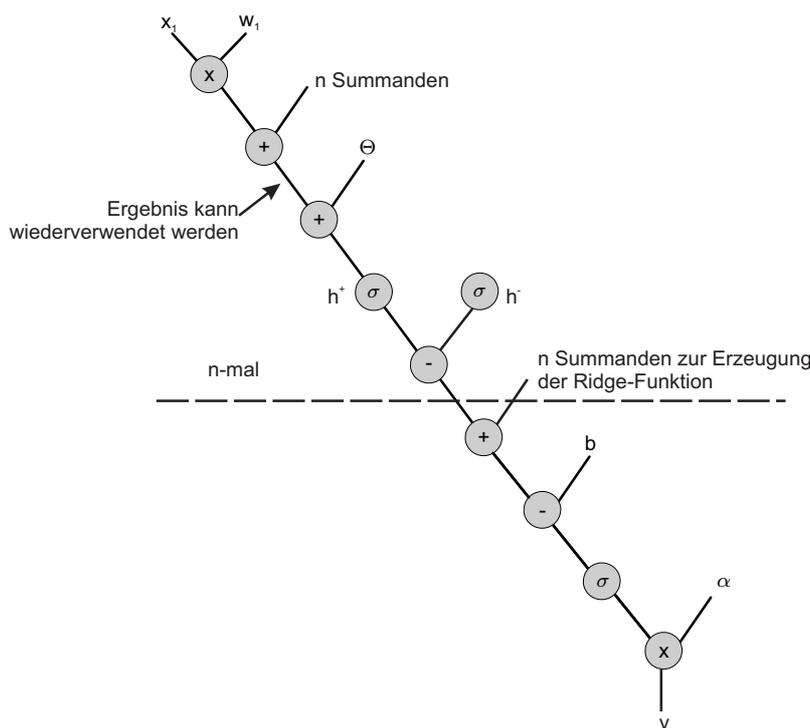


Abbildung 3.6: Datenflussgraph einer lokalen Funktion eines LCNNs

berechnen.

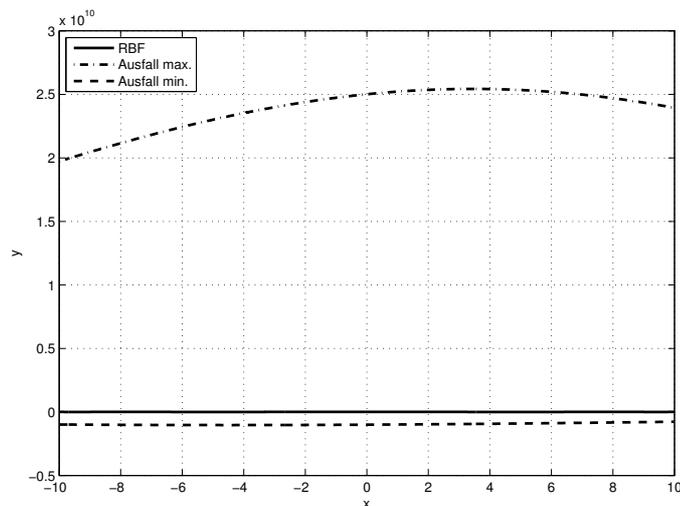
Durch die Einführung von  $\Theta_1$  bzw.  $\Theta_2$  in (3.111) werden weniger Addierer benötigt. Die Darstellung in (3.111) mit  $\Theta$  benötigt  $n$  Addierer und  $n$  Multiplizierer pro Sigmoid, während in der ursprünglichen Form  $2n + 1$  Addierer und  $n$  Multiplizierer pro Sigmoid notwendig waren. Allerdings müssen die beiden Parameter  $\Theta_1$  und  $\Theta_2$  zuvor mit Hilfe der Gewichte und des Zentrums berechnet werden.

Insgesamt werden  $2n$  verschobene Sigmoide zur Erzeugung einer lokalen Funktion benötigt. Allerdings können die Werte des Skalarproduktes  $\mathbf{w}^T \mathbf{x}$  wiederverwendet werden, so dass pro verschobenem Sigmoid neben dem Ergebnis  $\mathbf{w}^T \mathbf{x}$  jeweils ein zusätzlicher Addierer für  $\Theta_1$  bzw.  $\Theta_2$  benötigt wird. Zur Bildung der Funktion  $f$  (vgl. (2.18)) werden  $n - 1$  Addierer für die Summation der verschobenen Sigmoide gebraucht. Abschließend werden noch ein Subtrahierer ( $b$ ), ein Ausgangssigmoid und ein Multiplizierer benötigt. So werden insgesamt die in (3.108)-(3.110) aufgezählten Elemente zur Erzeugung einer lokalen Funktion verwendet.

Für ein komplettes LCNN werden insgesamt  $m$  lokale Funktionen benötigt, sowie zusätzlich  $m - 1$  Addierer. Analog werden auch für ein RBF-Netz insgesamt  $m$  Gaußfunktionen zur Bestimmung der Ausgabe benötigt. Dafür ergeben sich die in Tabelle 3.1 dargestellte Anzahl von Rechenelementen für das jeweilige neuronale Netz.

**Tabelle 3.1:** Anzahl der benötigten Rechenelemente zur Umsetzung des neuronalen Netzes

Netztyp	Addierer	Multiplizierer	exp-Funktion/Sigmoid
RBF <sub>I</sub>	$m(2n - 1) + m - 1$	$m(n + 2)$	$m \cdot 1$
RBF <sub>II</sub>	$m \cdot n + m - 1$	$m \cdot 3n$	$m \cdot n$
LCNN	$m(n^2 + 3n) + m - 1$	$m(n^2 + 1)$	$m(2n + 1)$



**Abbildung 3.7:** Auswirkungen des Ausfalls von Neuronen im RBF-Netz

**Beispiel.** Zur Verdeutlichung der Auswirkung auf ein neuronales Netz, falls ein Neuron bei der Berechnung ausfällt, wird hier das RBF-Netz zur Approximation der si-Funktion verwendet. Von den 10 Neuronen des Netzes wird ein Element aus dem Netz entfernt bzw. dessen Ausgangsgewicht zu Null gesetzt. Wie man an Abbildung 3.7 erkennen kann, entsteht hier ein sehr großer Approximationsfehler  $\Delta y \approx 10^{10}$ , der die Funktionalität des Netzes zur Approximation der si-Funktion zunichtemacht. Weiter erkennt man, dass ebenfalls unterschiedliche Auswirkungen bei dem Entfernen eines Neurons auftreten können. So liegt in dem einen Fall der Fehler in der Größenordnung  $10^{10}$ , während bei dem anderen Fall dieser eine Größenordnung kleiner ist. Dieses Phänomen wird in Abschnitt 4.1 näher betrachtet. Ziel bei den Betrachtungen der Fehlertoleranz ist es, diese Auswirkungen abzuschätzen und durch eine obere Schranke begrenzen zu können. Daneben ist die Betrachtung der unterschiedlichen Auswirkungen bei dem Wegfall eines Neurons wichtig, so dass hierfür entsprechende Maße entwickelt werden.

### 3.2.2 Ausfallwahrscheinlichkeit

Bei der Umsetzung eines RBF-Netzes und des LCNNs in analoge oder digitale Hardware können die zur Berechnung eingesetzten Bauelemente und Schaltungen ausfallen, so dass insgesamt die Berechnung des Ausgabewertes durch das Netz nicht mehr korrekt stattfindet. Dieser Ausfall einer Berechnung kann entweder als geänderter Parameter aufgefasst werden, dessen Einfluss in Abschnitt 3.2.3 näher betrachtet wird, oder der Ausfall wirkt sich nur auf einen bestimmten Teil aus, was als zusätzliches Rauschen angesehen werden kann.

Es wird nun die mittlere Zeit bis zum Auftreten eines fehlerhaften Ausgabewertes berechnet, wenn die einzelnen Elemente ausfallen können. Dabei werden in diesem Abschnitt die zur Laufzeit auftretenden dynamischen Fehler betrachtet. Die hier berechneten Zeiten und Ausfallraten können als Eingabe für die Berechnungen in Abschnitt 3.3.2 dienen.

Für die Berechnungen werden die folgenden Annahmen getroffen. Die Lebensdauern der einzelnen Komponenten, die zur Berechnung des Ausgabewertes benötigt werden, seien bekannt und können als Zufallsvariable  $\mathcal{X}_1, \dots, \mathcal{X}_\nu$  dargestellt werden, wobei  $\nu_i$  die Anzahl von benötigten Elementen bezeichnet.

Zuerst seien die Ausfallzeiten  $\mathcal{X}_i$  der einzelnen Elemente exponentialverteilt, d.h.

$$\mathcal{X}_i \propto e^{\lambda_i}. \quad (3.112)$$

Dabei wird in der Analyse angenommen, dass Multiplizierer und Dividierer sowie Addierer und Subtrahierer die gleichen Ausfallzeiten besitzen.

Der Ausgabewert des Systems ist genau dann nicht mehr korrekt, wenn ein Element des jeweiligen neuronalen Netzes ausfällt, d.h., für die Ausfallzeit  $U$  des Systems gilt

$$U = \min \{ \mathcal{X}_1, \dots, \mathcal{X}_\nu \}. \quad (3.113)$$

Gilt weiter die Annahme, dass die Zufallsvariablen  $\mathcal{X}_i$ , d.h., die Ausfallzeiten der Recheneinheiten, voneinander unabhängig sind, folgt für die Wahrscheinlichkeit, dass das LCNN länger als  $t$  Zeiteinheiten korrekt arbeitet

$$P(U > t) = P(\mathcal{X}_1 > t, \dots, \mathcal{X}_\nu > t) \quad (3.114)$$

$$= \prod_{i=1}^{\nu} P(\mathcal{X}_i > t) = \prod_{i=1}^{\nu} (1 - P(\mathcal{X}_i \leq t)) \quad (3.115)$$

$$= \prod_{i=1}^{\nu} (1 - F^{\mathcal{X}_i}(t)) \quad (3.116)$$

$$= \prod_{i=1}^{\nu_\sigma} (1 - F^{\mathcal{X}_i}(t)) \cdot \prod_{i=1}^{\nu_A} (1 - F^{\mathcal{X}_i}(t)) \cdot \prod_{i=1}^{\nu_M} (1 - F^{\mathcal{X}_i}(t)), \quad (3.117)$$

wobei  $\nu_\sigma$  die Anzahl der Sigmoiden,  $\nu_A$  die Anzahl der Addierer und  $\nu_M$  die Anzahl der Multiplizierer bezeichnet.

Die Betrachtungen können in analoger Weise auf ein RBF-Netz übertragen werden, wobei allerdings anstelle der Sigmoiden exp-Funktionen verwendet werden. Dabei können beide Berechnungsvarianten (vgl. Abbildung 3.4 und 3.5) verwendet werden, wobei hier nur die Variante I betrachtet wird, da bei dieser weniger exp-Funktionen und Multiplizierer benötigt werden. Da diese beiden Recheneinheiten komplexer als eine Addition sind [BK80], wird hier diese Form der Berechnung gewählt. Es wird angenommen, dass die Ausfallzeit umgekehrt proportional zu der benötigten Fläche oder der Komplexität einer Schaltung verläuft.

Da die einzelnen Zufallsvariablen der beiden neuronalen Netze exponentialverteilt sind, folgt somit für die Wahrscheinlichkeit bei einem LCNN, dass die Lebensdauer größer als  $t$  Zeiteinheiten ist

$$\begin{aligned} P(U > t) &= \prod_{i=1}^{\nu_\sigma} (1 - (1 - e^{-\lambda_\sigma t})) \cdot \prod_{i=1}^{\nu_A} (1 - (1 - e^{-\lambda_A t})) \cdot \prod_{i=1}^{\nu_M} (1 - (1 - e^{-\lambda_M t})) \\ &= \prod_{i=1}^{\nu_\sigma} e^{-\lambda_\sigma t} \cdot \prod_{i=1}^{\nu_A} e^{-\lambda_A t} \cdot \prod_{i=1}^{\nu_M} e^{-\lambda_M t} \end{aligned} \quad (3.118)$$

$$= e^{-(\lambda_\sigma \nu_\sigma + \lambda_A \nu_A + \lambda_M \nu_M)t}, \quad (3.119)$$

wobei bei dem LCNN zur Darstellung einer einzigen lokalen Funktion die Anzahl von Elementen aus (3.108), (3.109) und (3.110) benötigt wird (vgl. auch Tabelle 3.1). Die Ausfallraten der einzelnen Einheiten  $\lambda_\sigma, \lambda_A, \lambda_M$  sind von der verwendeten Technologie und der Architektur des jeweiligen Rechenelementes abhängig.

Somit folgt für die Verteilungsfunktion und die Dichtefunktion der Lebensdauer einer lokalen Funktion beim LCNN

$$F^U(t) = 1 - e^{-(\lambda_\sigma(2n+1) + \lambda_A(n^2+3n) + \lambda_M(n^2+1))t} \quad (3.120)$$

$$f^U(t) = \frac{(\lambda_\sigma(2n+1) + \lambda_A(n^2+3n) + \lambda_M(n^2+1)) \times e^{-(\lambda_\sigma(2n+1) + \lambda_A(n^2+3n) + \lambda_M(n^2+1))t}}{e^{-(\lambda_\sigma(2n+1) + \lambda_A(n^2+3n) + \lambda_M(n^2+1))t}} \quad (3.121)$$

und für den Erwartungswert der Lebensdauer folgt dann

$$E(U) = \frac{1}{\lambda_\sigma(2n+1) + \lambda_A(n^2+3n) + \lambda_M(n^2+1)}. \quad (3.122)$$

Analog verläuft die Analyse für ein RBF-Netz, wobei die Anzahl der Elemente aus Tabelle 3.1 verwendet werden. Hier gilt damit für die Verteilungsfunktion und die Dichtefunktion der Lebensdauer bei einem RBF

$$F^U(t) = 1 - e^{-(\lambda_e + \lambda_A(2n-1) + \lambda_M(n+2))t} \quad (3.123)$$

$$f^U(t) = (\lambda_e + \lambda_A(2n-1) + \lambda_M(n+2)) e^{-(\lambda_e + \lambda_A(2n-1) + \lambda_M(n+2))t} \quad (3.124)$$

und weiter für den Erwartungswert der Lebensdauer einer Gaußfunktion beim RBF

$$E(U) = \frac{1}{\lambda_e + \lambda_A(2n-1) + \lambda_M(n+2)}. \quad (3.125)$$

Ebenfalls können die Zufallsvariablen  $\mathcal{X}_i$  auch durch eine Weibull-Verteilung beschrieben werden [LXM03], welche zum einen die Exponentialverteilung für  $\alpha = 1$  beinhaltet und zum anderen oft in der Zuverlässigkeitstheorie angewendet wird [BSM05], da mit ihr die Ausfälle von Bauteilen flexibel beschrieben werden können. Somit gilt für die Zufallsvariablen

$$\mathcal{X}_i \propto \text{Wei}(\alpha_i, \beta_i) \quad (3.126)$$

und für die Verteilungsfunktion

$$F^{\mathcal{X}_i}(t) = 1 - e^{-\left(\frac{t}{\beta_i}\right)^{\alpha_i}}. \quad (3.127)$$

Unter diesen Annahmen folgt für die Verteilungsfunktion der Lebensdauer einer lokalen Funktion des LCNs in analoger Weise wie zuvor

$$F^U(t) = 1 - e^{-(2n+1)\left(\frac{t}{\beta_\sigma}\right)^{\alpha_\sigma}} \cdot e^{-(n^2+3n)\left(\frac{t}{\beta_A}\right)^{\alpha_A}} \cdot e^{-(n^2+1)\left(\frac{t}{\beta_M}\right)^{\alpha_M}} \quad (3.128)$$

bzw. für die Wahrscheinlichkeitsdichte

$$f^U(t) = \left( \frac{\alpha_\sigma(2n+1)\left(\frac{t}{\beta_\sigma}\right)^{\alpha_\sigma}}{t} + \frac{\alpha_A(n^2+3n)\left(\frac{t}{\beta_A}\right)^{\alpha_A}}{t} + \frac{\alpha_M(n^2+1)\left(\frac{t}{\beta_M}\right)^{\alpha_M}}{t} \right) \cdot e^{-(2n+1)\left(\frac{t}{\beta_\sigma}\right)^{\alpha_\sigma}} \cdot e^{-(n^2+3n)\left(\frac{t}{\beta_A}\right)^{\alpha_A}} \cdot e^{-(n^2+1)\left(\frac{t}{\beta_M}\right)^{\alpha_M}}, \quad (3.129)$$

wobei die Parameter  $\alpha_i, i \in \{\sigma, A, M\}$  und  $\beta_i, i \in \{\sigma, A, M\}$  die einzelnen Parameter der jeweiligen Rechenelemente bezeichnen.

Für den Fall  $\alpha_\sigma = \alpha_A = \alpha_M = \alpha$  folgt für den Erwartungswert

$$E(U) = \frac{1}{\sqrt[\alpha]{\frac{2n+1}{\beta_\sigma^\alpha} + \frac{n^2+3n}{\beta_A^\alpha} + \frac{n^2}{\beta_M^\alpha}}} \Gamma\left(1 + \frac{1}{\alpha}\right), \quad (3.130)$$

wobei  $\Gamma(x)$  die Gammafunktion<sup>4</sup> oder das Eulersche Integral zweiter Gattung bezeichnet und wie folgt definiert ist [BSM05, S. 455]

$$\Gamma(x) = \int_0^{\infty} e^{-t} t^{x-1} dt \quad x > 0. \quad (3.131)$$

Ähnliche Berechnungen können für eine Gaußfunktion bei einem RBF-Netz durchgeführt werden. Analog folgt dann für die Dichtefunktion der Zufallsvariablen  $U$

$$f^U(t) = \left( \frac{\alpha_e}{\beta_e^{\alpha_e}} t^{\alpha_e-1} + \frac{(2n-1)\alpha_A}{\beta_A^{\alpha_A}} t^{\alpha_A-1} + \frac{(n+2)\alpha_M}{\beta_M^{\alpha_M}} t^{\alpha_M-1} \right) \times e^{-\left(\left(\frac{t}{\beta_e}\right)^{\alpha_e} + (2n-1)\left(\frac{t}{\beta_A}\right)^{\alpha_A} + (n+2)\left(\frac{t}{\beta_M}\right)^{\alpha_M}\right)t}. \quad (3.132)$$

Auch hier gilt dann für den Erwartungswert der Lebensdauer einer Basisfunktion eines RBF-Netzes

$$E(U) = \frac{1}{\sqrt{\frac{1}{\beta_e^{\alpha_e}} + \frac{2n-1}{\beta_A^{\alpha_A}} + \frac{n+2}{\beta_M^{\alpha_M}}}} \Gamma\left(1 + \frac{1}{\alpha}\right). \quad (3.133)$$

Für den Vergleich der beiden Netze wird angenommen, dass die Ausfallraten der Elemente in beiden Netzen durch identische Zufallsvariablen beschrieben werden können, d.h., es gilt für den Parameter  $\lambda$  der Exponentialverteilung

$$\lambda_e = \lambda_{\sigma} \quad (\lambda_A)_{\text{RBF}} = (\lambda_A)_{\text{LCNN}} \quad (\lambda_M)_{\text{RBF}} = (\lambda_M)_{\text{LCNN}}. \quad (3.134)$$

Somit folgt für die Ausfallrate eines RBF-Netzes für eine Exponentialverteilung

$$\lambda_{\text{RBF}} = \lambda_e + (2n-1)\lambda_A + (n+2)\lambda_M \quad (3.135)$$

und für den Erwartungswert der Lebensdauer

$$E(U_{\text{RBF}}) = \frac{1}{\lambda_{\text{RBF}}} = \frac{1}{\lambda_e + (2n-1)\lambda_A + (n+2)\lambda_M} \quad (3.136)$$

sowie für die Varianz der Zufallsvariablen

$$\sigma^2 = \frac{1}{\lambda_{\text{RBF}}^2} = \frac{1}{(\lambda_e + (2n-1)\lambda_A + (n+2)\lambda_M)^2}. \quad (3.137)$$

Das LCNN besitzt analog eine Ausfallrate von

$$\lambda_{\text{LCNN}} = (2n+1)\lambda_{\sigma} + (n^2+3n)\lambda_A + (n^2+1)\lambda_M \quad (3.138)$$

<sup>4</sup> Mit der Gammafunktion kann der Begriff der Fakultät auf beliebige Zahlen  $x$ , auch komplexe Zahlen, erweitert werden.

und einen Erwartungswert

$$E(U_{\text{LCNN}}) = \frac{1}{(2n+1)\lambda_\sigma + (n^2+3n)\lambda_A + (n^2+1)\lambda_M}. \quad (3.139)$$

Die Varianz folgt analog.

Durch Vergleich der Ausfallraten des LCNNs (3.138) und des RBFs (3.135) folgt

$$\lambda_{\text{RBF}} < \lambda_{\text{LCNN}} \quad (3.140)$$

$$\Rightarrow E(U_{\text{RBF}}) > E(U_{\text{LCNN}}) \quad (3.141)$$

$$\mathcal{O}(\lambda_{\text{RBF}}) = n \quad \mathcal{O}(\lambda_{\text{LCNN}}) = n^2. \quad (3.142)$$

Somit besitzt die Funktionalität eines RBF-Netzes eine größere Zeitspanne als ein LCNN. Dies ist aufgrund der hohen Anzahl von Rechenoperationen beim LCNN offensichtlich.

Allerdings kann die Annahme  $\lambda_e = \lambda_\sigma$  nicht immer zutreffend sein. Um dennoch gleiche Ausfallraten zu erhalten muss gelten

$$\lambda_e = (2n+1)\lambda_\sigma + (n^2+n+1)\lambda_A + (n^2-n-1)\lambda_M. \quad (3.143)$$

Bei der Erfüllung von (3.143) sind beide Ausfallraten der Netze gleich groß. Gilt eine Ordnungsrelation zwischen  $\lambda_e$  und der rechten Seite der Gleichung, so ergibt sich eine höhere Lebensdauer für das RBF (<) oder das LCNN (>).

Die Betrachtungen können auf das ganze neuronale Netz ausgeweitet werden. Hier stellt dann der Ausfall einer Gaußfunktion oder einer lokalen Funktion eine Zufallsvariable dar, und in analoger Weise wie zuvor kann die Lebensdauer des Gesamtsystems mit  $m$  Basis- oder lokalen Funktionen bestimmt werden. Als Ausfallraten können dann die Ergebnisse aus (3.135) oder (3.138) verwendet werden.

Daneben kann der Ausfall von Gaußfunktionen im gesamten neuronalen Netz als Markov-Prozess dargestellt werden. Diese Betrachtung wird in Abschnitt 3.3 näher untersucht.

### 3.2.3 Auswirkungen auf neuronale Netze

In diesem Abschnitt werden die Auswirkungen von Fehlern und Defekten auf die Approximationseigenschaften der neuronalen Netze untersucht. Dafür werden die Parameter des Netzes als mögliche Fehlerquellen, die von einem Stuck-At-Fehler betroffen sind, angesehen.

## 3.2.3.1 RBF-Netze

Bei einem RBF-Netz können die Zentren, die Ausgangsgewichte und die Varianzen bedingt durch einen Ausfall oder Defekt unterschiedliche Werte annehmen. Dadurch produziert das Netz gegenüber der ursprünglichen Funktion einen Fehler, der möglichst gering für alle Situationen ausfallen soll.

Zuerst wird angenommen, dass die Ausgangsgewichte  $\alpha_i$  von dem Fehler oder Defekt betroffen sind. Um von der Implementierung unabhängige Aussagen treffen zu können, wird davon ausgegangen, dass jeder beliebige Wert  $\mu$  angenommen werden kann. Somit folgt für die Differenz zwischen der durch das ursprüngliche RBF-Netz beschriebenen Funktion und des fehlerhaften Netzes

$$\begin{aligned} f_m(\mathbf{x}) - \hat{f}_m(\mathbf{x}) &= \sum_{i=1}^m \alpha_i \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) - \sum_{i=1}^m \hat{\alpha}_i \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) \\ &= \sum_{i=1}^m (\alpha_i - \hat{\alpha}_i) \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right), \end{aligned} \quad (3.144)$$

wobei  $\hat{f}_m(\mathbf{x})$  die Funktion des gestörten Netzes beschreibt.

Im Allgemeinen können beliebig viele Neuronen von einem Ausfall betroffen sein, so dass sich ein großer Fehler ergeben kann. Ist nur ein einziges Ausgangsgewicht  $\alpha_k$  von einem Fehler betroffen, so verschwinden viele Summanden aus (3.144), und es folgt

$$f_m(\mathbf{x}) - \hat{f}_m(\mathbf{x}) = (\alpha_k - \mu) \cdot \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_k\|^2}{2\sigma_k^2}\right). \quad (3.145)$$

Tritt in einem Zentrum des RBF-Netzes ein Fehler auf, so führt dies zu einer unbeabsichtigten Verschiebung der Gaußfunktion im Raum. Allgemein können auch hier mehrere Fehler gleichzeitig auftreten. Ist die  $k$ -te Gaußfunktion von einer Verschiebung des Zentrums betroffen, und wird in diesem Zentrum der  $\nu$ -te Wert verändert, so führt dies zu einer Differenz der beiden Funktionen der Netze

$$\begin{aligned} f_m(\mathbf{x}) - \hat{f}_m(\mathbf{x}) &= \sum_{i=1}^m \alpha_i \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) - \sum_{i=1}^m \alpha_i \exp\left(-\frac{\|\mathbf{x} - \hat{\mathbf{c}}_i\|^2}{2\sigma_i^2}\right) \\ &= \alpha_k \left[ \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_k\|^2}{2\sigma_k^2}\right) - \exp\left(-\frac{\|\mathbf{x} - \hat{\mathbf{c}}_k\|^2}{2\sigma_k^2}\right) \right] \end{aligned} \quad (3.146)$$

$$= \alpha_k \exp\left(-\frac{\sum_{\substack{j=1 \\ j \neq \nu}}^n (x_j - c_{kj})^2}{2\sigma_k^2}\right) \times \quad (3.147)$$

$$\left[ \exp\left(-\frac{(x_\nu - x_{k\nu})^2}{2\sigma_k^2}\right) - \exp\left(-\frac{(x_\nu - \mu)^2}{2\sigma_k^2}\right) \right], \quad (3.148)$$

wobei  $c_{kj}$  den Eintrag  $j$  des Zentrums der Gaußfunktion  $k$  bezeichnet.

In analoger Weise können die Varianzen der Gaußfunktionen von dem Fehler oder Defekt betroffen sein, was zu einer Veränderung der Überdeckung des Raumes durch die jeweilige Gaußfunktion führt. Ist wiederum eine Varianz betroffen, so führt dies zu

$$\begin{aligned} f_m(\mathbf{x}) - \hat{f}_m(\mathbf{x}) &= \sum_{i=1}^m \alpha_i \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) - \sum_{i=1}^m \alpha_i \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\hat{\sigma}_i^2}\right) \\ &= \alpha_k \cdot \left[ \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_k\|^2}{2\sigma_k^2}\right) - \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_k\|^2}{2\mu^2}\right) \right]. \end{aligned} \quad (3.149)$$

Um Aussagen über die Fehlertoleranz eines RBF-Netzes gegenüber dem Auftauchen von Fehlern in seinen Parametern treffen zu können, wird der mittlere quadratische Fehler betrachtet, der aufgrund der geänderten Parameter und Struktur entsteht. Eine Strukturänderung kann sich durch den Ausfall einer Verbindung zu dem linearen Ausgangsneuron ergeben. So besitzt dieses Netz eine veränderte Struktur, da insgesamt weniger Basisfunktionen verwendet werden.

Dazu werden die Eingangsvektoren als Zufallsvektoren betrachtet, und es folgt für den mittleren quadratischen Fehler im Ausgang, wenn ein Ausgangsgewicht von einem Fehler oder Defekt betroffen ist

$$\text{mse}_\alpha = (\alpha_k - \mu)^2 \cdot E \left[ \underbrace{\exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_k\|^2}{\sigma_k^2}\right)}_{\leq 1} \right]. \quad (3.150)$$

Mit dem Mittelwertsatz der Integralrechnung [BSM05] kann der mittlere quadratische Fehler aus (3.150) abgeschätzt werden

$$\text{mse}_\alpha \leq (\alpha_k - \mu)^2. \quad (3.151)$$

Mit dem verallgemeinerten Mittelwertsatz kann der Erwartungswert zweier Funktionen  $f(x)$  und  $\varphi(x)$  über einem abgeschlossenen Intervall berechnet werden, wenn beide Funktionen stetig sind, und eine der beiden Funktionen ihr Vorzeichen in diesem Bereich nicht ändert. Sowohl die Gaußfunktion als auch die Wahrscheinlichkeitsdichte ändern ihr Vorzeichen nicht, und wird  $\varphi(x)$  gleich der Wahrscheinlichkeitsdichte gesetzt, so ergibt sich die Abschätzung nach (3.151). Diese Vorgehensweise ist für alle Verteilungen mit einer Dichtefunktion über einem abgeschlossenen Intervall gültig, d.h.

$$\int_a^b (\alpha_k - \mu)^2 e^{-\frac{(x-c_k)^2}{\sigma_i^2}} p_X(x) dx = (\alpha_k - \mu)^2 e^{-\frac{(\xi-c_k)^2}{\sigma_k^2}} \underbrace{\int_a^b p_X(x) dx}_{=1}. \quad (3.152)$$

Für Verteilungen, die über die gesamten reellen Zahlen definiert sind, wie z.B. die Gaußverteilung, kann der Mittelwertsatz der Integralrechnung nicht angewandt werden, da ein abgeschlossenes Intervall benötigt wird. Allerdings kann mit der Beziehung [Ste04b]:

$$\int_a^\infty f(x)dx \leq \int_a^\infty g(x)dx, \quad (3.153)$$

wobei  $f(x) \leq g(x)$  für alle  $x \in [a, \infty)$  gilt, die Abschätzung in (3.151) auch für Verteilungen über die kompletten reellen Zahlen bestimmt werden.

Es gilt dabei

$$e^{-\frac{(x-c_k)^2}{\sigma_k^2}} \cdot p_X(x) \leq p_X(x), \quad (3.154)$$

da  $p_X(x) \geq 0 \forall x \in \mathbb{R}$ . Somit folgt

$$\begin{aligned} \int_{-\infty}^\infty e^{-\frac{(x-c_k)^2}{\sigma_k^2}} p_X(x)dx &= \int_{-\infty}^a e^{-\frac{(x-c_k)^2}{\sigma_k^2}} p_X(x)dx + \int_a^\infty e^{-\frac{(x-c_k)^2}{\sigma_k^2}} p_X(x)dx \\ &\leq \int_a^\infty p_X(x)dx + \int_{-\infty}^a e^{-\frac{(x-c_k)^2}{\sigma_k^2}} p_X(x)dx \end{aligned} \quad (3.155)$$

$$= \int_a^\infty p_X(x)dx - \int_\infty^{-a} e^{-\frac{(x-c_k)^2}{\sigma_k^2}} p_X(-x)dx \quad (3.156)$$

$$= \int_a^\infty p_X(x)dx + \int_{-a}^\infty e^{-\frac{(x-c_k)^2}{\sigma_k^2}} p_X(-x)dx, \quad (3.157)$$

und da für  $x \in \mathbb{R}$  gilt  $\exp(-x^2) \leq 1$  folgt

$$\int_{-\infty}^\infty e^{-\frac{(x-c_k)^2}{\sigma_k^2}} p_X(x)dx \leq \int_a^\infty p_X(x)dx + \int_{-a}^\infty p_X(-x)dx \quad (3.158)$$

$$= \int_a^\infty p_X(x)dx - \int_\infty^{-a} p_X(-x)dx \quad (3.159)$$

$$= \int_{-\infty}^\infty p_X(x)dx = 1. \quad (3.160)$$

Damit ist der mittlere quadratische Fehler ohne Beschränkung der Gewichte bei einem Fehler oder Defekt der Gewichte nicht nach oben beschränkt. Zusätzlich ist dieser Fehler von dem gestörten Wert des Parameters abhängig, so dass dieser auch von der verwendeten Implementierung abhängig ist. Bei der analogen Realisierung können beliebige Werte  $\mu$  für den Fehler oder Defekt des Ausgangsgewichtes angenommen werden, wobei bei einer digitalen Realisierung sich quantisierte Werte ergeben. Im ersten Fall werden jedoch die Gewichte durch die Operationsbedingungen eingeschränkt [Raz00] und können nur innerhalb eines bestimmten Intervalls Werte annehmen, da z.B. die maximale Versorgungsspannung die Parameter nach oben beschränkt. Bei einer digitalen Implementierung ergibt sich aufgrund der begrenzt-

ten Genauigkeit der Zahlendarstellung und des endlichen Speichers ein maximaler Wert der Gewichte

$$|\alpha_k| \leq B \quad \forall k = 1 \dots n. \quad (3.161)$$

Neben den Gewichten wird auch der Fehler oder Defekt mit der gleichen Begründung durch eine obere Schranke beschränkt, so dass (3.151) mit diesen Randbedingungen weiter abgeschätzt werden kann

$$\text{mse}_\alpha \leq (B + |\mu|)^2. \quad (3.162)$$

So kann eine obere Grenze für den mittleren quadratischen Fehler bei einem Ausfall oder einem Defekt der Gewichte angegeben werden, wobei die obere Grenze von der verwendeten Realisierung abhängig ist.

In der gleichen Art können (3.148) und (3.149) ausgewertet werden, was zu

$$\begin{aligned} \text{mse}_{c_k} &= \alpha_k^2 E \left[ \left( \exp \left( -\frac{\|\mathbf{x} - \mathbf{c}_k\|^2}{2\sigma_k^2} \right) - \exp \left( -\frac{\|\mathbf{x} - \hat{\mathbf{c}}_k\|^2}{2\sigma_k^2} \right) \right)^2 \right] \\ &\leq \alpha_k^2 \end{aligned} \quad (3.163)$$

$$\begin{aligned} \text{mse}_\sigma &= \alpha_k^2 E \left[ \left( \exp \left( -\frac{\|\mathbf{x} - \mathbf{c}_k\|^2}{2\sigma_k^2} \right) - \exp \left( -\frac{\|\mathbf{x} - \mathbf{c}_k\|^2}{2\mu^2} \right) \right)^2 \right] \\ &\leq \alpha_k^2 \end{aligned} \quad (3.164)$$

führt.

Auch hier folgt aus (3.163) und (3.164), dass der mittlere quadratische Fehler im Ausgang keine obere Grenze bei einem Defekt oder Fehler in den Zentren oder Varianzen besitzt. Mit der gleichen Begründung wie zuvor sind jedoch die Ausgangsgewichte durch die technischen Randbedingungen beschränkt, was zu einer oberen Grenze führt, die bei einem Fehler in den Varianzen und Zentren unabhängig von der verwendeten Realisierung ist. Somit folgt für diese obere Grenze

$$\text{mse} \leq B^2. \quad (3.165)$$

**Bemerkung.** Aufgrund der linearen Eigenschaft des Ausgangsneurons ergibt sich bei mehreren Ausfällen oder Defekten der Parameter eine lineare Überlagerung der einzelnen Fehler. Die obere Grenze wird dann durch die Summe der jeweiligen betroffenen Neuronen bestimmt.

### 3.2.3.2 Basisfunktionen mit Produkt-Stabilisatoren

Wie zuvor für die radialen Basisfunktionen kann die Analyse eines Fehlers oder Defektes für die Basisfunktionen mit Produkt-Stabilisatoren vorgenommen werden.

Auch hier wird angenommen, dass die Parameter des Netzes durch einen einzelnen Fehler oder Defekt betroffen sind, und dass der Parameter den Wert  $\mu$  annimmt.

Ist bei diesem Basisfunktionsnetzwerk ein Ausgangsgewicht betroffen, so folgt für die Differenz der Funktionen der beiden Netze

$$f_m(\mathbf{x}) - \hat{f}_m(\mathbf{x}) = \sum_{i=1}^m \alpha_i e^{-\|\mathbf{x}-\mathbf{c}_i\|_1} - \sum_{i=1}^m \hat{\alpha}_i e^{-\|\mathbf{x}-\mathbf{c}_i\|_1}, \quad (3.166)$$

wobei  $\hat{f}_m(\mathbf{x})$  die Funktion des gestörten Netzes beschreibt und  $\hat{\alpha}_i$  den veränderten Gewichten entspricht.

Falls nur ein Ausgangsgewicht von diesem Fehler bzw. Defekt betroffen ist, so vereinfacht sich die Gleichung zu

$$f_m(\mathbf{x}) - \hat{f}_m(\mathbf{x}) = (\alpha_k - \mu) e^{-\|\mathbf{x}-\mathbf{c}_k\|_1}. \quad (3.167)$$

Mit (3.167) kann wie schon zuvor der mittlere quadratische Fehler im Ausgang abgeschätzt werden, und dies führt zu

$$\text{mse}_\alpha = (\alpha_k - \mu)^2 \cdot \underbrace{E[e^{-2\|\mathbf{x}-\mathbf{c}_k\|_1}]}_{\leq 1}. \quad (3.168)$$

In (3.168) kann der Mittelwertsatz der Integralrechnung [BSM05] angewendet werden, womit der mittlere Fehler nach oben abgeschätzt werden kann

$$\text{mse}_\alpha \leq (\alpha_k - \mu)^2. \quad (3.169)$$

Man erkennt an (3.169), dass der mittlere quadratische Fehler nicht nach oben beschränkt ist und von der jeweiligen Realisierungsform abhängig ist. Mit der gleichen Begründung wie zuvor bei den radialen Basisfunktionen sind die Gewichte und der Wert des Fehlers oder Defektes durch die Umgebungsbedingungen nach oben durch eine Schranke begrenzt. Für den Fall beschränkter Gewichte ( $|\alpha| \leq B$ ) kann (3.169) weiter abgeschätzt werden

$$\text{mse}_\alpha \leq (B + |\mu|)^2. \quad (3.170)$$

In der gleichen Art können auch die Fehler in den Zentren der Basisfunktionen behandelt werden, falls der  $\nu$ -te Eintrag eines Zentrums von einem Fehler betroffen ist. Dies führt zu einer Differenz der beiden Netze von

$$\begin{aligned} f_m(\mathbf{x}) - \hat{f}_m(\mathbf{x}) &= \sum_{i=1}^m \alpha_i e^{-\|\mathbf{x}-\mathbf{c}_i\|_1} - \sum_{i=1}^m \alpha_i e^{-\|\mathbf{x}-\hat{\mathbf{c}}_i\|_1} \\ &= \alpha_k [e^{-\|\mathbf{x}-\mathbf{c}_k\|_1} - e^{-\|\mathbf{x}-\hat{\mathbf{c}}_k\|_1}] \end{aligned} \quad (3.171)$$

$$= \alpha_k \exp\left(-\sum_{\substack{j=1 \\ j \neq \nu}}^n |x_j - c_{kj}|\right) \times \quad (3.172)$$

$$[e^{-|x_\nu - c_{k\nu}|} - e^{-|x_\nu - \mu|}]. \quad (3.173)$$

Mit (3.171) kann der mittlere quadratische Fehler bestimmt werden

$$\text{mse}_{\mathbf{c}_k} = \alpha_k^2 E \left[ \underbrace{(e^{-\|\mathbf{x}-\mathbf{c}_k\|_1} - e^{-\|\mathbf{x}-\hat{\mathbf{c}}_k\|_1})^2}_{\leq 1} \right] \quad (3.174)$$

$$\leq \alpha_k^2. \quad (3.175)$$

Aus (3.175) kann ebenfalls gefolgert werden, dass der Fehler nicht nach oben begrenzt ist, solange die Gewichte frei gewählt werden können. Im Gegensatz zu einem Fehler in den Ausgangsgewichten aus (3.169) hängt dieser Fehler jedoch nicht von der Realisierungsform ab. Für nach oben beschränkte Gewichte  $|\alpha| \leq B$  kann der Fehler nach oben durch

$$\text{mse}_{\mathbf{c}_k} \leq B^2 \quad (3.176)$$

abgeschätzt werden.

### 3.2.3.3 LCNN

Die Analyse kann ebenfalls für das LCNN durchgeführt werden. Bei dem LCNN können neben den Ausgangsgewichten, die Gewichte der Ridge-Funktionen, die Zentren und die Steilheiten der Sigmoiden von einem Defekt oder Fehler betroffen sein. In ähnlicher Weise folgt hier für eine Störung der Ausgangsgewichte

$$y_m(\mathbf{x}) - \hat{y}_m(\mathbf{x}) = \sum_{\nu=1}^m \alpha_\nu \mathcal{L}(\mathbf{w}_\nu, \mathbf{r}, \mathbf{x}) - \sum_{\nu=1}^m \hat{\alpha}_\nu \mathcal{L}(\mathbf{w}_\nu, \mathbf{r}, \mathbf{x}) \quad (3.177)$$

$$= (\alpha_k - \mu) \mathcal{L}(\mathbf{W}_k, \mathbf{r}_k, \mathbf{x}) \leq \alpha_k - \mu, \quad (3.178)$$

wobei angenommen wird, dass nur ein Gewicht von einem Fehler betroffen ist.

In (3.178) wurde ausgenutzt, dass der Ausgangssigmoid  $\sigma_o$  des LCNNs durch Eins nach oben beschränkt ist. So folgt mit (3.178) für den maximalen Fehler im Ausgang

$$\text{mse}_\alpha \leq (\alpha_k - \mu)^2 \leq (B + |\mu|)^2. \quad (3.179)$$

Die obere Grenze in (3.179) ergibt sich aufgrund der technologischen Randbedingung der Realisierung, wie schon bei den beiden Basisfunktionsnetzen zuvor.

Bei Fehlern in den Gewichten der Ridge-Funktionen oder Defekten in den Zentren ergibt sich aufgrund der oberen Schranke des Ausgangssigmoids des LCNNs ebenfalls wie bei den Basisfunktionsnetzwerken ein maximaler, mittlerer quadratischer Fehler von

$$\text{mse} \leq \alpha_k^2 \leq B^2. \quad (3.180)$$

Neben den Ausgangsgewichten, den Gewichten der Ridge-Funktionen und den Zentren der lokalen Funktionen besitzt das LCNN als weitere Parameter die Steigungen der Sigmoide im Ausgang und in den Ridge-Funktionen. Ein Fehler oder Defekt innerhalb dieser Parameter wird aufgrund des Ausgangssigmoids begrenzt. So ergibt sich für diese Parameter die gleiche Abschätzung wie in (3.180).

### 3.3 Ausfallwahrscheinlichkeit

In diesem Abschnitt sollen die Auswirkungen aufgrund erhöhter Unzuverlässigkeit der Bauelemente untersucht werden. Die so resultierenden Fehler lassen sich in drei Kategorien einteilen (vgl. Abschnitt 2.3.4), wobei in diesem Abschnitt die Effekte von Defekten bei der Halbleiterproduktion und der transienten Fehler auf den Entwurf eines Systems analysiert werden.

#### 3.3.1 Statische Defekte

Bei der Fertigung der Halbleiterbauelemente treten Defekte, z.B. durch falsche Maskenausrichtung oder Überätzung, auf, welche die Ausbeute an funktionsfähigen Schaltungen auf einer Halbleiterscheibe reduzieren. Allerdings resultieren nicht alle der produktionsbedingten Defekte in einen Ausfall eines Bauelementes, sondern dieser Vorgang hängt zudem von der Position des Defektes, seiner Größe, dem Layout der Schaltung und der Integrationsdichte auf dem Baustein ab. Eine Bestimmung dieses Zusammenhangs findet sich in [Cun90].

Zur Berechnung der Ausbeute eines gefertigten Bausteins oder der Zuverlässigkeit eines neuronalen Netzes kann die Binomialverteilung bzw. die Poissonverteilung angewendet werden. Zum einen besitzen diese einen einfachen mathematischen Charakter, und zum anderen stimmen die Ergebnisse dieser Modelle sehr gut mit

den beobachteten Verteilungsfunktionen überein [Cun90]. Somit ergibt sich für die Wahrscheinlichkeit, dass  $k$  Fehler aufgrund von Defekten bei einem neuronalen Netz mit insgesamt  $m$  Neuronen auftauchen zu

$$P(X = k) = \begin{cases} \binom{m}{k} \left(\frac{\lambda}{m}\right)^k \left(1 - \frac{\lambda}{m}\right)^{m-k} & \text{falls } x = 0, 1, \dots, m \\ 0 & \text{sonst} \end{cases}. \quad (3.181)$$

Dabei bezeichnet  $\lambda/m$  die Auftretswahrscheinlichkeit eines Fehlers, und  $\lambda$  erfasst die mittlere Anzahl von Fehlern in dem neuronalen Netz. Diese Verteilung strebt für  $m \rightarrow \infty$  gegen die Poissonverteilung

$$P(X = k) = \frac{e^{-\lambda} \lambda^k}{k!}. \quad (3.182)$$

Damit das neuronale Netz korrekt arbeitet, muss die Wahrscheinlichkeit<sup>5</sup> für  $k = 0$  bestimmt werden. Die Auftretswahrscheinlichkeit eines Fehlers ergibt sich aus der Anzahl von Defekten, die bei der Fertigung des neuronalen Netzes auftauchen, sowie der Wahrscheinlichkeit  $\Theta$ , dass dieser Defekt auch in einen Fehler resultiert. Bei der Annahme einer Gleichverteilung der Defekte über das neuronale Netz folgt für die Auftretswahrscheinlichkeit

$$\lambda/m = \frac{A_{nn} \cdot d \cdot \Theta}{m}, \quad (3.183)$$

wobei  $A_{nn}$  die Fläche des neuronalen Netzes und  $d$  die Dichte der Defekte beschreibt.

Diese Annahmen führen zu pessimistischen Ergebnissen, da Defekte aufgrund fehlerhafter Fertigung die Tendenz zu einer Bündelung zeigen. So führt ein systematischer Fehler bei der Fertigung zu einer Häufung von Defekten an bestimmten Positionen. Für genauere Ergebnisse kann  $\lambda$  selbst als Zufallsvariable aufgefasst werden und so Korrelationen zwischen einer Bündelung von Fehlern und den Neuronen erfasst werden. Für genauere Informationen sei auf [Cun90] und die Referenzen innerhalb der Veröffentlichung verwiesen. Die dort beschriebenen Formen werden in dem Fahrplan der Halbleiterindustrie genutzt [Sem05].

Somit ergibt sich bei  $N$  exakten Kopien eines Neurons die Wahrscheinlichkeit, dass mindestens eine Basisfunktion korrekt arbeitet und somit die Funktionsweise eines Neurons sichergestellt werden kann, zu

$$P(E_k) = \sum_{i=k}^N \binom{N}{i} (e^{-\lambda})^i (1 - e^{-\lambda})^{N-i}, \quad (3.184)$$

wobei  $E_k$  das Ereignis von mindestens  $k$  korrekt arbeitenden Neurons bezeichnet.

<sup>5</sup> Es wird bei den Vergleichen davon ausgegangen, dass die Schaltungen keine internen Mechanismen zur Kompensierung von Defekten besitzen. Somit führt ein einzelner Defekt zu dem Ausfall der Schaltung.

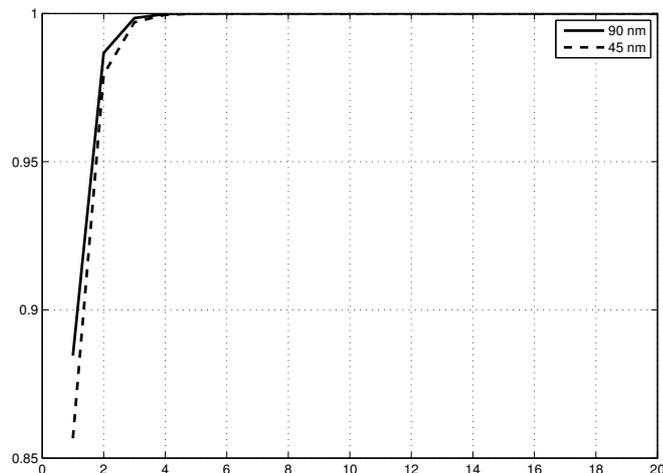


Abbildung 3.8: Ausbeute bei mehreren Kopien

Abbildung 3.8 zeigt die Wahrscheinlichkeit mindestens eines korrekt arbeitenden Neurons nach (3.184) gegenüber der Anzahl von redundanten Kopien. Für die Parameter der Verteilung wurden die entsprechenden Werte aus [Sem05] verwendet, wobei für die Bausteinfläche die dort angegebene Fläche für eine Ausbeute von 75 % gewählt wurde. Der Parameter  $\Theta$  wurde zu  $\Theta = 0,5$  gesetzt. Es zeigt sich, dass für eine zuverlässige Funktionalität des neuronalen Netzes aufgrund von Defekten bei der Fertigung nur eine geringe Anzahl zusätzlicher Neuronen vorgesehen werden muss. Der Unterschied der Wahrscheinlichkeiten für die dargestellten Technologiegenerationen fällt sehr gering aus, so dass vermutet werden kann, das Problem zahlreicher Defekte bei der Fertigung würde nicht bestehen. Allerdings handelt es sich bei dem Fahrplan um angestrebte Ziele, so dass nur eine begrenzte Anzahl von Defekten erlaubt wird, um eine gewünschte Ausbeute zu erzielen. Innerhalb des Fahrplans wird von einer gewünschten Ausbeute auf die maximale Defektdichte geschlossen.

### 3.3.2 Dynamische Fehler

Der Ausfall von Elementen in einem System, speziell der Ausfall von Neuronen in einem neuronalen Netz, kann durch einen stochastischen Prozess  $X(t) = (X_t, t \in T)$  beschrieben werden. Ein stochastischer Prozess  $X(t) = (X_t, t \in \mathbb{N}_0)$  mit abzählbarem Zustandsraum wird als Markov-Prozess bezeichnet. Für einen zeitkontinuierlichen Markov-Prozess muss gelten

$$P(X(s+t) = j | X(s) = i, X(u), 0 \leq u \leq s) = P(X(s+t) = j | X(s) = i), \quad (3.185)$$

d.h. der Übergang in den nächsten Zustand ist nur von dem aktuellen Zustand abhängig, und nicht wie dieser erreicht worden ist. Weiter werden nur homogene Prozesse betrachtet, und für einen homogenen Prozess gilt

$$P(X(s+t) = j | X(s) = i) = P(X(t) = j | X(0) = i). \quad (3.186)$$

Dabei kann der Prozess durch eine Übergangsmatrix  $\mathbf{P}(t)$  sowie durch eine Ratenmatrix  $\mathbf{R}$  oder eine Generatormatrix  $\mathbf{Q}$  beschrieben werden. Die letzten beiden Beschreibungsformen können angewendet werden, da der Markov-Prozess gedächtnislos ist und deshalb die Aufenthaltsdauer in einem Zustand exponentialverteilt ist. Für weitere Eigenschaften und Möglichkeiten der Beschreibung sei auf die Literatur verwiesen [Cas93].

Unter der Annahme, dass die Zeit bis zu einem Ausfall eines Neurons exponentialverteilt ist, kann dieses Verhalten mit Hilfe eines Markov-Prozesses modelliert werden. Wie in Abschnitt 3.2 gezeigt wurde, ist die Lebensdauer einer Gaußfunktion bzw. einer lokalen Funktion exponentialverteilt. So ist auch der Aufenthalt in einem Zustand des Systems exponentialverteilt, wenn der Zustand die Anzahl der zu einem Zeitpunkt  $t$  fehlerhaften Neuronen bezeichnet.

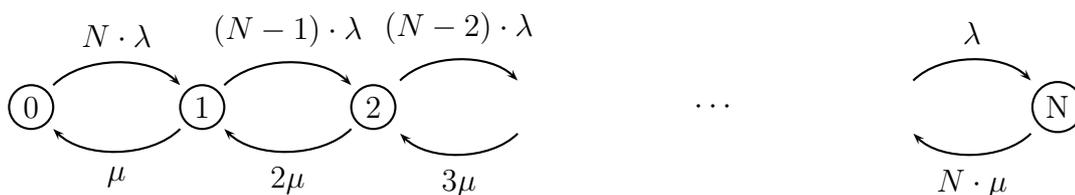
Zusätzlich wird angenommen, dass die transienten Fehler nach einer bestimmten Zeit wieder verschwinden und so die Neuronen wieder korrekt arbeiten können. Diese Zeitspanne sei ebenfalls exponentialverteilt. Somit ergibt sich der in Abbildung 3.9 dargestellte Markov-Prozess für ein Netz mit  $N$  Basisfunktionen<sup>6</sup>. Die gleiche Beschreibung des Markov-Prozesses ergibt sich für den Fall, dass ein einzelnes Neuron durch  $N$  Kopien ersetzt wird, wobei nur der Prozess bzgl. des Ausfalls dieses einen Neurons betrachtet wird.

Eine Topologieänderung kann den mittleren quadratischen Fehler im Ausgang aufgrund des Ausfalles von Neuronen reduzieren. Durch Einfügen von redundanten Neuronen kann die Approximation der Ausgangswerte über einem Bereich des Definitionsbereiches von mehreren Neuronen übernommen werden, und im Falle eines Ausfalls entsteht ein reduzierter Fehler (vgl. Abschnitt 3.2).

Hier können zwei wesentliche Vorgehensweisen unterschieden werden [ER06a]. Ein Neuron des Netzes kann durch  $N$  exakte Kopien ersetzt werden, d.h., bei den Basisfunktionen werden identische Zentren und Varianzen verwendet, während bei einem LCNN die gleichen Gewichtsmatrizen und Zentren eingesetzt werden. Die erste Möglichkeit besteht darin, die Ausgangsgewichte dieser Neuronen anzupas-

---

<sup>6</sup> Hier werden  $N$  Neuronen verwendet, um die Auswirkungen des Einfügens zusätzlicher Neuronen zu erfassen. Die Aussagen über den stationären Zustand können jedoch auf ein entsprechend dimensioniertes neuronales Netz übertragen werden.



**Abbildung 3.9:** Markov-Prozess des Ausfalls von Neuronen

sen, so dass jedes Neuron zu einem Ausgangswert einen reduzierten Beitrag leistet. Hier folgt für die neuen Ausgangsgewichte

$$\hat{\alpha}_k = \frac{\alpha_k}{N},$$

wobei  $\hat{\alpha}_k$  die neuen Ausgangsgewichte der  $N$  Neuronen und  $\alpha_k$  das Gewicht des ursprünglichen Neurons bezeichnen.

Daneben können die Gewichte der eingefügten Neuronen auch unverändert bleiben. Um den korrekten Wert des ursprünglichen Neurons auf einen Eingangsvektor liefern zu können, kann eine Mehrheitsentscheidung dieser Neuronen angewendet werden, wobei allerdings eine ungerade Anzahl  $N$  von Neuronen in das Netz eingefügt werden muss. Dieses Vorgehen entspricht einem Wiederholungskode in der Kodierungstheorie [HP03] oder wird als *N-tuple modular redundancy* (NMR) bezeichnet [HJ03a].

Statistisch gesehen wird bei dem ersten Verfahren der Erwartungswert der Ausgabe aller Neuronen bestimmt, während das zweite Verfahren verschiedene Klassen<sup>7</sup> für den Ausgangswert einsetzt, und die Klasse mit den meisten Treffern ausgewählt wird.

Der Markov-Prozess aus Abbildung 3.9 wird durch die Generatormatrix  $\mathbf{Q}$

$$\mathbf{Q} = \begin{pmatrix} -N \cdot \lambda & N \cdot \lambda & \dots & 0 \\ \mu & -(\mu + (N-1)\lambda) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & N \cdot \mu & -N \cdot \mu \end{pmatrix} \quad (3.187)$$

charakterisiert, wobei diese Matrix eine  $(N+1) \times (N+1)$  Matrix darstellt.

Der Zustandsvektor  $\boldsymbol{\pi}(t)$  wird durch die Differentialgleichung

$$\frac{\partial \boldsymbol{\pi}(t)}{\partial t} = \boldsymbol{\pi}(t) \cdot \mathbf{Q} \quad (3.188)$$

<sup>7</sup> Für den Fall kontinuierlicher Werte können Intervalle eingesetzt werden, und das Intervall, in dem die meisten Funktionswerte liegen, stellt die Ausgabe des Neurons dar.

beschrieben, und man erhält die stationäre Verteilung für

$$\boldsymbol{\pi}_S \mathbf{Q} = 0 \quad (3.189)$$

unter der Normierungsbedingung

$$\sum_{i=1}^N \pi_{S,i} = 1. \quad (3.190)$$

Dies ist gleichbedeutend mit dem Eigenwert der Norm  $\|\boldsymbol{\pi}_S\|_1 = 1$  zum Eigenwert  $\lambda = 0$  der Matrix  $\mathbf{Q}^T$ . Somit ergibt sich der stationäre Zustand des Markov-Prozesses zu

$$\boldsymbol{\pi}_S^T = \frac{1}{\left(1 + \frac{\lambda}{\mu}\right)^N} \begin{pmatrix} 1 \\ \binom{N}{1} \frac{\lambda}{\mu} \\ \vdots \\ \binom{N}{N} \frac{\lambda^N}{\mu^N} \end{pmatrix}. \quad (3.191)$$

Aus (3.191) können die Wahrscheinlichkeiten für den Aufenthalt in einem Zustand für den stationären Zustand abgelesen werden. Wird gefordert, dass das neuronale Netz mit einer Wahrscheinlichkeit  $P_S$  kein fehlerhaftes Element im stationären Zustand enthalten soll, so muss folgende Bedingung erfüllt sein

$$\frac{\lambda}{\mu} \leq \sqrt[N]{\frac{1}{P_S}} - 1. \quad (3.192)$$

Wie man an (3.192) erkennen kann, müssen die Elemente sehr zuverlässig sein, wenn man eine hohe Wahrscheinlichkeit für eine Funktionalität im stationären Zustand erreichen möchte. Für  $P_S \rightarrow 1$  folgt, dass  $\mu \rightarrow \infty$  oder  $\lambda \rightarrow 0$  gelten muss.

### 3.3.2.1 Stationäre Kosten

Die Betrachtungen für die Generatormatrix (3.187) und den stationären Zustand (3.191) gelten unabhängig von der Größe des Netzes. Für die Bestimmung der stationären Kosten wird jedoch angenommen, dass das ursprüngliche Netz nur aus einer Basisfunktion bzw. einer einzigen lokalen Funktion besteht. Diese wird dann durch mehrere Kopien ersetzt, und die stationären des neuen, erweiterten Netzes werden in Abhängigkeit der verwendeten Methode betrachtet.

Eine Analyse eines allgemeinen Netzes, bei dem ein bestimmtes Neuron durch mehrere Duplikate ersetzt wird, ist hier in Bezug auf die Berechnung des mittleren quadratischen Fehlers zu aufwendig. Bei dem vereinfachten Netz kann der Fehler sehr einfach bestimmt werden, da alle Neuronen innerhalb des Netzes gleiche Zen-

tren und Ausgangsgewichte<sup>8</sup> besitzen. Für das allgemeine Netz müssten zur Bestimmung des mittleren quadratischen Fehlers im stationären Zustand zusätzlich auch Mischterme mit anderen Neuronen berechnet werden, was zu einer Auswertung von

$$\sum_{k=1}^m \binom{m}{k} \binom{k+1}{2} = \frac{1}{8} 2^m (m+3)m = \mathcal{O}(m^2 2^m) \quad (3.193)$$

zusätzlichen Mischtermen führen würde. Deshalb wird für die Analyse der stationären Kosten das vereinfachte Netz verwendet.

Um die Kosten zu berechnen, die sich innerhalb dieses Prozesses ergeben, wird als Kostenmaß der mittlere quadratische Fehler verwendet, der sich beim Aufenthalt in einem Zustand ergibt. Für das Kopieren von Neuronen ergibt sich eine  $(N+1) \times (N+1)$  Kostenmatrix

$$U_c = \begin{pmatrix} 0 & \text{mse}_1 & 0 & \dots & 0 \\ 0 & 0 & \text{mse}_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \text{mse}_N \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}, \quad (3.194)$$

wobei  $\text{mse}_i$  den mittleren quadratischen Fehler bei dem Ausfall von  $i$  kopierten Neuronen bezeichnet. Da beim Kopieren der Neuronen nur die Ausgangsgewichte angepasst werden, kann dieser berechnet werden

$$\text{mse}_i = \left( \frac{i}{N} \right)^2 \cdot \text{mse}, \quad (3.195)$$

und  $\text{mse}$  bezeichnet den mittleren quadratischen Fehler des ursprünglichen Netzes.

Neben dem Einfügen von Duplikaten und dem anschließenden Anpassen der Ausgangsgewichte lässt sich mit den eingefügten Kopien eine Mehrheitsentscheidung des Ausgangswertes erreichen.

Für die Methode mittels Mehrheitsentscheidung ergeben sich eine gleiche Generatormatrix und somit eine gleiche stationäre Verteilung des Prozesses. Allerdings verändert sich die Kostenmatrix, da der mittlere quadratische Fehler entsteht, sobald mehr Neuronen ausgefallen sind als korrekt arbeiten. Für diesen Fall ergibt sich der komplette mittlere quadratische Fehler und nicht wie zuvor nur ein Teil davon. Somit kann die Kostenmatrix durch

$$(U_m)_{ij} = \begin{cases} \text{mse} & \text{für } i = \frac{N-1}{2}, \dots, N-1 \wedge j = i+1 \\ 0 & \text{sonst} \end{cases} \quad (3.196)$$

<sup>8</sup> In Abhängigkeit der verwendeten Methode können die Ausgangsgewichte auch verändert werden. Der stationäre Fehler kann aber sehr einfach für beide Verfahren bestimmt werden.

angegeben werden.

Die entstehenden Kosten, d.h. der mittlere quadratische Fehler im Ausgang, sind unabhängig von der Verweildauer in einem Zustand. Wenn eine bestimmte Anzahl von Neuronen ausgefallen ist, so hängt der sich einstellende mittlere quadratische Fehler nur davon ab, welche Neuronen von einem Ausfall betroffen sind. Diese Kosten können somit einem Übergang *in* einen Zustand hinein oder *aus* einem Zustand heraus zugeordnet werden. Für beide Vorgehensweisen ergeben sich gleiche stationäre Kosten, da eine asymptotische Betrachtung vorgenommen wird.

Mit Hilfe der Kostenmatrizen und der stationären Verteilung des Prozesses kann der sich einstellende mittlere quadratische Fehler beider Methoden berechnet werden, und es ergibt sich für die Kosten beim Kopieren von Neuronen

$$C_c = \frac{\lambda\mu(N^2\lambda^2 + 3\lambda\mu N - \lambda\mu + \mu^2)}{N(\lambda + \mu)^3} \text{mse} \quad (3.197)$$

und für die stationären Kosten bei der Mehrheitsentscheidung

$$C_m = \frac{1}{\left(1 + \frac{\lambda}{\mu}\right)^N} \sum_{i=\frac{N-1}{2}}^N \binom{N}{i} \frac{\lambda^i}{\mu^i} (N-i)\lambda \cdot \text{mse}. \quad (3.198)$$

Eine weitere Möglichkeit besteht darin, dass ein Neuron durch  $N$  exakte Kopien ersetzt wird und beim Abrufen zufällig eines dieser Kopien ausgewählt wird. Jedoch ist diese Methode nur für statische Fehler anwendbar. Für die zu erwartenden Kosten ergibt sich

$$E\{C\} = \sum_{i=1}^N C_i P(A_i), \quad (3.199)$$

wobei  $P(A_i)$  die Wahrscheinlichkeit der Auswahl des  $i$ -ten Neurons und  $C_i$  die entsprechenden Kosten bezeichnet. Dabei wird angenommen, dass für alle Neuronen die gleichen Exponentialverteilungen zu Grunde liegen. Somit ergibt sich für alle Neuronen identische Kosten, und somit folgt

$$E\{C\} = \sum_{i=1}^N C_i P(A_i) = C \sum_{i=1}^N P(A_i) = C. \quad (3.200)$$

Damit sind die sich einstellenden Kosten gleich dem ursprünglichen mittleren quadratischen Fehler, und es ergibt sich kein Vorteil dieser Methode gegenüber einem einzelnen Neuron.

Für das ursprüngliche Netz können die entsprechenden Betrachtungen für die stationären Kosten durchgeführt werden. Der Markov-Prozess aus Abbildung 3.9 reduziert sich auf zwei Zustände ('defekt' und 'fehlerfrei') für den Fall  $N = 1$ . Damit

ergibt sich mit der entsprechenden Generatormatrix und dem stationären Zustand Kosten von

$$C_{\text{orig}} = \frac{\mu\lambda}{\mu + \lambda} \text{mse}. \quad (3.201)$$

Ein Vergleich der Kosten aus (3.197) und (3.201) liefert, da  $\lambda > 0$  und  $\mu > 0$

$$C_{\text{orig}} < C_c \quad (3.202)$$

$$\Rightarrow 1 < \frac{N^2\lambda^2 + 3\lambda\mu N - \lambda\mu + \mu^2}{N(\mu + \lambda)^2}. \quad (3.203)$$

Die Ungleichung (3.203) beschreibt die Bedingung, für die das Einfügen der Neuronenkopien einen größeren mittleren quadratischen Fehler produziert als durch das ursprüngliche Netz

$$\lambda > \frac{\mu}{2N} \left( \sqrt{1 + 4N} - 1 \right). \quad (3.204)$$

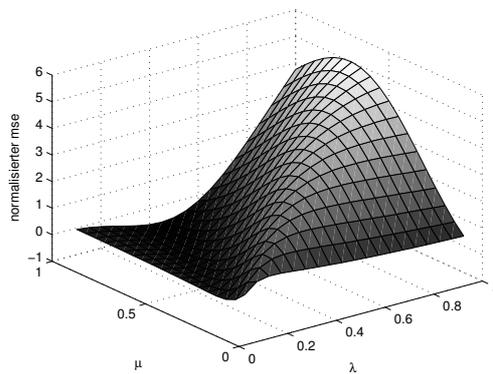
Abbildung 3.10 zeigt einen Vergleich der beiden Methoden des zusätzlichen Einfügens und der Mehrheitsentscheidung untereinander. In der Abbildung 3.10(a) ist die Differenz der stationären Kosten  $C_m - C_c$  dargestellt, so dass für positive Werte die Mehrheitsentscheidung höhere Kosten produziert. Bei negativen Werten ist das simple Einfügen der Mehrheitsentscheidung unterlegen, da größere Kosten entstehen.

Man kann deutlich erkennen, dass die Bestimmung der Mehrheit des Ausgangswertes dem einfachen Kopieren in den meisten Fällen der Reparatur- und Ausfallraten unterlegen ist. Welche Methode einen Vorteil gegenüber der jeweils anderen hat, hängt von der Ausfallrate  $\lambda$  und der Reparaturrate  $\mu$  ab. Sind diese bekannt, so kann die geeignete Technik ausgewählt werden. Weiter ist nur eine geringe Anzahl von Neuronen notwendig, um den mittleren quadratischen Fehler gering zu halten.

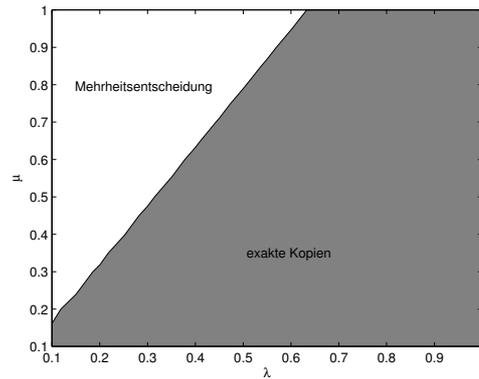
### 3.3.2.2 Einschwingverhalten

Neben den stationären Kosten ist auch das zeitliche Verhalten des Prozesses interessant, wenn zusätzliche Neuronen zur Erhöhung der Redundanz eingefügt werden. Dabei ist das Einschwingverhalten des Prozesses von besonderer Bedeutung. Diese Effekte können mit dem Lösen der Differentialgleichung elektrischer Schaltungen verglichen werden. Während des Einschwingvorgangs ergeben sich andere Signalverläufe als für den stationären Zustand für  $t \rightarrow \infty$ .

Analog zur Netzwerkanalyse sind für das Einschwingverhalten die Eigenwerte der Matrix  $\mathbf{Q}^T$  interessant, während für den stationären Zustand nur der Eigenvektor zum Eigenwert  $\lambda_i = 0$  Einfluss besitzt. Die Geschwindigkeit der Konvergenz des



(a) Differenz der Methoden



(b) Bereiche der besseren Methode

**Abbildung 3.10:** Vergleich der stationären Kosten der beiden vorgestellten Methoden. Die Abbildung zeigt die Differenz  $C_m - C_c$  für  $N = 25$  und die Bereiche der besseren Methode.

Prozesses ist vom Realteil der Eigenwerte abhängig. Das Einschwingverhalten wird somit von der kleinsten Komponente dominiert, d.h. ,

$$\min_{i=1,\dots,m+1} |\operatorname{Re}\{\lambda_i\}| \quad \lambda_i \neq 0 \quad (3.205)$$

bestimmt den Einschwingvorgang und für dessen jeweilige Zeitkonstante gilt

$$\tau = \max_{i=1,\dots,m+1} \left( -\frac{1}{\operatorname{Re}\{\lambda_i\}} \right). \quad (3.206)$$

Die Eigenwerte der Matrix  $\mathbf{Q}^T$  ergeben sich zu

$$\lambda_i = -i \cdot (\lambda + \mu) \quad \forall i = 0, \dots, m \quad (3.207)$$

wobei für alle Eigenwerte  $\lambda_i$  dieser Matrix gilt

$$\operatorname{Re}\{\lambda_i\} \leq 0, \quad (3.208)$$

was somit ein stabiles System zur Folge hat.

Aufgrund des Einfügens von zusätzlichen Neuronen wird die Konvergenzgeschwindigkeit des Einschwingverhaltens in (3.206) nicht verändert, da diese von dem kleinsten Realteil der Eigenwerte dominiert wird. Durch das Einfügen von zusätzlichen Neuronen werden jedoch nur Eigenwerte mit größerem Realteil hinzugefügt, die

sehr kurzzeitige Effekte hervorrufen. Somit bleibt die Konvergenzgeschwindigkeit erhalten, es ergibt sich für die Einschwingzeitkonstante

$$\tau = \frac{1}{\lambda + \mu}. \quad (3.209)$$

### 3.4 Zusammenfassung und Diskussion

In diesem Kapitel wurden die Robustheit und die Fehlertoleranz der radialen Basisfunktionen und des LCNNs untersucht. Für die Robustheit wurde eine mathematische Formulierung gefunden, mit der sich die Auswirkungen von Rauschen in den Eingängen und den Parametern auf den Ausgang bestimmen lassen. Die Analyse zeigte, dass künstliche neuronale Netze einen beliebigen Fehler im Ausgang produzieren, solange nicht gewisse Einschränkungen bei der Wahl der Parameter gemacht werden. Durch obere bzw. untere Schranken der Parameter eines Netzes kann der Fehler im Ausgang aufgrund gestörter Eingänge und Parametervariationen abgeschätzt werden. Diese Abschätzung kann zudem als Maß für den Vergleich zweier Netze hinsichtlich ihrer Rauscheigenschaften verwendet werden, wenn diese ähnliche Approximationseigenschaften besitzen. Diese können über die Vapnik-Chervonenkis-Dimension (VC-Dimension) [Vap98] erfasst werden. Je nach Anwendung können Aussagen über die Qualität eines neuronalen Netzes getroffen werden, ob ein Netz mit vielen Neuronen und Parametern einem kleinen Netz überlegen ist. Diese Aussage ist abhängig von Rauschleistung in den Eingängen bzw. in den Parametern. Bei Kenntnis der Signalleistung im Ausgang kann so die garantierte Auflösung eines neuronalen Netzes bestimmt werden, welche die Bewertung zweier Netze hinsichtlich ihrer Rauscheigenschaften vornehmen kann. Allgemeine Aussagen über die Kapazität des Netzes in Bezug zum Rauschverhalten lassen sich mit den Ergebnissen dieses Kapitels erreichen, indem die VC-Dimension und die Robustheit als Kostenfunktionen einer Mehrzieloptimierung verwendet werden.

Für die Fehlertoleranz wurde das Stuck-At-Fehlermodell verwendet und mit dessen Hilfe die Auswirkungen eines Fehlers während der Berechnung auf den Ausgang abgeschätzt. Hier ergeben sich analoge Ergebnisse wie bei der Robustheit des Netzes, so dass eine Toleranz gegenüber Fehlern nur für beschränkte Parameter des Netzes gewährleistet werden kann. Weiter wurden die radialen Basisfunktionen mit dem LCNN hinsichtlich der Zeit bis zum Auftreten eines Fehlers verglichen, wobei aufgrund der kompakteren Bauform eine Aussage zugunsten eines RBF-Netzes getroffen werden muss. Zudem wurde der sich einstellende Approximationsfehler bei unzuverlässigen Recheneinheiten bestimmt sowie einfache Methoden zur dessen Verringerung vorgestellt und miteinander verglichen. Diese Analyse ist ohne Einschränkung auch für das LCNN gültig, wobei nur die entsprechenden Parameter zur Beschreibung der Markov-Prozesse verwendet werden müssen. Die Methoden zur Erhöhung der Redundanz lassen sich direkt in Hardware umsetzen. Es werden

keine zusätzlichen Entwurfsschritte benötigt, und beide Verfahren lassen sich in die Entwurfswerkzeuge integrieren. Mit Kenntnis der Randbedingungen der jeweiligen Technologie ließe sich durch das Entwicklungswerkzeug das entsprechende Verfahren automatisiert anwenden. So kann der Entwurf der Systeme für zukünftige Technologien verbessert werden, indem der Entwurfsablauf bzw. die Entwurfswerkzeuge leicht modifiziert würden.

# Methodologie zur Fehlertoleranz und Robustheit

In zukünftigen Technologien, besonders in der Nanotechnologie, müssen die eingesetzten Architekturen fehlertolerant und robust sein [BRRN05]. Neuronale Netze scheinen aufgrund ihrer verteilten Informationsverarbeitung eine geeignete Architektur für die Anforderungen der neuen Technologien. In diesem Kapitel sollen Methoden vorgestellt werden, wie die Zuverlässigkeit neuronaler Netze erhöht werden kann. Dabei wurden die Aspekte der Robustheit gegenüber Parameterschwankungen und die Toleranz gegenüber einem Ausfall von Neuronen in Kapitel 3 näher untersucht. Da hier nun von technischen Realisierungen der neuronalen Netze ausgegangen wird, ist theoretisch die Zuverlässigkeit gegeben (siehe Abschnitt 3.1, Abschnitt 3.2 und 3.3). Basierend auf diesen Ergebnissen sollen durch erweiterte Entwurfsverfahren die Eigenschaften der Netze bzgl. dieser Anforderungen verbessert werden.

## 4.1 Erhöhen der Fehlertoleranz

Fehlertoleranz bezeichnet das Verhalten eines neuronalen Netzes gegenüber dem Ausfall einzelner bzw. mehrerer Elemente. Dabei können zum einen die Neuronen selbst ausfallen, indem sie fehlerhafte Werte bei der Berechnung liefern, und zum anderen können die Verbindungen zwischen den Neuronen ausfallen oder den Wert verändern. Eine Veränderung der Werte bei der Berechnung durch die Neuronen bzw. bei der Übertragung mittels der Verbindungen kann als Parametervariation oder Rauschen aufgefasst werden und wird in Abschnitt 4.2 näher untersucht.

Der Ausfall von Verbindungen bzw. von Neuronen kann mittels des *Stuck-At-Fehler*-modells [Smi97] direkt beschrieben werden. Dabei bleibt der Wert einer Verbindung oder einer Berechnung immer auf dem gleichen Wert bestehen. Bei einem Basisfunktionsnetz entspricht der Wegfall einer Verbindung dem Ausfall eines Neurons bzw. dem Fehlen eines Einganges. Somit kann dieser Fehler einem Neuron zugeordnet werden. Im Gegensatz dazu kann dies bei einem MLP nicht vorgenommen werden, da bei einem vollvermaschten Netz ein Ausfall eines Neurons alle seine abgehenden Verbindungen betreffen würde, während der Ausfall einer einzelnen

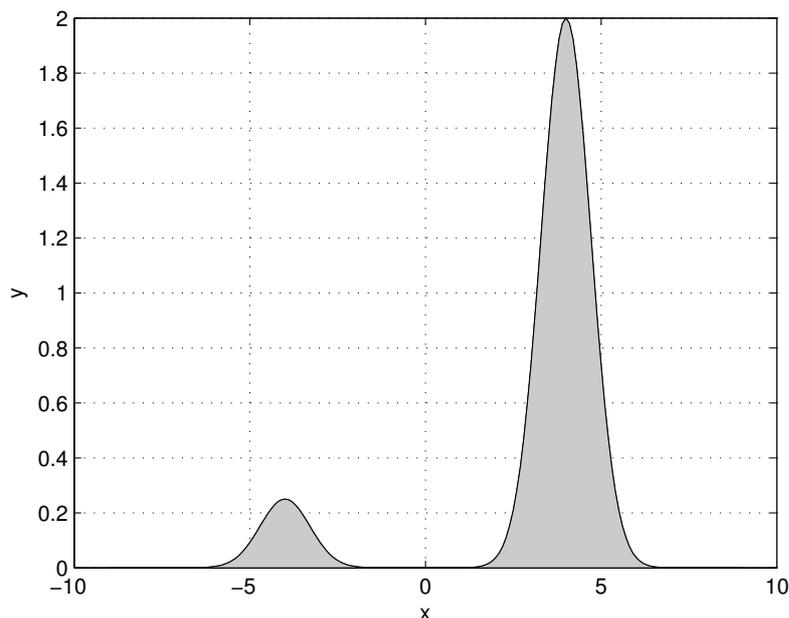


Abbildung 4.1: Approximation mittels zweier Basisfunktionen

Verbindung lokal begrenzt wäre. Im Folgenden sollen die radialen Basisfunktionsnetzwerke durch Ermittlung der Bedeutung von Neuronen in ihrer Fehlertoleranz erhöht werden.

*Beispiel.* Wie schon bei der Approximation der si-Funktion in Abschnitt 3.2.1 gesehen, hängen die Auswirkungen eines Wegfalls von Neuronen von den jeweiligen Gaußfunktionen mit ihren Parametern ab. In Abbildung 3.7 variierte der Fehler um mehr als eine Größenordnung.

#### 4.1.1 Bedeutung von Neuronen bei einem RBF

Die Methoden zur Erhöhung der Fehlertoleranz können für jedes Neuron angewendet werden, so dass bei einem Ausfall von diesem die Methoden greifen können. Allerdings ist die Fläche bei Realisierungen von Systemen in Hardware begrenzt, so dass die zusätzlichen Methoden Fläche benötigen und evtl. nicht für alle Neuronen angewendet werden können. Um eine optimale Auswahl der Neuronen treffen zu können, wird die Bedeutung von Neuronen eingeführt.

Abbildung 4.1 zeigt ein Beispiel für eine Approximation einer Funktion mittels eines radialen Basisfunktionsnetzes mit zwei Neuronen. Wird angenommen, dass beide Neuronen ausfallen können, so erkennt man, dass das rechte Neuron bei einem Ausfall einen größeren Fehler als das linke Neuron produzieren wird. Dies liegt an der von dem Neuron eingeschlossenen Fläche; je größer der Bereich unter der jeweiligen

Aktivierungsfunktion des Neurons desto größer wird auch der resultierende Fehler bei einem Ausfall sein.

Um die Bedeutung eines Neurons zu bestimmen, wird die Differenz zwischen der Ausgangsfunktion des ursprünglichen Netzes  $f_m(\mathbf{x})$  und des Netzes mit einem ausgefallenen Neuron  $\hat{f}_m(\mathbf{x})$  berechnet. Dazu wird dementsprechend das  $k$ -te Neuron aus der Ausgangsfunktion entfernt, womit dann für  $\hat{f}_m(\mathbf{x})$  folgt

$$\hat{f}_m(\mathbf{x}) = \sum_{i=1}^{k-1} \alpha_i \exp\left(-\sum_{j=1}^n \frac{(x_j - c_{ij})^2}{2\sigma_{ij}^2}\right) + \sum_{i=k+1}^m \alpha_i \exp\left(-\sum_{j=1}^n \frac{(x_j - c_{ij})^2}{2\sigma_{ij}^2}\right) \quad (4.1)$$

$$= \sum_{i=1}^{k-1} \alpha_i \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) + \sum_{i=k+1}^m \alpha_i \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right). \quad (4.2)$$

Dabei wird in (4.1) ein verallgemeinertes RBF-Netz verwendet, wobei die Kovarianzmatrix  $\Sigma_i$  eine Diagonalmatrix mit den Elementen  $\sigma_{ij}^2$  darstellt. Für identische Elemente auf der Hauptdiagonalen ergibt sich das bekannte RBF-Netz in (4.2).

Somit berechnet sich die Differenz zwischen beiden Funktionen zu

$$d^q(f_m, \hat{f}_m) = |f_m - \hat{f}_m|^q = \left| \alpha_k \exp\left(-\frac{1}{2} \sum_{j=1}^n \frac{(x_j - c_{kj})^2}{\sigma_{kj}^2}\right) \right|^q \quad (4.3)$$

$$= |\alpha_k|^q e^{-\frac{q\|\mathbf{x} - \mathbf{c}_k\|^2}{2\sigma_k^2}}, \quad (4.4)$$

wobei  $q$  die höheren Momente der Distanz bestimmt.

Allerdings hängt (4.3) immer noch von den aktuellen Eingangswerten ab. Jedoch kann der Eingangsvektor  $\mathbf{x}$  als ein Vektor von Zufallsvariablen beschrieben werden

$$\underline{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix}, \quad (4.5)$$

wobei  $X_i$  die Zufallsvariable des entsprechenden Einganges bezeichnet.

Für die Approximation einer Funktion mittels eines RBF-Netzes wird eine Menge von Testvektoren  $d = \{(\mathbf{x}^k, y^k) \in \mathbb{R}^n \times \mathbb{R}\}_{k=1}^M$ , mit  $M$  der Anzahl von Testdaten, bereitgestellt, die auf der Wahrscheinlichkeitsdichte  $p_{\underline{X}}(\mathbf{x})$  basieren. Somit kann auf (4.3) der Erwartungswert angewendet und die Bedeutung eines Neurons definiert werden:

**Definition 6** (Bedeutung eines Neurons). *Es seien  $y$  die ursprüngliche Netzfunktion und  $\hat{y}$  die Funktion des Netzes, aus dem das Neuron  $k$  entfernt wurde. Weiter bezeichne  $d^q(y, \hat{y}) = |y - \hat{y}|^q$  eine Metrik. Dann heißt*

$$s_k = E \{d^q(y, \hat{y})\} = \int_{\underline{X}} d^q(y, \hat{y}) p_{\underline{X}}(\mathbf{x}) d\mathbf{x} \quad (4.6)$$

Bedeutung des Neurons  $k$ .

Für das RBF-Netz kann die Bedeutung  $s_k$  jedes einzelnen Neurons angegeben werden, falls Informationen über die interne Struktur vorhanden sind. So folgt für ein verallgemeinertes RBF-Netz und für die Vereinfachung bei identischen Elementen der Hauptdiagonalen

$$s_k = E [d^q(y, \hat{y})] = \int_{\underline{X}} |\alpha_k|^q \exp \left( -\frac{q}{2} \sum_{j=1}^n \frac{(x_j - c_{kj})^2}{\sigma_{kj}^2} \right) p_{\underline{X}}(\mathbf{x}) d\mathbf{x} \quad (4.7)$$

$$= \int_{\underline{X}} |\alpha_k|^q e^{-\frac{q \|\mathbf{x} - \mathbf{c}_k\|^2}{2\sigma_k^2}} p_{\underline{X}}(\mathbf{x}) d\mathbf{x}. \quad (4.8)$$

Die Bedeutung existiert dabei für jede beliebige Verteilung, da das Integral in (4.7) bzw. (4.8) stets konvergiert. Für ein bestimmtes Integral im Intervall  $[a; b]$  ist dieses stets für eine stetige Funktion definiert [Heu03]. Für unbestimmte Integrale bei verschiedenen Verteilungen ergibt sich aufgrund des Majorantenkriteriums die Konvergenz des Integrals [Heu03], da die Gaußfunktion durch Eins beschränkt ist.

Damit legt (4.8) den statistischen Beitrag eines Neurons auf den Ausgang des RBF-Netzes fest [HSS05]. Der Parameter  $q$  bestimmt dabei die Gewichtung der Differenz und bezeichnet somit die höheren Momente.

Ist nun die Wahrscheinlichkeitsdichte der Eingänge bekannt, kann die Bedeutung jedes Neurons innerhalb des Netzes berechnet werden. Bei unbekannter Verteilung der Eingänge können Schätzverfahren angewendet werden, so dass (4.8) bestimmt werden kann [MH98].

Es soll nun beispielhaft für zwei wichtige Dichtefunktionen der Eingänge die Bedeutung berechnet werden. Dabei wird davon ausgegangen, dass die Eingänge voneinander stochastisch unabhängig sind, d.h., es gilt

$$p_{\underline{X}}(\mathbf{x}) = \prod_{i=1}^n p_{X_i}(x_i). \quad (4.9)$$

Damit kann die gemeinsame Dichtefunktion als Multiplikation der einzelnen Wahrscheinlichkeitsdichten dargestellt werden.

Für den Fall einer verallgemeinerten Gaußfunktion, bei der sich keine Diagonalmatrix für die Kovarianzmatrix ergibt, folgt für die Definition der Bedeutung über den Erwartungswert

$$s_k = \int_{\underline{X}} |\alpha_k|^q \exp\left(-\frac{q}{2}(\mathbf{x} - \mathbf{c}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \mathbf{c}_k)\right) p_{\underline{X}}(\mathbf{x}) d\mathbf{x}. \quad (4.10)$$

Zur Bestimmung der Bedeutung kann das Integral für bestimmte Spezialfälle und Randbedingungen gelöst werden. Eine Vereinfachung kann vorgenommen werden, welche die Eigenschaften der Niveaulinien bei einer Gaußfunktion ausnutzt. Diese stellen ein Ellipsoid dar und die Halbachsen können zur Berechnung der Bedeutung genutzt werden. Auf diese Vereinfachung wird in Abschnitt 4.1.2.1 näher eingegangen.

#### 4.1.1.1 Gleichverteilung der Eingänge

Bei einer Gleichverteilung der Eingänge und stochastischer Unabhängigkeit folgt für die Wahrscheinlichkeitsdichte des Zufallsvariablenvektors  $\underline{X}$

$$p_{\underline{X}}(\mathbf{x}) = \prod_{i=1}^n p_{X_i}(x_i) = \prod_{i=1}^n \begin{cases} \frac{1}{b_i} & d_i - \frac{b_i}{2} \leq x \leq d_i + \frac{b_i}{2} \\ 0 & \text{sonst} \end{cases}, \quad (4.11)$$

wobei  $b_i$  die Breite der Gleichverteilung  $X_i$  und  $d_i$  den Erwartungswert jeder einzelnen Zufallsvariablen bezeichnet.

Löst man (4.7) oder analog (4.8) für die Dichtefunktion aus (4.11), so erhält man für die Bedeutung  $s_k$  jedes einzelnen Neurons

$$s_k = \int_{\underline{X}} |\alpha_k|^q \exp\left(-\frac{q}{2} \sum_{j=1}^n \frac{(x_j - c_{kj})^2}{\sigma_{kj}^2}\right) \prod_{i=1}^n p_{X_i}(x_i) d\mathbf{x} \quad (4.12)$$

$$= \int_{X_1} \cdots \int_{X_n} |\alpha_k|^q \exp\left(-\frac{q}{2} \sum_{j=1}^n \frac{(x_j - c_{kj})^2}{\sigma_{kj}^2}\right) \prod_{i=1}^n p_{X_i}(x_i) dx_1 \cdots dx_n \quad (4.13)$$

$$= |\alpha_k|^q \int_{X_1} \cdots \int_{X_n} \prod_{i=1}^n e^{-\frac{q(x_i - c_{ki})^2}{2\sigma_{ki}^2}} p_{X_i}(x_i) dx_1 \cdots dx_n \quad (4.14)$$

$$= |\alpha_k|^q \left(\frac{\sqrt{\pi}}{\sqrt{2q}}\right)^n \prod_{i=1}^n \frac{\sigma_{ki}}{b_i} \left( \operatorname{erf}\left(\frac{\sqrt{2q}(b_i - 2d_i + 2c_{ki})}{4\sigma_{ki}}\right) - \operatorname{erf}\left(\frac{\sqrt{2q}(-b_i - 2d_i + 2c_{ki})}{4\sigma_{ki}}\right) \right), \quad (4.15)$$

wobei die Bedingung  $\sigma_{ki} > 0$  erfüllt sein muss, was aufgrund der positiven definiten Matrix  $\boldsymbol{\Sigma}_k$  gewährleistet ist. Zusätzlich beschreibt  $\operatorname{erf}(\cdot)$  die Fehler-Funktion [BSM05] und  $c_{ki}$  den Eintrag  $i$  des  $k$ -ten Zentrums.

Wenn alle Wahrscheinlichkeitsdichten der einzelnen Zufallsvariablen  $X_i$  identische Breiten  $b_i$  und identische Erwartungswerte besitzen, kann (4.15) weiter vereinfacht werden. Ebenso können bei identischen Elementen auf der Hauptdiagonalen der Kovarianzmatrix  $\Sigma_k$  die einzelnen Varianzen vor das Produkt in (4.15) gezogen werden.

Für eine vereinfachte Bestimmung der Bedeutung einer Basisfunktion nach (4.15) kann die Integration über den gesamten reellen Eingangsraum  $(-\infty; \infty)$  durchgeführt werden. Da die Eingänge als stochastisch unabhängig angenommen werden, können die Integrale getrennt gelöst werden. Bei der Integration über den gesamten Eingangsraum ist der Einfluss der Gaußfunktion aufgrund ihres exponentiellen Abfallens für endliche Varianzen gering. Dieses Verhalten ist in Abbildung 4.2(a) dargestellt. Mit dieser Vereinfachung ergibt sich für die Bedeutung der Neuronen

$$s_k \approx \int_{\mathbb{R}^n} |\alpha_k|^q \exp\left(-\frac{q}{2} \sum_{j=1}^n \frac{(x_j - c_{kj})^2}{\sigma_{kj}^2}\right) p_{\underline{X}}(\mathbf{x}) d\mathbf{x} \quad (4.16)$$

$$= |\alpha_k|^q \left(\frac{\sqrt{2\pi}}{\sqrt{q}}\right)^n \prod_{i=1}^n \frac{\sigma_{ki}}{b_i}, \quad (4.17)$$

wobei weitere Vereinfachungen für den Fall identischer Elemente der Hauptdiagonalen vorgenommen werden können.

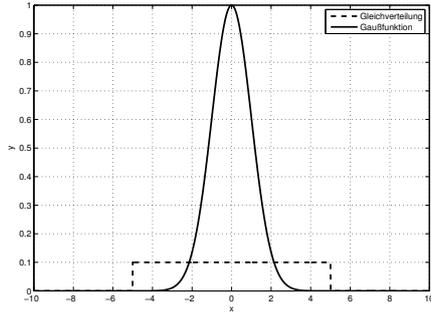
Wie man an (4.15) und (4.17) erkennen kann, verschwindet der Einfluss eines Neurons nur für den Fall verschwindender Varianzen  $\sigma_k$  oder Gewichte  $\alpha_k$ . Die Differenz der beiden erf-Funktionen in (4.15) ist für alle Werte der Verteilungsfunktion bzw. der Gewichte ungleich Null, da die beiden Argumente voneinander verschieden sind. Für den Fall von verschwindenden Gewichten und Varianzen würden die entsprechenden Basisfunktionen für die Approximation nicht mehr genutzt, so dass sich eine Bedeutung von Null nicht ergeben kann. Damit stellen die Maße nach (4.15) und (4.17) eine sinnvolle Bestimmung des Einflusses der Neuronen auf den Ausgang dar.

#### 4.1.1.2 Normalverteilung der Eingänge

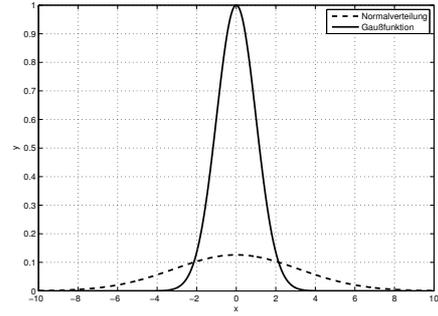
Sind die einzelnen Zufallsvariablen der Eingänge durch eine normalverteilte Wahrscheinlichkeitsdichte beschrieben, so folgt für die gemeinsame Dichtefunktion

$$p_{\underline{X}}(\mathbf{x}) = \prod_{i=1}^n p_{X_i}(x_i) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\phi_i} e^{-\frac{(x_i - \mu_i)^2}{2\phi_i^2}}, \quad (4.18)$$

wobei  $\phi_i^2$  die Varianz jeder einzelnen Zufallsvariablen beschreibt.



(a) bei einer Gleichverteilung



(b) bei einer Gaußverteilung

**Abbildung 4.2:** Vereinfachte Bestimmung der Bedeutung einer Basisfunktion

Mit der gemeinsamen Dichtefunktion kann die Bedeutung für jede Basisfunktion eines verallgemeinerten RBF-Netzes nach (4.7) berechnet werden zu

$$s_k = \int_{\underline{X}} |\alpha_k|^q \exp\left(-\frac{q}{2} \sum_{j=1}^n \frac{(x_j - c_{kj})^2}{\sigma_{kj}^2}\right) \prod_{i=1}^n p_{X_i}(x_i) d\mathbf{x} \quad (4.19)$$

$$= |\alpha_k|^q \int_{X_1} \cdots \int_{X_n} \prod_{i=1}^n e^{-\frac{q(x_i - c_{ki})^2}{2\sigma_{ki}^2}} p_{X_i}(x_i) dx_1 \cdots dx_n \quad (4.20)$$

$$= |\alpha_k|^q \prod_{i=1}^n \sigma_{ki} \frac{e^{-\frac{q(\mu_i - c_{ki})^2}{2(\sigma_{ki}^2 + q\phi_i^2)}}}{\sqrt{\sigma_{ki}^2 + q\phi_i^2}}. \quad (4.21)$$

In ähnlicher Weise wie zuvor bei der Gleichverteilung der Eingänge kann die Bestimmung der Bedeutung nach (4.21) vereinfacht werden. Die Daten sind zumeist über einen großen Eingangsbereich verteilt, d.h., die Wahrscheinlichkeitsdichte besitzt eine große Varianz. Die Gaußfunktionen des RBF-Netzes überdecken dagegen nur einen kleinen Teilbereich der Eingangsdaten, so dass  $\sigma_{ki} \ll \phi_i$  gilt. So sind die Auswirkungen der Basisfunktion außerhalb eines Bereiches nur noch gering, wie man in Abbildung 4.2(b) erkennen kann. Sind weiterhin die Elemente der Hauptdiagonalen identisch, d.h.  $\sigma_{ki} = \sigma_k, \forall i = 1 \dots n$ , so vereinfacht sich (4.21) zu

$$s_k \approx |\alpha_k|^q \sigma_k^n \prod_{i=1}^n \frac{1}{\phi_i \sqrt{q}} e^{-\frac{(\mu_i - c_{ki})^2}{2\phi_i^2}}. \quad (4.22)$$

Weiter kann (4.22) ähnlich wie auch bei der Gleichverteilung der Eingänge vereinfacht werden, wenn alle Zufallsvariablen identische Dichtefunktionen besitzen ( $\mu_i = \mu, \phi_i = \phi \quad \forall i = 1 \dots n$ ).

Ähnlich wie schon bei der Bedeutung eines Neurons bei der Gleichverteilung der Eingänge kann diese auch bei einer Gaußverteilung nicht verschwinden. (4.21) und

(4.22) zeigen, dass ebenfalls nur für verschwindende Gewichte bzw. Varianzen der Einfluss eines Neurons auf den Ausgang verschwindet. Für diese Werte der Parameter werden die jeweiligen Basisfunktionen im Netz nicht genutzt.

### 4.1.1.3 Simulationsergebnisse

Zur simulativen Überprüfung der Bedeutung von Neuronen wird diese für verschiedene RBF-Netze bestimmt, wobei diese unterschiedliche Testfunktionen [HK04, Ros60] approximieren. Für diese Aufgabe wurden eine Menge von Vektoren anhand dieser Funktionen für das Lernen sowie eine Menge zum Testen der Netze erstellt. Zum Lernen wurden insgesamt 1000 Lernvektoren  $\{\mathbf{x}^i, y^i\}$  mit  $\dim(\mathbf{x}) = 2$  und analog 5000 Wertepaare zum Überprüfen der Resultate jeweils im Intervall  $x_i \in [-5; 5]$  erstellt. Als Lernverfahren wurde der in NETLAB [Nab02] eingesetzte Algorithmus verwendet. Beim Trainieren der Netze wurde als Abbruchkriterium ein mittlerer quadratischer Fehler kleiner 0,1 gewählt, wobei von einem kleinen Netz ausgegangen wurde. Wurde dieses Kriterium nicht durch das aktuelle Netz erfüllt, so wurde eine zusätzliche Gaußfunktion hinzugefügt. Nach dem Lernen wurde der auf die Leistung des Ausganges normierte, mittlere quadratische Fehler (nmse) bzgl. dieser beiden Mengen bestimmt. Zusätzlich wurde dieser Fehler bei einem Ausfall eines Elementes bestimmt. Einmal wurde die Basisfunktion aus dem Netz entfernt, welche nach (4.15) oder (4.21) den größten Einfluss besitzt. Weiter wurde die Basisfunktion mit der geringsten Bedeutung entfernt sowie die Einflussreichste durch 10 exakte Kopien ersetzt und deren Gewichte angepasst (vgl. Abschnitt 3.3.2), wobei eines dieser Neuronen entfernt wurde.

Tabelle 4.1 zeigt die Auswirkungen des Ausfalls eines Neuron auf den normierten mittleren quadratischen Fehler für die Testfunktionen bei beiden Dichtefunktionen. Bei der Simulation wurde überprüft, ob die Bedeutung das Neuron mit dem größten Einfluss identifiziert. Dazu wurde jede einzelne Basisfunktion aus dem Netz entfernt und der resultierende Fehler bestimmt. Es ergab sich für alle Testfälle eine Übereinstimmung.

**Tabelle 4.1:** Einstellende normierte, mittlere quadratische Fehler für verschiedene RBF-Netzwerke und Testfunktionen

Funktion	m	nmse				
		Training	Test	Ausfall		
				max $s_k$	min $s_k$	mit Korrektur
Gleichverteilung						
Griewank	13	0,1957	0,1985	$4,80 \cdot 10^8$	$1,96 \cdot 10^6$	$3,97 \cdot 10^6$
	12	0,2136	0,2187	$2,97 \cdot 10^8$	$1,53 \cdot 10^6$	$2,45 \cdot 10^6$
	10	0,2163	0,2261	$1,04 \cdot 10^9$	$1,14 \cdot 10^7$	$8,60 \cdot 10^6$
Rosenbrock <sup>1</sup>	24	0,0000	0,0000	$1,22 \cdot 10^{11}$	$4,15 \cdot 10^7$	$1,01 \cdot 10^9$
	24	0,0000	0,0000	$7,93 \cdot 10^{14}$	$2,25 \cdot 10^6$	$6,55 \cdot 10^{12}$
	23	0,0000	0,0000	$2,13 \cdot 10^{12}$	$7,44 \cdot 10^6$	$1,76 \cdot 10^{10}$
Rechenberg	7	0,0002	0,0003	$5,81 \cdot 10^1$	$2,03 \cdot 10^0$	$4,81 \cdot 10^{-1}$
	7	0,0002	0,0003	$2,37 \cdot 10^2$	$3,53 \cdot 10^0$	$1,95 \cdot 10^0$
	6	0,0003	0,0003	$1,60 \cdot 10^2$	$3,56 \cdot 10^{-1}$	$1,32 \cdot 10^0$
Schaffer	2	0,2910	0,2919	0,6241	0,3417	0,2938
	2	0,2846	0,2878	0,4108	0,3600	0,2884
	2	0,2895	0,2910	0,5534	0,3442	0,2925
Normalverteilung						
Griewank	3	0,0435	0,0413	0,2187	0,1084	0,0426
	3	0,0452	0,0426	0,1703	0,0946	0,0435
	3	0,0436	0,0413	0,1515	0,1289	0,0421
Rosenbrock <sup>1</sup>	20	0,0000	0,0000	$6,90 \cdot 10^{12}$	$9,17 \cdot 10^6$	$5,70 \cdot 10^{10}$
	22	0,0000	0,0000	$5,21 \cdot 10^{13}$	$1,03 \cdot 10^8$	$4,31 \cdot 10^{11}$
	22	0,0000	0,0000	$2,63 \cdot 10^{14}$	$9,61 \cdot 10^7$	$2,17 \cdot 10^{12}$
Rechenberg	6	0,0016	0,0019	$1,88 \cdot 10^3$	$3,83 \cdot 10^1$	$1,55 \cdot 10^{12}$
	5	0,0017	0,0022	$1,03 \cdot 10^3$	$4,38 \cdot 10^1$	$8,49 \cdot 10^0$
	6	0,0029	0,0033	$7,73 \cdot 10^2$	$2,91 \cdot 10^1$	$6,38 \cdot 10^0$
Schaffer	4	0,2115	0,2009	13,656	0,7602	0,2977
	4	0,2218	0,2093	1,1543	0,3796	0,2204
	4	0,2255	0,2141	0,9916	0,2686	0,2168

<sup>1</sup> Aufgrund der hohen Ausgangsleistung der Rosenbrockfunktion ergeben sich sehr kleine normierte Fehler.

Wie man an der Tabelle 4.1 erkennen kann, können sehr große Fehler bei dem Ausfall eines Neurons entstehen, die unzulässig groß für die Aufgabe der Funktionsapproximation sind. Die am Ausgang auftretenden Fehler übersteigen die ursprünglichen Approximationseigenschaften um mehrere Größenordnungen. Um diese zu reduzieren, muss in das Netz Redundanz eingefügt werden, um so den Fehler bei einem Ausfall zu reduzieren. Ein Trainieren eines großen Netzes ohne zusätzliche

Randbedingungen für die Fehlertoleranz muss nicht zwangsläufig zu einem fehler-toleranten Netz führen. Bei einer großen Anzahl von Neuronen besteht die Gefahr, dass bei der Approximation ein zu komplexes Modell angewendet wird und so eine Überanpassung durch das Netz vorgenommen wird. Dies muss durch entsprechende Randbedingungen vermieden werden, so dass mehrere Neuronen den gleichen Eingangsraum überdecken und die Gewichte reduziert werden können.

Tabelle 4.1 zeigt, dass der mittlere quadratische Fehler bei zusätzlicher Fehlerkorrektur in einigen Fällen zwei Größenordnungen unter dem ursprünglichen Fehler liegt. Allerdings kann dieses Verhalten nur für große absolute Werte der Bedeutung eines Neurons beobachtet werden (vgl. die Ergebnisse der Schaffer-Funktion).

Dies beruht auf der Bestimmung des quadratischen Fehlers. Dieser berechnet sich zu

$$\text{mse} = \frac{1}{M} \sum_{i=1}^M (f_m(\mathbf{x}^i) - y^i)^2 \quad (4.23)$$

und falls das Neuron mit der maximalen Bedeutung ausfällt, dominiert dieser Term den Fehler nach (4.23). Diese Dominanz ergibt sich nur für den Fall einer sehr hohen Bedeutung des Neurons im Vergleich zu den anderen. Ist der Unterschied gering, so ergibt sich nur eine kleine Verbesserung.

Wird das Neuron mit der höchsten Bedeutung durch mehrere Kopien ersetzt und deren Ausgangsgewichte reduziert, verringert sich auch deren Einfluss auf den mittleren quadratischen Fehler in (4.23). Fällt eine dieser Kopien aus, so dominiert dieses Neuron den mittleren quadratischen Fehler. Da jedoch das Ausgangsgewicht um den Faktor  $\hat{\alpha} = \frac{\alpha}{N}$  reduziert wurde, sinkt der Fehler  $\propto 1/N^2$  aufgrund des quadratischen Fehlers nach (4.23). Dieser Effekt reduziert sich, falls die Neuronen gleiche Werte bei der Bedeutung annehmen und eine Dominanz eines Termes nicht vorhanden ist. So ist die Dominanz eines einzelnen Terms nicht mehr gewährleistet, und der Abfall fällt geringer aus.

Um die Berechnung der Bedeutung zu verifizieren, wurde für die Testfunktionen der entstehende relative Fehler zwischen der Bedeutung und dem Fehler bei einem Wegfall bestimmt. Da die Bedeutung als Erwartungswert des mittleren quadratischen Fehlers bei dem Ausfall einer Basisfunktion definiert wurde, sind dafür 50 neuronale Netze für die verschiedenen Testfunktionen und Wahrscheinlichkeitsverteilungen der Eingänge trainiert worden. Die Anzahl von Basisfunktionen in den Netzen wurde einer Gleichverteilung im Intervall  $[1; 100]$  entnommen. Nach dem Training ist dann der Erwartungswert des relativen Fehlers zwischen der Bedeutung eines Neurons und dem mittleren quadratischen Fehler bei dessen Wegfall bestimmt worden, wobei die Bedeutung sowohl exakt als auch mit Hilfe der Näherungsformeln bestimmt wurde.

Bei der vereinfachten Berechnung des Einflusses einer Basisfunktion wird sich ein größerer relativer Fehler ergeben, da die Randbedingungen nicht immer eingehal-

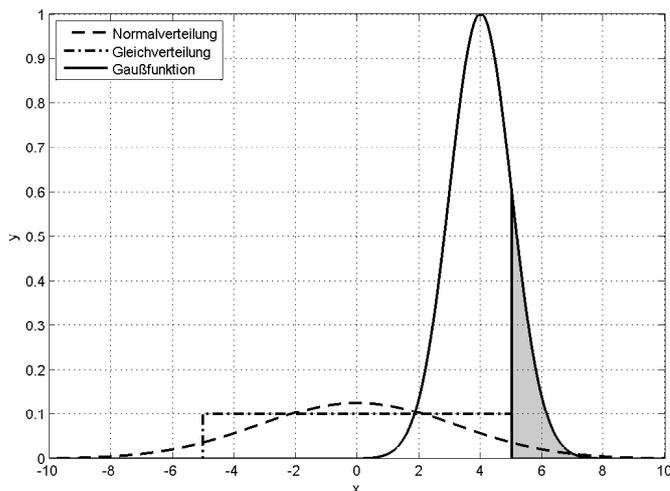
**Tabelle 4.2:** Fehler zwischen der Bedeutung eines Neurons und dem mittleren quadratischen Fehler bei einem Ausfall

Funktion	Verteilung	relativer Fehler		$E(\mathcal{V})$
		se	s	
Rechenberg	Normal	0,0016	6,4700	0,20
	Gleich	0,0004	4,8404	0,22
Rosenbrock	Normal	0,0019	8,3356	0,16
	Gleich	0,0002	4,5173	0,28
Schwefels-Rochen	Normal	0,0011	7,2981	0,08
	Gleich	0,0005	4,5489	0,24
Schwefel	Normal	0,0007	5,3259	0,34
	Gleich	0,0021	4,5459	0,33
Griewank	Normal	0,0008	7,4036	0,10
	Gleich	0,0002	4,7400	0,31
Schaffer	Normal	0,0006	10,0158	0,08
	Gleich	0,0009	4,7281	0,22

ten werden. Für einen Einsatz dieser Technik zur Fehlerkorrektur ist jedoch die Identifikation der Basisfunktionen mit dem größten Einfluss wichtig und nicht unbedingt der absolute Wert der Bedeutung. Zur Fehlerkorrektur ist z.B. ein zusätzlicher Gleichanteil nicht von Relevanz. Dafür wird eine Zufallsvariable  $\mathcal{V}$  eingeführt, die den Wert 1 annimmt, wenn insgesamt die exakte Bestimmung der Bedeutung mit der vereinfachten zu 30 % in der oberen Reihenfolge übereinstimmt. Für diesen Fall sind die Mengen, welche 30 % aller Neuronen mit der jeweils größten Bedeutung enthalten, für beide Berechnungsvorschriften gleich. Sind beide Mengen ungleich, so wird der Zufallsvariablen  $\mathcal{V}$  der Wert 0 zugewiesen.

Tabelle 4.2 zeigt den relativen Fehler für die Bestimmung der Bedeutung nach (4.15) und (4.21) sowie (4.17) und (4.22). Es zeigt sich eine sehr gute Übereinstimmung zwischen dem tatsächlichen mittleren quadratischen Fehler und dem durch die Bedeutung vorausgesagten Werten. Die Vereinfachungen führen zu sehr großen relativen Fehlern. Auch sieht man, dass nur in wenigen Fällen die Mengen mit 30 % der Basisfunktionen mit maximaler Bedeutung übereinstimmen. Besonders für eine Normalverteilung ergeben sich schlechtere Werte als bei einer Gleichverteilung der Eingänge.

Die Vereinfachung bei einer Gleichverteilung bestand darin, die Integration zur Bestimmung des Erwartungswertes über den gesamten Eingangsraum durchzuführen. Dies ist zulässig, wenn die Gaußfunktionen nur einen kleinen Bereich des Einganges abdecken und ihr Zentrum nicht am Randbereich der Wahrscheinlichkeitsdichte liegt. Wie man in Abbildung 4.3 erkennen kann, ist die Bestimmung der Bedeutung



**Abbildung 4.3:** Ursachen für die ungenaue Bestimmung der Bedeutung

desto ungenauer je weiter das Zentrum an dem Randbereich der Verteilung liegt. Bei der vereinfachten Bestimmung fließt der schraffierte Bereich mit in die Berechnung ein, wobei dieser jedoch keinen Einfluss besitzt und zu einem Fehler führt. Da bei dem verwendeten NETLAB-Algorithmus die Lage der Zentren anhand der Verteilung der Eingangsdaten gewählt wird, ergibt sich so eine Gleichverteilung der Mittelpunkte über den Eingangsraum. Dadurch existieren an den Randbereichen der Verteilungsdichte mehrere Basisfunktionen, deren Einfluss aufgrund der Vereinfachung als zu groß eingeschätzt wird.

Bei einer Normalverteilung der Eingänge ergibt sich ein solches Problem nicht. Die schlechteren Ergebnisse bei einer Normalverteilung resultieren aus zwei Faktoren. Zuerst ist die Vereinfachung nur für den Fall  $\sigma_{ki} \ll \phi_i$  gültig, d.h. für viel kleinere Varianzen der Basisfunktionen als bei der Wahrscheinlichkeitsverteilung der Eingänge. Da beim Trainieren mit NETLAB die Varianzen aller Basisfunktionen gleich dem maximalen Abstandsquadrat zweier Zentren gesetzt werden, ist diese Bedingung nicht erfüllt. Hieraus resultiert somit ein Fehler, der gerade bei einer kleinen Anzahl von Basisfunktionen verstärkt auftritt, da hier sehr große Varianzen verwendet werden.

Der weitere Grund für die schlechteren Ergebnisse bei einer Normalverteilung ist die unzureichende Repräsentation unter MATLAB. Bei der Erzeugung der Trainingsvektoren basierend auf der Dichtefunktion werden diese Daten nur innerhalb eines endlichen Bereiches erzeugt. Bei der Normalverteilung können jedoch Werte aus dem gesamten Eingangsbereich angenommen werden, was zusätzlich mit in den Fehler einfließt. Die erste Einschränkung dominiert jedoch den entstehenden Fehler bei der vereinfachten Bestimmung der Bedeutung.

*Beispiel.* Bei der Approximation der si-Funktion durch ein RBF-Netz und dem Wegfall eines Neurons (siehe Abbildung 3.7) kann so die Bedeutung aller Neuronen bestimmt und die Auswirkungen eines Wegfalls abgeschätzt werden. Dabei wird durch die vorgestellte Methode ( $q = 2$ ) genau das Neuron identifiziert, welches bei einem Ausfall den maximalen, mittleren quadratischen Fehler liefert. Ebenfalls kann die Gaußfunktion mit dem minimalen Kostenanstieg ermittelt werden.

### 4.1.2 Bedeutung der Neuronen beim LCNN

Im Abschnitt zuvor wurde die Bedeutung von Neuronen bei einem radialen Basisfunktionsnetzwerk betrachtet. In ähnlicher Weise lässt sich auch für das LCNN eine Analyse hinsichtlich der Bedeutung der Neuronen durchführen. Wird innerhalb eines LCNNs aus (2.21) die  $k$ -te lokale Funktion entfernt, so ergibt sich für den gestörten Ausgang

$$\hat{y}_m(\mathbf{x}) = \sum_{\nu=1}^{k-1} \alpha_\nu \mathcal{L}(\mathbf{W}_\nu, \mathbf{r}_\nu, \mathbf{x}) + \sum_{\nu=k+1}^m \alpha_\nu \mathcal{L}(\mathbf{W}_\nu, \mathbf{r}_\nu, \mathbf{x}) \quad (4.24)$$

und damit für die Differenz der Ausgänge in ähnlicher Weise wie in (4.3)

$$d^q(y_m, \hat{y}_m) = |y_m - \hat{y}_m|^q = |\alpha_k \mathcal{L}(\mathbf{W}_k, \mathbf{r}_k, \mathbf{x})|^q \quad (4.25)$$

$$= |\alpha_k|^q \mathcal{L}(\mathbf{W}_k, \mathbf{r}_k, \mathbf{x})^q, \quad (4.26)$$

wobei  $q$  die entsprechenden Momente bezeichnet.

Weiter hängt (4.25) auch wieder von den aktuellen Eingangswerten ab, und somit kann wie zuvor mit der Wahrscheinlichkeitsdichte der Zufallsvariablen die Bedeutung bestimmt werden

$$s_k = E[d^q(y_m, \hat{y}_m)] = \int_{\underline{X}} |\alpha_k|^q \mathcal{L}(\mathbf{W}_k, \mathbf{r}_k, \mathbf{x})^q p_{\underline{X}}(\mathbf{x}) d\mathbf{x}, \quad (4.27)$$

wobei auch hier für (4.27) verschiedene Dichtefunktionen zugrunde gelegt werden können. Das Integral konvergiert dabei mit der gleichen Begründung wie in Abschnitt 4.1.1. Für Verteilungen mit Dichtefunktionen über einem abgeschlossenen Intervall ergibt sich die Konvergenz des Integrals direkt, und für Dichtefunktionen, die zu einem uneigentlichen Integral führen, folgt die Konvergenz mit dem Majorantenkriterium. Allerdings lässt sich das Integral aus (4.27) nicht geschlossen lösen, und man erhält keine geschlossenen Formeln wie bei einem RBF (siehe (4.15) oder (4.21)).

Für den Fall einer Gleichverteilung der Eingänge lässt sich das Integral aus (4.27) jedoch mit Hilfe der Monte-Carlo-Methode lösen, da hier über einen  $n$ -dimensionalen Quader integriert werden muss. Bei einer hoch-dimensionalen Gaußverteilung ist

dagegen die Lösung von (4.27) wesentlich komplexer, da in diesem Fall nicht über ein endliches Intervall integriert werden kann. Bei einer eindimensionalen Gaußverteilung befinden sich ca. 90 % der Fläche unter der Gaußfunktion in dem Intervall  $[-3\sigma; 3\sigma]$ . Dieser prozentuale Anteil sinkt jedoch stark mit Zunahme der Dimension der Verteilung [VF05]. Somit muss bei einer hoch-dimensionalen Verteilung über den kompletten Raum  $\mathbb{R}^n$  integriert werden, da sonst ein zusätzlicher, systematischer Fehler entsteht.

Bei der Monte-Carlo-Methode kann das Integral

$$\int_{\mathcal{B}} f(\mathbf{x}) d\mathbf{x} \quad \mathcal{B} \subseteq \mathbb{R}^n \quad (4.28)$$

durch

$$\int_{\mathcal{B}} f(\mathbf{x}) d\mathbf{x} = \lambda_s(\mathcal{B}) E\{f(\mathcal{X})\} \quad (4.29)$$

berechnet werden [Sch98a], wobei  $\lambda_s(\mathcal{B})$  das Lebesgue-Maß der abgeschlossenen Menge  $\mathcal{B}$  bezeichnet. Für einen Hyperquader,  $x_i \in [a_i; b_i]$ , gilt dabei

$$\lambda_s(\mathcal{B}) = \prod_{i=1}^n (b_i - a_i). \quad (4.30)$$

Im Falle eines mehrdimensionalen Quaders ließe sich das Integral aus (4.28) nach dem Satz von Fubini mit Hilfe von eindimensionalen Verfahren lösen, z.B. der Trapez- und der Simpson-Regel. Allerdings ist bei der Monte-Carlo-Methode die Fehlerschranke bei der Approximation des Integrals nicht von der Dimension abhängig, so dass diese Methode den anderen Verfahren überlegen ist [Sch98a]. Im Folgenden wird für die Lösung von (4.27) eine Monte-Carlo-Methode unter MATLAB verwendet.

Für die Bedeutung einer lokalen Funktion im LCNN ergibt sich aus (4.27) und (4.29) bei einer Gleichverteilung

$$s_k = \int_{\underline{X}} |\alpha_k|^q \mathcal{L}(\mathbf{W}_k, \mathbf{r}_k, \mathbf{x})^q p_{\underline{X}}(\mathbf{x}) d\mathbf{x} \quad (4.31)$$

$$= \int_{\underline{X}} |\alpha_k|^q \mathcal{L}(\mathbf{W}_k, \mathbf{r}_k, \mathbf{x})^q \prod_{i=1}^n \frac{1}{b_i - a_i} d\mathbf{x} \quad (4.32)$$

$$= E\left\{ |\alpha_k|^q \mathcal{L}(\mathbf{W}_k, \mathbf{r}_k, \mathbf{x})^q \right\}. \quad (4.33)$$

#### 4.1.2.1 Bedeutung in Bezug zum RBF

Das LCNN wurde entworfen, um dem Fluch der Dimensionalität [Hay99, VF05] im Vergleich zu einem RBF-Netz zu begegnen [GS93, GS92]. Dabei ist das Ziel einer lokalen Funktion, eine verallgemeinerte Gaußfunktion zu approximieren. In Abschnitt

4.1.1 konnte die Bedeutung jeder Basisfunktion mit Hilfe einer geschlossenen Formel bestimmt werden, was im Vergleich zur Anwendung numerischer Verfahren den Einsatz zur Laufzeit vereinfacht.

Mit Hilfe der engen Beziehung zwischen einem verallgemeinerten RBF-Netz aus (2.31) und einem LCNN ist es möglich, die Parameter eines LCNNs in die entsprechenden Variablen des Basisfunktionsnetzes umzuwandeln. Das LCNN versucht mittels der Überlagerung mehrerer Sigmoiden, die Gaußfunktion zu approximieren. Als Parameter einer lokalen Funktion werden dafür die Gewichtsmatrix  $\mathbf{W}$ , das Zentrum  $\mathbf{r}$  und ein Ausgangsgewicht  $\alpha$  benötigt (vgl. Abschnitt 2.2.2). Für eine analoge Gaußfunktion können das Zentrum und das Ausgangsgewicht sofort zur Berechnung der Bedeutung herangezogen werden. Für die räumliche Ausdehnung des LCNNs ist die Gewichtsmatrix verantwortlich, die eine Normierung der einzelnen Eingänge wie bei einem verallgemeinerten RBF-Netz vornimmt. Zur Bestimmung der Varianz der analogen Gaußfunktion kann diese Beziehung genutzt werden.

Nach (2.29) gilt für die Inverse der Kovarianzmatrix einer verallgemeinerten Gaußfunktion

$$\frac{1}{2}\Sigma^{-1} = \mathbf{C}^T \mathbf{C} = \mathbf{W}^T \mathbf{W}, \quad (4.34)$$

da die Gewichtsmatrix  $\mathbf{W}$  bei einem LCNN die Normierung der Eingänge vornimmt. Für den Fall einer Diagonalmatrix lassen sich die Varianzen zur Berechnung der Bedeutung direkt aus der Matrix entnehmen.

Im Allgemeinen entsteht bei der Normierung aber keine Diagonalmatrix, da eine Normierung hinsichtlich der kartesischen Koordinaten nicht immer sinnvoll ist. So entsteht ein gedrehtes Hyperellipsoid [KK00], dessen Längen der Halbachsen für die Standardabweichungen  $\sigma_{ki}$  verwendet werden können. Unter der Annahme, dass eine Rotation um die Symmetrieachse einer Gaußfunktion sich nur gering auf die Bedeutung auswirkt, können die Quadrate der Halbachsen als Varianz für die analoge Gaußfunktion verwendet werden. Eine Rotation einer Gaußfunktion um dessen Symmetrieachse besitzt z.B. nur dann geringe Auswirkungen, wenn die Halbachsen ungefähr gleich lang sind, oder die Wahrscheinlichkeitsdichte in Richtung der Halbachsen annähernd die gleichen Parameter wie in Richtung der Eingänge besitzt. Dies ergibt sich für z.B. kleine Drehwinkel.

Für die Länge der Halbachsen gilt [KK00]

$$\sigma_{ki} = \frac{1}{2\sqrt{\lambda_i}} \|\mathbf{v}_i\|_2. \quad (4.35)$$

In (4.35) bezeichnet  $\sigma_{ki}$  die Standardabweichung einer verallgemeinerten Gaußfunktion mit einer Diagonalmatrix als Kovarianzmatrix. Für die Berechnung werden die Eigenwerte  $\lambda_i$  und die dazugehörigen Eigenvektoren  $\mathbf{v}_i$  der Matrix  $\mathbf{W}^T \mathbf{W}$  benötigt. Für eine lokal begrenzte Funktion muss somit die Matrix  $\mathbf{W}^T \mathbf{W}$  positiv definit sein, was in analoger Weise auch für ein verallgemeinertes RBF-Netz zur Bestimmung

der Inversen gilt. Da jede positiv definite Matrix invertierbar ist, folgt hieraus direkt  $\text{rang}(\mathbf{W}^T \mathbf{W}) = n$ . Dies heißt für die Ridge-Funktion, welche die lokalen Funktionen bilden, dass diese nicht die gleiche Richtung im Raum besitzen dürfen.

#### 4.1.2.2 Simulationsergebnisse

Für das Testen der Bedeutung für ein LCNN werden die gleichen Test- und Randbedingungen sowie Approximationsfunktionen wie zuvor bei einem RBF-Netz in Abschnitt 4.1.1 verwendet. Zum Trainieren des LCNNs wird die Software *ELSY* [MS03] verwendet. Hier wird der Backpropagation-Algorithmus genutzt, wobei als Abbruchkriterium ein minimaler, summierter quadratischer Fehler von  $10^{-3}$  oder eine maximale Anzahl von 1000 Epochen gewählt wurde. Im Anschluss an das Training wird die Bedeutung der lokalen Funktionen des LCNNs einmal exakt über die Integration und über den Zusammenhang zu einem RBF-Netz bestimmt, wobei  $q = 2$  gewählt wird. Für die Bedeutung einer lokalen Funktion wird somit die euklidische Distanz definiert.

Zur numerischen Lösung des Integrals aus (4.27) wird die Monte-Carlo-Integration für eine Gleich- und eine Normalverteilung der Eingänge verwendet. Bei den gleichverteilten Eingängen ergibt sich so kein systematischer Fehler, wobei jedoch bei einer Gaußverteilung die Integration über den gesamten Eingangsraum durchgeführt werden muss. Die numerische Auswertung des Integrals erfolgt im Intervall  $[\mu - 3\phi, \mu + 3\phi]$ , was einen systematischen Fehler zur Folge hat. Solange die jeweilige lokale Funktion außerhalb dieses Bereiches sehr kleine Funktionswerte annimmt, kann dieser Fehler klein gehalten werden. Die genäherte Berechnung der Bedeutung einer lokalen Funktion wird mit den exakten Formeln für die jeweilige Verteilung aus Abschnitt 4.1.1 bestimmt, wobei die Halbachsen des Hyperellipsoids für die Varianz verwendet werden. Das weitere Vorgehen entspricht dem Verfahren aus Abschnitt 4.1.1.

Tabelle 4.3 zeigt die Ergebnisse für den auf die Ausgangsleistung des Netzes normierten mittleren quadratischen Fehler aus den Simulationen. Neben dem resultierenden Fehler für den Ausfall des Neurons mit der höchsten oder kleinsten Bedeutung wird in der letzten Spalte angezeigt, ob die exakte Bestimmung des Einflusses mit der genäherten übereinstimmt. Wenn insgesamt 50 % der lokalen Funktionen bei beiden Methoden identisch sind, wird dies als korrekt ('1') angezeigt.

Ähnlich wie bei einem RBF-Netz zeigt sich, dass mit Hilfe der Bedeutung die lokale Funktion gefunden werden kann, die bei einem Ausfall den größten Fehler erzeugt. Bezüglich des minimalen Fehlers ist die Identifikation derjenigen lokalen Funktion aufgrund der numerischen Integration schwieriger als bei einem RBF-Netz. Bei kleinen und ähnlichen Werten der Bedeutung kann der Fehler aufgrund der numerischen Integration die Reihenfolge verschieben. Wie man der Tabelle jedoch entnehmen

**Tabelle 4.3:** Normierter mittlerer quadratischer Fehler für verschiedene Testfunktionen beim LCNN

Funktion	m	nmse		nmse bei Ausfall		Bedeutung
		Training	Test	$\max(s_k)$	$\min(s_k)$	
Gleichverteilung						
Griewank	11	0,0012	0,0012	0,8601	0,0041	1
	12	0,0005	0,0006	0,3659	0,0008	1
Rosenbrock	23	0,0005	0,0057	0,2071	0,0057	0
	22	0,0004	0,0007	0,2643	0,0007	1
Rechenberg	7	0,0010	0,0010	0,1341	0,0103	1
	8	0,0012	0,0013	0,1646	0,0166	0
Schaffer	5	0,2582	0,2711	1,6128	0,2711	1
	6	0,1210	0,1376	0,3746	0,2131	1
Normalverteilung						
Griewank	6	0,0004	0,0018	0,4408	0,0018	1
	5	0,0026	0,0026	0,5730	0,0026	0
Rosenbrock	22	0,0005	0,0402	0,1911	0,0413	0
	23	0,0003	0,0258	0,0938	0,0258	1
Rechenberg	7	0,0030	0,0060	0,2270	0,0237	0
	8	0,0020	0,0093	0,2124	0,0056	0
Schaffer	6	0,0085	0,0100	0,2024	0,0128	1
	5	0,0258	0,0282	0,2945	0,0343	1

kann, ist der Unterschied zwischen dem Fehler bei dem Wegfall des bedeutsamsten Neurons gegenüber dem Geringsten sehr groß.

Bei einem Ausfall einer lokalen Funktion entsteht ein Fehler, der die Approximati-  
onseigenschaften des Netzes erheblich verschlechtert. Die zu Beginn des Trainings  
gewünschten Merkmale für die Approximation, z.B. der mittlere quadratische Fehler,  
steigen bei einem Ausfall im ungünstigsten Fall um zwei Größenordnungen an.  
Auch bei dem Wegfall der lokalen Funktion mit dem geringsten Einfluss verschlech-  
tert sich die Funktionalität des Netzes. Da bei den hier betrachteten Netzen die  
Anzahl von lokalen Funktionen annähernd minimal ist, ergibt sich für diese beiden  
Fälle eine Verschlechterung der Eigenschaften. So kann an dem absoluten Wert der  
Bedeutung die Fehlertoleranz bzw. die Komplexität des Netzes abgelesen werden.  
Bei sehr großen Werten entstehen hohe Fehler bei einem Ausfall, so dass sich für  
besonders tolerante Netze möglichst geringe Ausprägungen ergeben sollten.

Ebenso wurde wie bei den RBF-Netzen die Differenz zwischen der Bedeutung (er-  
warteter mittlerer quadratischer Fehler) und dem Fehler beim Wegfall einer lokalen  
Funktion (realer mittlerer quadratischer Fehler) bestimmt. Hier ergibt sich eine gute  
Übereinstimmung des erwartenden Wertes mit dem Ermittelten für die Gleichver-

teilung der Eingänge. Bei der Normalverteilung ergibt sich eine größere Differenz, da die numerische Integration nur über einen endlichen Bereich ausgeführt wird.

Die genäherte Bestimmung des Einflusses der lokalen Funktionen über den Zusammenhang zu einem verallgemeinerten RBF-Netz kann zur Verbesserung der Fehlertoleranz herangezogen werden. Zwar ergeben sich bei der genäherten Berechnung Fehler in der Reihenfolge, welche aber bei einer geringen Anzahl an lokalen Funktionen und einem hohen Schwellenwert (50 %) besonders anfällig hierfür ist. Weiter muss diese Reihenfolge des Einflusses aufgrund der numerischen Integration nicht exakt sein, da diese bei einer hohen Anzahl von gleichgroßen Werten sich ändern kann. Des Weiteren liefert die ELSY-Software ein LCNN, in welchem die Matrix  $\mathbf{W}$  eine beliebige Form hat, und so keine Diagonalmatrix für  $\mathbf{W}^T \mathbf{W}$  vorliegen muss. So entsteht durch die Verwendung eines gedrehten Hyperellipsoids ein systematischer Fehler, da die Hauptachsen eine neue Basis des Vektorraumes bilden, die Längen der Halbachsen jedoch bzgl. der alten Basis verwendet wurden. Exakterweise muss die Wahrscheinlichkeitsdichte entlang der Halbachsen bestimmt werden und hierüber die Integration durchgeführt werden.

Für eine Fehlerkorrektur ist jedoch die Identifikation der größten Bedeutung wichtig, was durch das genäherte Verfahren erreicht werden kann. Dieses Vorgehen kann bei ähnlich großen Werten nicht korrekt funktionieren, wobei die Auswirkungen für die Betrachtung der Fehlertoleranz gering sind, da beide lokale Funktionen bei einem Wegfall die Approximationseigenschaften des Netzes deutlich herabsetzen. Zur Verbesserung der Fehlertoleranz kann in analoger Weise wie bei einem RBF-Netz vorgegangen werden. So kann die lokale Funktion durch exakte Kopien ersetzt werden (siehe Abschnitt 3.3), wobei die Ausgangsgewichte der lokalen Funktionen angepasst werden oder die NMR-Technik angewendet wird. Die Ergebnisse in Kapitel 3.3 sind unabhängig von der Art des Netzes bestimmt worden und so auch für die lokalen Funktionen des LCNNs gültig. Es können sich für diese jedoch andere Ausfallraten ergeben, die in Abhängigkeit der Architektur bestimmt werden müssen. Mit dieser zusätzlichen Redundanz des Netzes kann die Fehlertoleranz verbessert werden.

### 4.1.3 Kontrolle der Modellkomplexität

Für die neuronalen Netze ist die Kontrolle ihrer Modellkomplexität von besonderer Bedeutung [Moo91]. Durch eine geringe Komplexität kann das Training vereinfacht werden, da die Konvergenzgeschwindigkeit erhöht und die Gefahr durch lokale Minima reduziert werden kann [Hay99, Bis95b]. Diese Kontrolle über die Komplexität kann auf verschiedenen Ebenen des neuronalen Netzes vorgenommen werden. Zum einen kann die Dimension des Eingangesvektors reduziert werden, so dass das Lernverfahren innerhalb eines in der Dimension reduzierten Eingangsraumes arbeitet. Dies vereinfacht besonders das Lernverfahren und kann sich positiv auf die Konver-

genzeigenschaften des Verfahrens auswirken. Zum anderen kann durch die Anzahl der überlagerten Funktionen in der verborgenen Ebene die Komplexität des Modells gesteuert werden. Hier ist eine automatisierte Vorgehensweise wünschenswert, da sich besonders bei der Wahl dieser beiden Parameter große Schwierigkeiten ergeben [Hay99].

Die Zusammenhänge zwischen der Anzahl von Neuronen (Funktionen) und den Eigenschaften des Netzes sind mathematisch komplex und teilweise können Angaben über eine notwendige Anzahl nur für Spezialfälle getroffen werden [Bis95b, Vap98, Kur92]. Dabei können keine exakten mathematischen Aussagen über den Zusammenhang der Approximationsgenauigkeit und der Generalisierungseigenschaft hinsichtlich der Anzahl von Parametern gemacht werden. Zur Bestimmung einer geeigneten Modellkomplexität werden beim Trainieren verschiedene Vorgehensweisen verwendet. So kann von einem kleinen Netz ausgegangen werden und die Anzahl von Funktionen sukzessiv erhöht werden, bis die gewünschten Eigenschaften erfüllt sind. Andererseits kann ein großes Netz stetig in seiner Komplexität verringert werden, um schließlich die erforderliche Genauigkeit zu erreichen. Für beide Verfahren ergeben sich Vor- und Nachteile [Bis95b, Hay99]. Weiter existieren Lernverfahren, welche die Komplexität des Modells während des Trainings z.B. durch einen evolutionären Anteil [GRO<sup>+</sup>03] mit berücksichtigen oder Heuristiken verwenden [MU91].

Durch die in Abschnitt 4.1.1 und 4.1.2 eingeführten Verfahren ist es möglich, die Anzahl von versteckten Elementen für das RBF und das LCNN zu reduzieren, und so einen Kompromiss zwischen Approximations- und Generalisierungseigenschaft herzustellen. Hier kann die auf den maximalen Wert normierte Bedeutung verwendet werden, um die Komplexität des Modells zu verringern. So können z.B. alle Neuronen aus dem Netz entfernt werden, deren Einfluss einen definierten Schwellenwert nicht überschreitet.

Zur Veranschaulichung des Verfahrens wurden verschiedene Simulationen durchgeführt. Um zu zeigen, dass die in dieser Arbeit eingeführte Bedeutung die Komplexität des Netzes verringern kann und sich dies positiv auf die Eigenschaften auswirkt, wurde zunächst für die Testfunktionen aus dem vorherigen Abschnitt ein Netz trainiert. Dafür wurde das Training aus dem vorherigen Abschnitt mit den gleichen Randbedingungen verwendet. Insgesamt wurden für jede Testfunktion 50 verschiedene Netze gebildet. Anschließend wurde die Bedeutung der Neuronen bestimmt, bzgl. ihres Maximums normiert, und alle Neuronen, deren normierte Bedeutung unterhalb von 0,1 lag, wurden aus dem Netz entfernt. Dieses neue Netz wurde erneut trainiert, und der Approximationsfehler wurde für das neue Netz bestimmt.

In einem Hypothesentest [Pap01] ist aus diesen Daten die Hypothese getestet worden, dass sich mit dieser reduzierten Zahl ein kleinerer mittlerer quadratischer Fehler bzgl. der Testdaten ergibt. Als Null-Hypothese wurde dabei

$$H_0 : E(\text{mse}_{\text{original}}) \leq E(\text{mse}_{\text{reduziert}}) \quad (4.36)$$

**Tabelle 4.4:** Ergebnisse des Hypothesentests für zwei RBF-Netze bei Reduktion der Komplexität

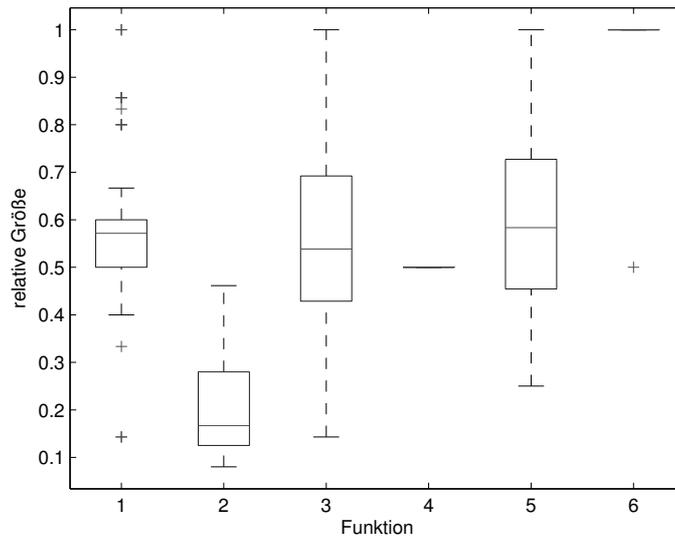
Funktion	Verteilung	Null-Hypothese	p
Rechenberg	Gleichverteilung	zurückweisen	<0,01%
	Normalverteilung	zurückweisen	<0,01%
Rosenbrock	Gleichverteilung	zurückweisen	<0,01%
	Normalverteilung	zurückweisen	<0,01%
Schwefels-Rochen	Gleichverteilung	zurückweisen	<0,01%
	Normalverteilung	zurückweisen	<0,01%
Schwefelfunktion	Gleichverteilung	zurückweisen	<0,01%
	Normalverteilung	zurückweisen	<0,01%
Griewank	Gleichverteilung	zurückweisen	<0,01%
	Normalverteilung	zurückweisen	<0,01%
Schaffer	Gleichverteilung	zurückweisen	<0,01%
	Normalverteilung	akzeptieren	5,49%

aufgestellt und als Signifikanzniveau 2 % gewählt.

Tabelle 4.4 zeigt die Ergebnisse des Hypothesentests. Dargestellt sind die Ergebnisse für die verschiedenen Testfunktionen mit den Wahrscheinlichkeitsverteilungen und das Signifikanzniveau. Man erkennt, dass die eingeführte Bedeutung der Neuronen zur Reduktion der Modellkomplexität verwendet werden kann. Mit verkleinerten Netzen ergeben sich zudem kleinere Approximationsfehler in den Testvektoren. Einzig bei der Schaffer-Funktion bewirkt eine geringere Anzahl von Basisfunktionen keine besseren Eigenschaften des Netzes. Der Grund hierfür ist, dass hier bereits ein kleines Netz verwendet wird. Bei einem minimalen Netz kann eine Verringerung der Basisfunktionen keine Verbesserungen der Approximation erreichen. Da bei der Schaffer-Funktion sehr wenige Basisfunktionen ausreichend sind, kann hier keine Verbesserung erreicht werden, und die Null-Hypothese muss akzeptiert werden.

Abbildung 4.4 zeigt die Häufigkeitsverteilung, den Boxplot, der resultierenden Netzgröße für die Testfunktionen aus Tabelle 4.4. Die Reihenfolge in der Abbildung entspricht der Ordnung innerhalb der Tabelle. Es lässt sich erkennen, dass dieses Vorgehen zu wesentlich kleineren Netzen führt, und der Hypothesentest zeigt zudem, dass diese Netze bessere Approximationseigenschaften besitzen. Die statistische Auswertung für die Schaffer-Funktion in Abbildung 4.4 zeigt die Auswirkungen bei einem minimalen Netz. Die Verringerung der Komplexität durch die Bedeutung ist nur bedingt möglich, da alle Basisfunktionen essentiell zum Ausgang beitragen. So können keine Funktionen aus dem Netz entfernt werden, wie man an der Verteilung in Abbildung 4.4 erkennen kann.

Bisher wurden die Ergebnisse für die radialen Basisfunktionen betrachtet. In Tabelle 4.5 sind die Ergebnisse für das LCNN dargestellt. Für gleichverteilte Eingänge sind die Parameter verschieden großer Netze auf Grundlage der gleichen Voraussetzungen



**Abbildung 4.4:** Relative Größe des Netzes für die Testfunktionen

aus Abschnitt 4.1.2 bestimmt worden. Im Anschluss daran ist auch hier die Anzahl der Neuronen mit einer relativen Bedeutung größer als 10% mit Hilfe der exakten Methode und der Näherung sowie der Mittelwert der Bedeutung aller Neuronen bestimmt worden.

Die ermittelten Netze zeigen bedingt die Eigenschaft eines überangepassten Netzes. Bei der Schwefel-Funktion und einem LCNN mit insgesamt 20 lokalen Funktionen steigt der mittlere quadratische Fehler im Ausgang um den Faktor 10 an, so dass in diesem Fall ein zu komplexes Modell gewählt wurde, und die Parameter in einem lokalen Minimum hängen bleiben. Für die anderen Testfälle liegt der Anstieg des Fehlers bei einem Faktor von 2-5. Diese Erhöhung gegenüber der Testmenge fällt deutlich stärker aus als bei den minimalen Netzen aus Tabelle 4.3, und da bei beiden Experimenten die gleichen Bedingungen verwendet wurden, liegt ein zu komplexes Modell vor.

Wie Tabelle 4.5 zeigt, stimmt die ermittelte Anzahl von lokalen Funktionen für die exakte Methode und dessen Näherung gut überein. Die Abweichungen bei der Rechenberg-Funktion basieren auf dem Fehler, der aufgrund der Drehung der Hyperellipsoide entsteht. Diese Anzahl von lokalen Funktionen kann für ein erneutes Training verwendet werden, wobei die Eigenschaften des Netzes mit dem Vorherigen verglichen werden können. So kann die Komplexität des Netzes reduziert werden. Direkt aus der Anzahl von Neuronen auf eine geeignete Komplexität zu schließen ist nur bedingt möglich, da bei der Definition der Bedeutung diese hinsichtlich der lokalen Funktionen selbst festgelegt wurde. Dies ist mit dem Ziel der Erhöhung der Fehlertoleranz durchgeführt worden, so dass diese Definition keine Aussagen über die Approximationseigenschaften des Netzes enthält.

**Tabelle 4.5:** Reduzierung der Modellkomplexität mit Hilfe der Bedeutung bei einem LCNN

Funktion	m	nmse		Bedeutung		$E\{s\}$
		Training	Test	exakt	vereinfacht	
Griewank	10	0,0005	0,0006	4	4	0,0310
	20	0,0002	0,0009	3	4	0,0164
	30	0,0001	0,0002	3	4	0,0101
	40	0,0001	0,0003	5	5	0,0078
Schwefel	10	0,0005	0,0006	4	4	1,0543
	20	0,0001	0,0016	4	4	0,6343
	30	0,0002	0,0006	12	12	0,2865
	40	0,0002	0,0009	11	12	0,1926
Rechenberg	10	0,0008	0,0010	8	4	29,3992
	20	0,0005	0,0009	12	7	10,9471
	30	0,0005	0,0010	9	5	8,0648
	40	0,0004	0,0013	18	19	5,5930

Wie man der Tabelle 4.5 weiter entnehmen kann, sinkt der Erwartungswert der Bedeutung mit zunehmender Anzahl von Neuronen ab. Dies ist ein Hinweis darauf, dass mit zunehmender Anzahl von lokalen Funktionen das LCNN in seiner Zuverlässigkeit erhöht wird, falls der Ausfall der lokalen Funktionen als gleichwahrscheinlich angenommen werden kann. Somit ermöglicht der Backpropagation-Algorithmus eine Verbesserung der Fehlertoleranz des LCNNs, obwohl hierfür keine speziellen Vorkehrungen innerhalb des Verfahrens getroffen werden. Mit der Wahl eines größeren Netzes können nicht nur die Approximationseigenschaften verbessert werden, sondern auch die Fehlertoleranz. Der Nachteil allerdings gegenüber dem Einfügen von exakten Kopien aus Abschnitt 3.3.2 besteht in der Gefahr eines überangepassten Modells und dem Finden von lokalen Minima durch das Backpropagation-Verfahren. Allerdings verschlechtern sich die Generalisierungseigenschaften des LCNNs nach Tabelle 4.5 nur minimal. Die exakten Kopien vermeiden eine zu hohe Komplexität auf jeden Fall, erreichen jedoch keine verbesserten Eigenschaften im Approximationsfehler.

Zur Bestimmung der Fehlertoleranz war es ausreichend, die Bedeutung einzelner Funktionen in Bezug auf die Übertragungsfunktion des Netzes festzulegen (siehe Abschnitt 4.1.1). Für die Modellkomplexität ist es dagegen sinnvoll, diese hinsichtlich der approximierenden Funktion auszulegen, um so anhand der Parameter des Netzes erkennen zu können, ob ein überangepasstes Modell vorliegt. Diese Definition der Bedeutung kann dies für den Extremfall leisten. Hier werden jeder einzelne Trainingspunkt oder sehr viele durch eine einzelne Basisfunktion beschrieben, so dass die Zentren des Netzes den Eingangsvektoren  $\mathbf{x}^i$  entsprechen und die Aus-

gangsgewichte der jeweiligen Ausgabe  $y^i$ . Die Gaußfunktionen sind sehr schmal und überdecken somit nur einen sehr kleinen Teil des Eingangsraumes. Dadurch ergibt sich eine große Menge von unbedeutenden Neuronen (normiert auf den maximalen Wert), was somit ein überangepasstes Modell anzeigt.

Zur Reduktion der Modellkomplexität kann die Ähnlichkeit zweier Basisfunktionen verwendet werden. Die Gaußfunktionen eines RBF-Netzes reagieren unterschiedlich auf die Eingangsvektoren. Je mehr sich diese Reaktionen auf die Eingaben ähneln, desto redundanter sind beide Basisfunktionen. Ergibt sich eine hohe Übereinstimmung, so kann eine Basisfunktion aus dem Netz entfernt oder die beiden Funktionen durch eine Neue ersetzt werden. In beiden Fällen ergibt sich eine Verringerung der Anzahl von Funktionen im Netz und dadurch eine Reduzierung der Komplexität. So ist das Vorgehen komplementär zu dem Erhöhen der Fehlertoleranz durch Einfügen von Redundanz.

Die Ähnlichkeit zweier Basisfunktionen kann durch die Kreuzkorrelation der beiden Funktionen [Kam04] ausgedrückt werden. Für diese und somit auch für die Ähnlichkeit zweier Funktionen gilt

$$\rho_{ij} = \frac{E \{g_i(\mathbf{x}) \cdot g_j(\mathbf{x})\}}{\sqrt{E \{g_i^2(\mathbf{x})\} \cdot E \{g_j^2(\mathbf{x})\}}}, \quad (4.37)$$

wobei  $g_i(\mathbf{x})$  die jeweilige Gaußfunktion des RBF-Netzes bezeichnet.

Wird (4.37) für die Gaußfunktionen ausgewertet, so ergibt sich für das Ähnlichkeitsmaß zweier Gaußfunktionen

$$\rho_{ij} = \frac{\text{sign}(\alpha_i) \text{sign}(\alpha_j) \sqrt{2\sigma_i\sigma_j} e^{-\frac{\|\mathbf{c}_i - \mathbf{c}_j\|^2}{2(\sigma_i^2 + \sigma_j^2)}}}{\sqrt{\sigma_i^2 + \sigma_j^2}}, \quad (4.38)$$

wobei  $\text{sign}(\cdot)$  die Signum-Funktion darstellt.

Der Korrelationskoeffizient aus (4.38) kann nun dazu verwendet, die Komplexität des RBF-Netzes zu reduzieren, indem Basisfunktionen aus dem Netz entfernt werden, deren Ähnlichkeit einen Schwellenwert übersteigt.

Diese beiden Basisfunktionen werden durch eine einzige Gaußfunktion mit einem verschobenen Zentrum und einer veränderten Varianz ersetzt. Für das neue Zentrum wird der Schwerpunkt des Volumens der Funktion verwendet, welche durch eine Überlagerung der beiden Funktionen beschrieben wird. Für den neuen Mittelpunkt ergibt sich

$$\mathbf{m} = \frac{1}{\alpha_i\sigma_i^n + \alpha_j\sigma_j^n} (\alpha_i\sigma_i^n \mathbf{c}_i + \alpha_j\sigma_j^n \mathbf{c}_j). \quad (4.39)$$

Die Bestimmung des Schwerpunktes des Volumens als neuen Mittelpunkt der neuen Gaußfunktion nach (4.39) ist nur für den Fall

$$\text{sign}(\alpha_i) = \text{sign}(\alpha_j) \quad (4.40)$$

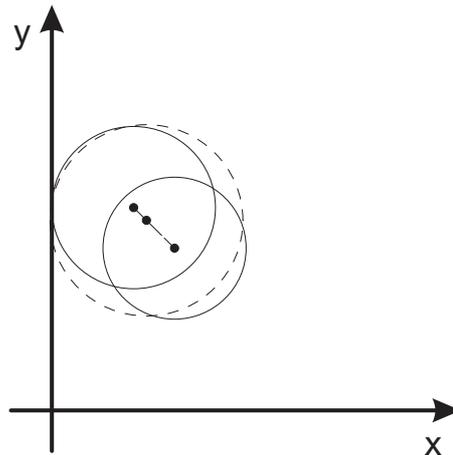
gültig, da sich hier das neue Zentrum gewichtet gegenüber den alten Mittelpunkten berechnet. Daraus folgt, dass der Korrelationskoeffizient aus (4.38) positiv ist. Für gegensätzliche Vorzeichen kann das neue Zentrum weit von den ursprünglichen Mittelpunkten entfernt liegen. Um dies zu vermeiden, werden bei den folgenden Analysen zwei Vorgehen unterschieden. Bei der ersten Variante werden nur Basisfunktionen betrachtet, deren positive Korrelationskoeffizienten einen Schwellenwert übersteigen. Hier wird zwischen einem positiven und einem negativen Korrelationskoeffizienten unterschieden. So ist sichergestellt, dass beide Gaußfunktionen die gleichen Vorzeichen besitzen.

Bei der zweiten Variante wird der Korrelationskoeffizient der Aktivierung der Neuronen bestimmt, d.h., es erfolgt keine Berücksichtigung der Ausgangsgewichte in (4.38). So werden auch die Basisfunktionen mit unterschiedlichen Vorzeichen erfasst. Auch hier werden alle Basisfunktionspaare durch eine neue Gaußfunktion ersetzt, deren Absolutbetrag des Korrelationskoeffizienten einen festgelegten Schwellenwert übersteigt. Zur Bestimmung des neuen Zentrums wird der Schwerpunkt der Fläche gewählt, die durch die Aktivierung der beiden Gaußfunktionen beschrieben wird. Dies bedeutet, dass die Ausgangsgewichte in (4.39) lediglich zu Eins gesetzt werden müssen. So entstehen kleinere Netze als bei dem ersten Verfahren, da insgesamt mehr Basisfunktionspaare erfasst werden. Dieses Vorgehen wird im Folgenden als zweiseitig bezeichnet, während das erste Verfahren als Einseitiges deklariert wird.

Für beide Verfahren wird als Standardabweichung der neuen Gaußfunktion die Summe aus dem Mittelwert der beiden alten Varianzen und dem halben Abstand des neuen Zentrums zu dem näheren Mittelpunkt der alten Basisfunktionen verwendet, d.h. für die neue Varianz folgt

$$\sigma^2 = \left( \frac{\sigma_i + \sigma_j + \min(\|\mathbf{m} - \mathbf{c}_i\|, \|\mathbf{m} - \mathbf{c}_j\|)}{2} \right)^2. \quad (4.41)$$

Abbildung 4.5 zeigt die Wahl des neuen Zentrums und der neuen Varianz für den Fall  $\dim(\mathbf{x}) = 2$ . Eine genauere Approximation erhält man, wenn man unterschiedliche Varianzen für die Koordinatenachsen wählt und ein verallgemeinertes RBF-Netz erhält. So ergibt sich in Abbildung 4.5 kein Kreis, sondern eine Ellipse, welche die Daten besser approximieren kann. Da bei der Simulation mit dem Algorithmus aus NETLAB für alle Koordinaten die gleichen Varianzen verwendet werden, wird für den Vergleich auf ein verallgemeinertes RBF-Netz nach (2.31) verzichtet, da dieses einen kleineren Fehler bei der Approximation erlauben würde.



**Abbildung 4.5:** Wahl des neuen Zentrums und der neuen Varianz aufgrund der Ähnlichkeit von Gaußfunktionen

Die Wahl des neuen Ausgangsgewichtes ist für die Reduzierung der Komplexität von geringer Bedeutung. Für die Bestimmung der Ähnlichkeit bzw. des Korrelationskoeffizienten zweier Basisfunktionen ist nur das Vorzeichen der Gewichte entscheidend. Bei dem zweiseitigen Verfahren ist für die Bestimmung des neuen Mittelpunktes der Wert des Ausgangsgewichtes unwichtig, da nur die Ähnlichkeit der Aktivierung der Neuronen betrachtet wird. Bei dem einseitigen Verfahren wird für das neue Gewicht der Mittelwert aus beiden Ausgangsgewichten verwendet, wobei dieses Vorgehen umso genauer ist, desto höher der Korrelationskoeffizient und damit die Ähnlichkeit beider Basisfunktionen ist. Das neue Ausgangsgewicht ist für eine wiederholte Anwendung dieses Verfahrens notwendig, um das neue Zentrum bei einer hohen Korrelation zwischen dieser neuen Gaußfunktion und anderen Basisfunktionen zu bestimmen.

Bei beiden Verfahren werden die Ausgangsgewichte nach der Minimierung des Netzes mit Hilfe eines überbestimmten Gleichungssystems berechnet. Dazu kann mit Hilfe der Trainingsdaten ein Minimalproblem aufgestellt werden, dessen Lösung mit der Pseudoinversen bestimmt wird [GLL96]

$$\mathbf{H} \cdot \boldsymbol{\alpha} \approx \mathbf{y} \quad \text{mit} \quad \boldsymbol{\alpha} = (\alpha_1, \alpha_1, \dots, \alpha_m)^T, \quad \mathbf{y} = (y^1, y^2, \dots, y^M)^T$$

$$\mathbf{H} = \begin{pmatrix} e^{-\frac{\|\mathbf{x}^1 - \mathbf{c}_1\|^2}{2\sigma_1^2}} & \dots & e^{-\frac{\|\mathbf{x}^1 - \mathbf{c}_m\|^2}{2\sigma_m^2}} \\ \vdots & \ddots & \vdots \\ e^{-\frac{\|\mathbf{x}^M - \mathbf{c}_1\|^2}{2\sigma_1^2}} & \dots & e^{-\frac{\|\mathbf{x}^M - \mathbf{c}_m\|^2}{2\sigma_m^2}} \end{pmatrix} \quad (M \times m)\text{-Matrix}$$

$$\boldsymbol{\alpha} = \mathbf{H}^\dagger \cdot \mathbf{y}, \quad (4.42)$$

wobei  $\mathbf{H}^\dagger$  die pseudoinverse Matrix von  $\mathbf{H}$  darstellt, welche die Aktivierung der Neuronen für alle Trainingsdaten beinhaltet.

*Bemerkung* (geometrische Deutung). Für spezielle RBF-Netze ergibt sich eine geometrische Bedeutung. Beim Training des RBF-Netzes mittels NETLAB wird für alle Gaußfunktionen dieselbe Varianz verwendet. So vereinfacht sich die Berechnung des Korrelationskoeffizienten und aufgrund der Monotonie der exp-Funktion kann der verwendete Schwellenwert zur Entfernung des Basisfunktionspaares in einen äquivalenten Abstand zweier Zentren umgerechnet werden. So werden die Basisfunktionen ausgewählt, deren Abstände dieses Minimum unterschreiten.

Zur simulativen Auswertung der Komplexitätsreduktion mit Hilfe des Korrelationskoeffizienten wurden die gleichen Testfunktionen und das gleiche Vorgehen beim Training wie in Abschnitt 4.1.1 verwendet. Um die Auswirkungen zu erfassen, wurde ein Netz mit 50 Basisfunktionen trainiert und anschließend der Korrelationskoeffizient zwischen den Basisfunktionen bestimmt.

Das Paar von Basisfunktionen, welches den größten (absoluten) Korrelationskoeffizienten besitzt, wird anschließend durch eine neue Gaußfunktion ersetzt, wobei die Parameter nach (4.39) und (4.41) bestimmt werden. Das neue Ausgangsgewicht wird durch den Mittelwert der beiden ursprünglichen Ausgangsgewichte gebildet. Dieses Vorgehen wird solange wiederholt, bis der Korrelationskoeffizient unter einen definierten Schwellenwert fällt. Im Anschluss daran werden mit Hilfe der Trainingsdaten die Ausgangsgewichte bestimmt, indem das Gleichungssystem mit Hilfe der Pseudoinversen gelöst wird (vgl. (4.42)).

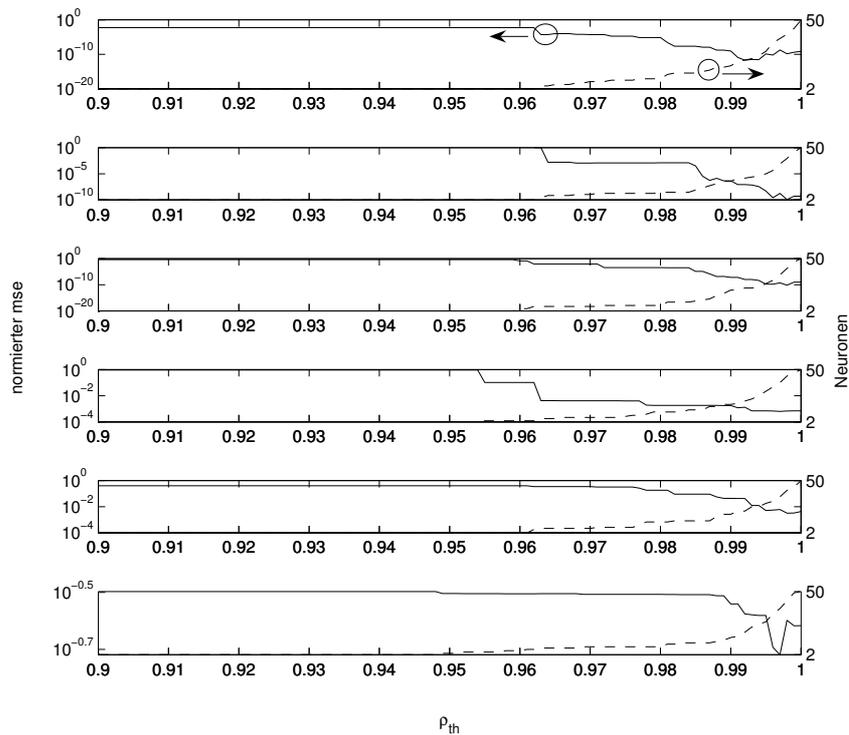
Durch die Berechnung des Korrelationskoeffizienten und somit der Ähnlichkeit von zwei Basisfunktionen kann das Modell in der Komplexität reduziert werden. Einen wichtigen Einfluss zur Reduktion des Netzes besitzt dabei die Wahl des Schwellenwertes  $\rho_{th}$ , für den die Basisfunktionen aus dem Netz entfernt werden.

Abbildung 4.6 zeigt den auf die Ausgangsleistung normierten mittleren quadratischen Fehler sowie die Netzgröße in Abhängigkeit des verwendeten Schwellenwertes  $\rho_{th}$  für verschiedene Testfunktionen<sup>1</sup>. Dabei werden alle Basisfunktionspaare beginnend bei maximaler Korrelation durch eine neue Gaußfunktion ersetzt, welche einen Schwellenwert überschreiten. Es handelt sich somit um das einseitige Verfahren.

Bei der Schaffer-Funktion, dem untersten Graphen in Abbildung 4.6, nimmt der Approximationsfehler ab, bis dieser wieder ansteigt, obwohl die Größe des Netzes zunimmt. Dieses Verhalten basiert auf dem verwendeten Trainingsverfahren. Der Algorithmus aus NETLAB verwendet den maximalen, quadratischen Abstand zweier Zentren als Varianz, so dass sehr große Breiten für die Gaußfunktionen verwendet werden. Dies ist aber aufgrund ihrer Form für die Schaffer-Funktion ungeeignet, da mit hohen Breiten die Konturen der Funktion nicht nachgebildet werden können. Deshalb existiert hier ein Minimum der Approximationsgüte in Abhängigkeit der

---

<sup>1</sup> Die Reihenfolge der Testfunktionen von oben nach unten in Abbildung 4.6 entspricht der gleichen Reihenfolge wie in Tabelle 4.6.



**Abbildung 4.6:** Netzgröße und Approximationsgenauigkeit in Abhängigkeit des Schwellenwertes  $\rho_{th}$

Netzgröße, da bei kleinen Netzen die Wahrscheinlichkeit kleiner Varianzen aufgrund des eingesetzten Verfahrens höher ist.

**Bemerkung.** Der Schwellenwert für den Korrelationskoeffizienten musste in Abbildung 4.6 sehr groß gewählt werden, um neben einer deutlichen Reduzierung der Modellkomplexität auch mäßige Verkleinerungen zu erzielen. Auch dies ist in der speziellen Vorgehensweise von NETLAB begründet. Bei einer hohen Anzahl von Neuronen werden sehr große Varianzen verwendet, um ein überangepasstes Modell zu vermeiden. Damit folgt für den Korrelationskoeffizienten aus (4.38)

$$\rho_{ij} \approx \text{sign}(\alpha_i) \text{sign}(\alpha_j) \quad (4.43)$$

und somit müssen sehr hohe Schwellenwerte verwendet werden, um das Netz nicht auf eine einzige Gaußfunktion zusammenschrumpfen zu lassen. Bei den Simulationen zeigte sich, dass für eine automatisierte Wahl dieses Schwellenwertes der Median des Abstandes aller Zentren zueinander geeignet ist. Für jedes Zentrum kann der Median der Distanz zu den anderen Zentren bestimmt werden, und das Minimum von allen Basisfunktionen dieses Wertes führt nach Anwendung der Gaußfunktion auf einen geeigneten Schwellenwert.

**Tabelle 4.6:** Fehler der Approximation bei der Reduzierung der Modellkomplexität mittels der Ähnlichkeit zweier Gaußfunktionen

Funktion	nmse RBF-Netz			m	H	p
	Original	Referenz	Reduz.			
Rechenberg (zweiseitig)	$1,06 \cdot 10^{-9}$	$1,17 \cdot 10^{-8}$	$4,33 \cdot 10^{-9}$	18,68	1	< 0,01%
	$8,65 \cdot 10^{-10}$	$3,00 \cdot 10^{-5}$	$1,48 \cdot 10^{-5}$	9,12	0	5,14%
Rosenbrock (zweiseitig)	$5,60 \cdot 10^{-10}$	$8,87 \cdot 10^{-6}$	$3,00 \cdot 10^{-7}$	18	1	0,46%
	$1,27 \cdot 10^{-10}$	0,0017	0,0007	8,88	1	< 0,01%
Schwefels-Rochen (zweiseitig)	$8,45 \cdot 10^{-10}$	$1,90 \cdot 10^{-7}$	$1,17 \cdot 10^{-7}$	18,98	1	< 0,01%
	$1,87 \cdot 10^{-9}$	0,0009	0,0003	9,56	1	< 0,01%
Schwefel (zweiseitig)	0,0007	0,0015	0,0013	18,48	1	0,18%
	0,0008	0,0024	0,0026	8,82	0	89,98%
Griewank (zweiseitig)	0,0069	0,0561	0,0535	18,7	0	24,12%
	0,0075	0,2551	0,2881	8,88	0	100%
Schaffer (zweiseitig)	0,2285	0,2567	0,2565	17,96	0	47,09%
	0,2450	0,3052	0,3074	8,8	0	100%

Um weiter die Auswirkungen dieser Methode zu testen, wurde für die gleichen Testfunktionen ein RBF-Netz mit 50 Gaußfunktionen trainiert. Anschließend wurde die Ähnlichkeit der Basisfunktionen bestimmt und so das Netz mit der einseitigen und zweiseitigen Variante minimiert. Als Schwellenwert für diese beiden Verfahren wurde  $\rho_{th} = 0,99$  gewählt. Die Parameter des minimierten Netzes wurden wie zuvor bestimmt. Daneben wurde ein weiteres Netz mit Hilfe von NETLAB trainiert, welches die gleiche Modellkomplexität wie das mit dem Korrelationskoeffizienten minimierte Netz besitzt. Für beide Netze wurde der auf die Ausgangssignalleistung normierte Fehler bestimmt.

Tabelle 4.6 zeigt den Mittelwert der Auswertung für insgesamt 50 Durchläufe jeder einzelnen Testfunktion. Wie erwartet ergibt sich für das ursprüngliche RBF-Netz ein sehr kleiner Approximationsfehler im Ausgang. Es lassen sich jedoch akzeptable Werte bei einer halben Netzgröße finden, wobei das Referenznetz mit NETLAB bestimmt wurde. Zusätzlich zu der Größe des Netzes enthält Tabelle 4.6 ebenfalls die Ergebnisse eines Hypothesentests. Als Null-Hypothese dieses statistischen Tests wurde

$$H_0 : E\{\text{mse}_{\text{ref}}\} \leq E\{\text{mse}_{\text{orig}}\} \quad (4.44)$$

aufgestellt und ein Signifikanzniveau von 2% verwendet. Die Nullhypothese wird im Falle  $H = 1$  verworfen und für  $H = 0$  beibehalten. Zusätzlich ist das Signifikanzniveau zu Beibehaltung der Nullhypothese angegeben<sup>2</sup>.

<sup>2</sup> Somit ist für die Rosenbrock-Funktion in 0,46% aller Fälle der Approximationsfehler des NETLAB-Netzes geringer als des minimierten Netzwerkes.

Tabelle 4.6 zeigt, dass mit der Ähnlichkeit der Basisfunktionen über den Korrelationskoeffizienten effiziente Netze gefunden werden können, d.h. Netze mit einer kleineren Anzahl von Neuronen aber ähnlichen Approximationseigenschaften. Besonders im Falle des einseitigen Verfahrens kann die Anzahl von Basisfunktionen halbiert werden, ohne unzulässig hohe Approximationsfehler im Ausgang zu erhalten<sup>3</sup>. Somit reduziert sich ebenfalls die Anzahl der freien Parameter des Netzes, die zur Approximation bestimmt werden müssen, um die Hälfte, und die Komplexität des Optimierungsproblems wird verringert.

Die Minimierung des Netzes mit gleichzeitiger Festlegung der Zentren und Varianzen über dieses Verfahren liefert bessere oder ähnlich gute Werte wie ein Training mittels NETLAB. So ist es möglich, eine bestehende Lösung einer Funktionsapproximation mittels der Ähnlichkeit der Basisfunktionen in seiner Komplexität zu reduzieren. Je nach Güte der vorherigen Lösung muss jedoch mit einem größeren Approximationsfehler gerechnet werden. Die Wahl eines geeigneten Schwellenwertes spielt eine große Rolle. Wie in Abbildung 4.6 bereits dargestellt, besteht bei einem zu kleinen Schwellenwert für den Korrelationskoeffizienten die Gefahr eines zu kleinen Netzes, welches eine angemessene Approximation nicht leisten kann. Die Wahl eines zu hohen Schwellenwertes liefert im ungünstigsten Fall keine Verringerung der Modellkomplexität, so dass möglichst hohe Schwellenwerte gewählt werden sollten.

Tabelle 4.6 zeigt, wie schon erläutert, dass bei dem zweiseitigen Verfahren deutlich kleinere Netze entstehen, da hier das Vorzeichen der Korrelation nicht betrachtet wird. So werden mehr Basisfunktionspaare bei der Minimierung mit einbezogen, wodurch sich ein höheres Potential bei der Reduzierung der Modellkomplexität ergibt. So kann nochmals die Anzahl von überlagerten Funktionen im Netz gegenüber dem einseitigen Verfahren annähernd halbiert werden. Bei dem einseitigen Vorgehen wird gegenüber dem Referenznetz für insgesamt vier der Testfunktionen ein geringerer Fehler erreicht, wenn die Wahl der Parameter auf die hier vorgestellte Art erfolgt. In den anderen beiden Fällen ergibt sich ein maximaler relativer Fehler von 4,7% gegenüber der Referenzlösung von NETLAB.

**Bemerkung.** Bei dem vorgestellten Verfahren wird ein RBF-Netz minimiert, welches bereits trainiert worden ist. In Abschnitt 4.3 werden Methoden zur Verbesserung der Robustheit und der Fehlertoleranz während des Lernens vorgestellt. In ähnlicher Weise kann der Korrelationskoeffizient während des Trainings eingesetzt werden, um die Komplexität des Netzes zu begrenzen. So kann als zusätzliche Randbedingung bei dem Optimierungsproblem ein maximaler Korrelationskoeffizient eingeführt werden. Bei gleichen Varianzen bedeutet dies geometrisch, dass die Zentren einen bestimmten Abstand nicht unterschreiten dürfen oder, bei unterschiedlichen

<sup>3</sup> Je nach Testfunktion, wie z.B. der Rosenbrock-Funktion, kann der Approximationsfehler stark ansteigen. Dies liegt an den großen Funktionswerten, so dass eine Reduzierung der Anzahl von Gaußfunktionen den Fehler stark erhöht. Der normierte Fehler liegt dennoch in zulässigen Bereichen.

Varianzen, dass die mit den Varianzen gewichteten Abstände einen Schwellenwert nicht unterschreiten dürfen. So wird für eine Menge von Trainingsdaten die maximale Anzahl von Funktionen begrenzt.

Neben der Verringerung von überlagerten Funktionen kann die Dimension des Eingangsraumes reduziert werden, was ebenfalls die Komplexität des Modells verringert. Die Möglichkeit, die Dimension des Einganges zu reduzieren, ist von dem Szenarium abhängig und unterscheidet sich in seinen Konsequenzen. Der Eingangsvektor  $\boldsymbol{x}$  kann als Zufallsvektor aufgefasst werden, ebenso wie der Ausgang des neuronalen Netzes als Zufallsvariable. Ist es nun die Aufgabe des Netzes den (nicht-linearen) Zusammenhang zwischen Eingangsvariablen und Ausgangsvariablen zu erlernen, so bietet es sich an, die Abhängigkeit zwischen beiden Zufallsvariablen zu bestimmen. Dies ist besonders dann sinnvoll, wenn es sich um Sensordaten handelt, die zum einen in großer Menge und zum anderen in hoher Dimensionalität<sup>4</sup> zur Verfügung stehen.

Besteht keine Abhängigkeit z.B. zwischen der Zufallsvariablen  $X_k$  und der Zufallsvariablen des Ausgangs  $Y$ , kann dieser Sensorwert oder dieses Signal für das Lernen des nichtlinearen Zusammenhanges vernachlässigt werden. Eine Möglichkeit zur Reduktion des Eingangsraumes wurde z.B. in [FWV06] vorgestellt, wo die signifikanten Zusammenhänge zwischen zwei Zufallsvariablen bestimmt werden. Alle nicht signifikanten Zufallsvariablen können so bei dem Entwurf und dem Training des neuronalen Netzes vernachlässigt werden.

## 4.2 Erhöhen der Robustheit

Nachdem durch die Bedeutung eines Neurons Aussagen über das Verhalten bei einem Wegfall dieser elementaren Recheneinheiten gemacht werden können, soll nun eine analoge Betrachtungsweise für das Rauschen vorgestellt werden. Additives, gaußförmiges Rauschen überlagert die Eingänge und die Parameter des Netzes, wobei diejenigen Elemente mit den größten Auswirkungen auf den Ausgang bestimmt werden sollen.

### 4.2.1 Empfindliche Parameter

Additives Rauschen überlagert jeden Eingang und jeden Parameter des Basisfunktionsnetzwerkes. In Kapitel 3.1 wurde bei der Berechnung angenommen, dass alle Eingänge und Parameter des Netzes durch identische Rauschprozesse gestört und überlagert werden, und deren Einfluss auf den mittleren quadratischen Fehler wurde

---

<sup>4</sup> Aufgrund der Miniaturisierung können sehr viele verschiedene Signale aufgenommen werden, was zu einem hochdimensionalen Eingangsvektor führt.

bestimmt. Allerdings können die bei der Analyse als gestört angenommenen Variablen durch unterschiedliche Rauschprozesse mit unterschiedlichen Mittelwerten und Varianzen überlagert werden. Somit hängt der Einfluss der jeweiligen Variablen zusätzlich von den Eigenschaften des Rauschens ab. In diesem Abschnitt wird dafür eine Methode vorgestellt, die diese Auswirkungen auf den Ausgangswert des Netzes bestimmt. Auch hier wird angenommen, dass die Rauschprozesse voneinander nicht abhängen und somit stochastisch unabhängig sind.

Mit diesen Annahmen kann der absolute Fehler einer Funktion  $f$  mit Hilfe der Taylorreihe und den Mittelwerten der einzelnen Zufallsvariablen bestimmt werden, welches auch als gaußsche Fehlerfortpflanzung bekannt ist [BSM05]. Somit kann die Varianz des Funktionswertes  $f$  ausgedrückt werden durch

$$\sigma_f^2 = \sum_{i=1}^M \left( \left. \frac{\partial f}{\partial w_i} \right|_{\mathbf{w}=\boldsymbol{\mu}} \right)^2 \sigma_{w_i}^2, \quad (4.45)$$

wobei  $\boldsymbol{\mu}$  den Erwartungswert der Eingangsvariablen  $w_i$  und  $M$  die Gesamtanzahl der gestörten Variablen bezeichnet.

(4.45) stellt für einen linearen Zusammenhang zwischen Funktionswert und Eingangsvariable eine exakte Abschätzung dar. Für nichtlineare Zusammenhänge hingegen repräsentiert (4.45) nur eine Approximation der Varianz, die aber für praktische Fälle ausreichend ist [BSM05].

Die Varianz des Ausgangs bezeichnet die mittlere Abweichung vom Funktionswert  $f$ , wenn eine oder alle Eingangsvariablen durch Abweichungen gestört werden. Als Eingangsvariablen für den Ausgangswert  $f_m(\mathbf{x})$  eines neuronalen Netzes können zum einen der Eingangsvektor  $\mathbf{x}$  und zusätzlich die Parameter des Basisfunktionsnetzwerkes, wie Ausgangsgewichte, Varianzen und Zentren, dienen. Somit können alle Parameter und Eingänge aus (2.28) als Eingangsvariablen für diese Analyse dienen.

Unter Berücksichtigung von (4.45) kann gefolgert werden, dass der größte Summand den höchsten Einfluss auf die Varianz des Ausgangs besitzt und damit als der Parameter oder Eingang mit dem empfindlichsten Verhalten gegenüber Rauschen angesehen werden kann. Somit kann dieser als

$$\Gamma = \left\| \left( \left. \frac{\partial f}{\partial w_i} \right|_{\mathbf{w}=\boldsymbol{\mu}} \right)^2 \sigma_{w_i}^2 \right\|_{\infty} = \max_{i=1 \dots M} \left\{ \left( \left. \frac{\partial f}{\partial w_i} \right|_{\mathbf{w}=\boldsymbol{\mu}} \right)^2 \sigma_{w_i}^2 \right\} \quad (4.46)$$

definiert werden.

Wird (4.45) auf die Netzwerkfunktion eines RBF-Netzes aus (2.28) angewendet und wird beachtet, dass sowohl der Eingangsvektor als auch die Parameter als Variablen zur Analyse betrachtet werden können, so folgt für die Eingänge

$$\sigma_{f_x}^2 \approx \sum_{i=1}^n \left( \left. \frac{\partial f_m(\mathbf{x})}{\partial x_i} \right|_{\mathbf{x}=\boldsymbol{\mu}} \right)^2 \sigma_{x_i}^2 \quad (4.47)$$

$$= \sum_{i=1}^n \left( \sum_{\nu=1}^m -\alpha_\nu \frac{(x_i - c_{\nu i})}{\sigma_\nu^2} e^{-\frac{\|\mathbf{x} - \mathbf{c}_\nu\|^2}{2\sigma_\nu^2}} \Big|_{\mathbf{x}=\boldsymbol{\mu}} \right)^2 \sigma_{x_i}^2, \quad (4.48)$$

wobei  $\sigma_{f_x}^2$  die Varianz des Ausganges aufgrund gestörter Eingänge darstellt.

Deshalb kann (4.48) dazu benutzt werden, den Eingang mit der höchsten Empfindlichkeit gegenüber Rauschen zu identifizieren. Aufgrund des nichtlinearen Zusammenhanges stellt (4.48) nur eine Abschätzung hinsichtlich der Varianz im Ausgang dar. In der gleichen Art und Weise können die Parameter des Netzes hinsichtlich ihrer Empfindlichkeit gegenüber Störungen durch Rauschen betrachtet werden. Für die Ausgangsgewichte  $\alpha_i$  des RBF-Netzes folgt somit für die Varianz des Ausganges  $\sigma_{f_\alpha}^2$

$$\sigma_{f_\alpha}^2 = \sum_{i=1}^m \left( \left. \frac{\partial f_m(\mathbf{x})}{\partial \alpha_i} \right|_{\alpha_i=\mu_i} \right)^2 \sigma_{\alpha_i}^2 \quad (4.49)$$

$$= \sum_{i=1}^m e^{-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\sigma_i^2}} \sigma_{\alpha_i}^2, \quad (4.50)$$

wobei (4.50) für die Ausgangsgewichte exakt ist, da es sich um einen linearen Zusammenhang handelt.

Die gleiche Analyse kann für die Zentren der Basisfunktionen und ihre Breiten durchgeführt werden, was zu den Ergebnissen der Varianz im Ausgang  $\sigma_{f_\sigma}^2$  und  $\sigma_{f_{x_k}}^2$  führt

$$\sigma_{f_\sigma}^2 \approx \sum_{i=1}^m \left( \left. \frac{\partial f_m(\mathbf{x})}{\partial \sigma_i} \right|_{\sigma_i=\mu_i} \right)^2 \sigma_{\sigma_i}^2 \quad (4.51)$$

$$= \sum_{i=1}^m e^{-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\sigma_i^2}} \left( \alpha_i \frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\sigma_i^3} \right)^2 \sigma_{\sigma_i}^2 \quad (4.52)$$

$$\sigma_{f_{c_k}}^2 \approx \sum_{\nu} \left( \left. \frac{\partial f_m(\mathbf{x})}{\partial c_{k\nu}} \right|_{c_{k\nu}=\mu_{k\nu}} \right)^2 \sigma_{c_{k\nu}}^2 \quad (4.53)$$

$$= \sum_{\nu=1}^n \sum_{k=1}^m \left( \alpha_k \frac{(x_\nu - c_{k\nu})}{\sigma_k^2} \right)^2 e^{-\frac{\|\mathbf{x} - \mathbf{c}_k\|^2}{\sigma_k^2}} \sigma_{c_{k\nu}}^2. \quad (4.54)$$

Normalerweise sind die Erwartungswerte der Eingänge und freien Parameter eines RBF-Netzes bekannt. Für die Eingänge ist zumeist die Verteilungsfunktion der Test- oder der Lerndaten verfügbar. Somit lassen sich deren Erwartungswerte berechnen. Bei den freien Parametern sind diese noch einfacher zu erhalten. Diese werden durch das Trainingsverfahren bestimmt und damit stellen diese Werte des Trainings die Erwartungswerte der Parameter dar. Dadurch kann die gesamte Varianz des Ausgangs aufgrund rauschender Gewichte und Parameter berechnet werden, wobei die Rauschprozesse alle unterschiedliche Mittelwerte und Varianzen aufweisen können. Durch Anwendung von (4.48), (4.50), (4.52) und (4.54) ergibt sich für die Varianz des Ausgangs bei einem RBF-Netzwerk

$$\begin{aligned}
\sigma_{f_m}^2 &= \sum_{i=1}^n \left( \sum_{\nu=1}^m -\alpha_{\nu} \frac{(x_i - c_{\nu i})}{\sigma_{\nu}^2} e^{-\frac{\|\mathbf{x} - \mathbf{c}_{\nu}\|^2}{2\sigma_{\nu}^2}} \Big|_{\mathbf{x}=\boldsymbol{\mu}} \right)^2 \sigma_{x_i}^2 + \sum_{i=1}^m e^{-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\sigma_i^2}} \sigma_{\alpha_i}^2 \\
&\quad + \sum_{i=1}^m e^{-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\sigma_i^2}} \left( \alpha_i \frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\sigma_i^3} \right)^2 \sigma_{\sigma_i}^2 \\
&\quad + \sum_{\nu=1}^n \sum_{k=1}^m \left( \alpha_k \frac{(x_{\nu} - c_{k\nu})}{\sigma_k^2} \right)^2 e^{-\frac{\|\mathbf{x} - \mathbf{c}_k\|^2}{\sigma_k^2}} \sigma_{c_{k\nu}}^2. \tag{4.55}
\end{aligned}$$

Zusammenfassend kann festgehalten werden, dass der Parameter oder Eingang mit der größten Empfindlichkeit im besonderen Maße von den Ausgangsgewichten der Basisfunktionen und ihren Breiten abhängig ist, da die Gaußfunktion nur Werte innerhalb eines Intervalls  $[0; 1]$  annehmen kann. Deshalb sind besonders die Parameter von Basisfunktionen betroffen, deren Ausgangsgewichte  $\alpha$  große Werte und deren Breiten  $\sigma^2$  sehr kleine Werte annehmen und somit sehr viel zum Rauschen im Ausgang beitragen. Um im Ausgang nur geringe Abweichungen  $\sigma_{f_m}$  zuzulassen, sollten nur kleine Ausgangsgewichte und möglichst große Breiten verwendet werden. Allerdings kann dies durch das verwendete Trainingsverfahren nur bedingt möglich sein, um weiterhin kleine mittlere quadratische Fehler beim Lernen zu erreichen.

Jedoch kann der verwendete Lernalgorithmus angepasst werden, so dass sich nur kleine Rauschleistungen im Ausgang ergeben (siehe Abschnitt 4.3). Hiermit lässt sich als Randbedingung für das Trainieren des neuronalen Netzes zusätzlich eine Beschränkung der Ausgangsgewichte bzw. eine Beschränkung der Rauschleistung im Ausgang einführen. Dies führt zu einem zusätzlichen Einfügen von Basisfunktionen mit gleichen Zentren und Breiten, deren Ausgangsgewichte einen bestimmten Wert nicht überschreiten. Eine Beschränkung der Varianzen ist nur schwer möglich, da diese zumeist vor dem eigentlichen Trainieren zusammen mit der Position der Zentren bestimmt werden. Eine Vergrößerung der Breite der Basisfunktionen wirkt sich über einen größeren Bereich des Eingangsraumes aus, so dass die Antworten der Neuronen nur noch bedingt als lokal angesehen werden können.

Derjenige Parameter oder Eingang  $\Gamma$ , der den größten Anteil an der Rauschleistung des Ausgangs besitzt, lässt sich durch Anwenden der Maximumsnorm (4.46) auf (4.55) bestimmen. In Abhängigkeit der Art des empfindlichsten Parameters (Eingang, Ausgangsgewicht, Breite oder Zentrum) können verschiedene Methoden angewendet werden, um dessen Einfluss auf den Ausgang zu minimieren.

## 4.2.2 Empfindliche Neuronen

In Abschnitt 4.2.1 wurden empfindliche Parameter eines RBF-Netzes bestimmt, um die Zuverlässigkeit des Netzes zu erhöhen und die Rauschleistung im Ausgang zu verringern. Bei sehr großen Netzen steigt der dafür erforderliche Berechnungsaufwand stark an. Weiter ist dies für den Einsatz neuronaler Netze in Hardwarerealisierungen aufgrund der sehr großen Anzahl von Basisfunktionen beim Einsatz in Nanotechnologien nur schwer durchführbar. Um alle Parameter und Eingänge auszuwerten, müssen insgesamt  $m+m+m \cdot n+n = 2m+n(m+1)$  Variablen ausgewertet werden. Wie schon im Kapitel 4.2.1 erwähnt, sind besonders Parameter und Eingänge von Basisfunktionen mit großen Ausgangsgewichten und kleinen Varianzen für Rauschen anfällig. Werden nun die empfindlichsten Neuronen bestimmt, so reduziert sich der Berechnungsaufwand auf nur noch  $m$  Basisfunktionen innerhalb des Netzes, was eine deutliche Reduktion des Berechnungsaufwandes bedeutet. Auch ist es einfacher ein empfindliches Neuron zu bestimmen, da eine Anwendung von Methoden zur Verbesserung der Empfindlichkeit auf jeden Parameter hinsichtlich des Flächenbedarfs schlecht handhabbar ist.

Basierend auf der gaußschen Fehlerfortpflanzung und der Taylorreihe lassen sich empfindliche Neuronen gegenüber Rauschen in der gleichen Art und Weise wie in Abschnitt 4.2.1 bestimmen. Da nun empfindliche Neuronen bestimmt werden sollen, kann jede Übertragungsfunktion eines einzelnen Neurons durch

$$g_k = \alpha_k e^{-\frac{\|x-c_k\|^2}{2\sigma_k}} \quad \forall k = 1 \dots m \quad (4.56)$$

ausgedrückt werden.

Zusätzlich kann das Rauschen als lineare Überlagerung der einzelnen Parameter und Eingänge angesehen werden, so dass folgt

$$\hat{x}_i = x_i + \Delta x_i \quad (4.57)$$

$$\hat{w}_i = w_i + \Delta w_i, \quad (4.58)$$

wobei  $x_i$  die Eingänge darstellen und  $w_i$  jeder beliebige Parameter des RBF-Netzes sein kann. Somit kann die Länge des Gradienten von (4.56) dazu benutzt werden, um als Maß für die Empfindlichkeit der Neuronen zu dienen [BSM05].

Diese zusätzlichen linearen Terme durch das Rauschen erzeugen einen absoluten Fehler im Ausgang des neuronalen Netzes, wo sich alle Störungen der Aktivierungen der einzelnen Neuronen summieren. Treten nur kleine Störungen  $\Delta x_i$  auf, so kann der Ausgangswert der Neuronen durch die Taylorreihe angenähert werden, wobei nach dem linearen Term die Näherung abgebrochen wird. Es folgt somit

$$g_k(\mathbf{x} + \mathbf{h}) = g_k(\mathbf{x}) + \nabla g_k(\mathbf{x})^T \mathbf{h} \quad \forall k = 1 \dots m, \quad (4.59)$$

wobei  $\mathbf{h}$  das Rauschen in den Eingängen der Neuronen darstellt. In (4.59) werden nur die linearen Terme betrachtet, was der Approximation durch eine Tangente in einem Punkt entspricht. Die Anwendung der Taylor-Formel und der Abbruch nach dem linearen Term sind zulässig, da bei der Verwendung von höheren Termen auch höhere Potenzen des Rauschens auftreten. Die Darstellung mit Hilfe der Taylorreihe für eine Funktion  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  lässt sich durch

$$f(\mathbf{x} + \mathbf{h}) = f(\mathbf{x}) + \sum_{i=1}^N \frac{1}{i!} \left( \frac{\partial}{\partial x_1} h_1 + \frac{\partial}{\partial x_2} h_2 + \dots + \frac{\partial}{\partial x_n} h_n \right)^i f(\mathbf{x}) + R_N \quad (4.60)$$

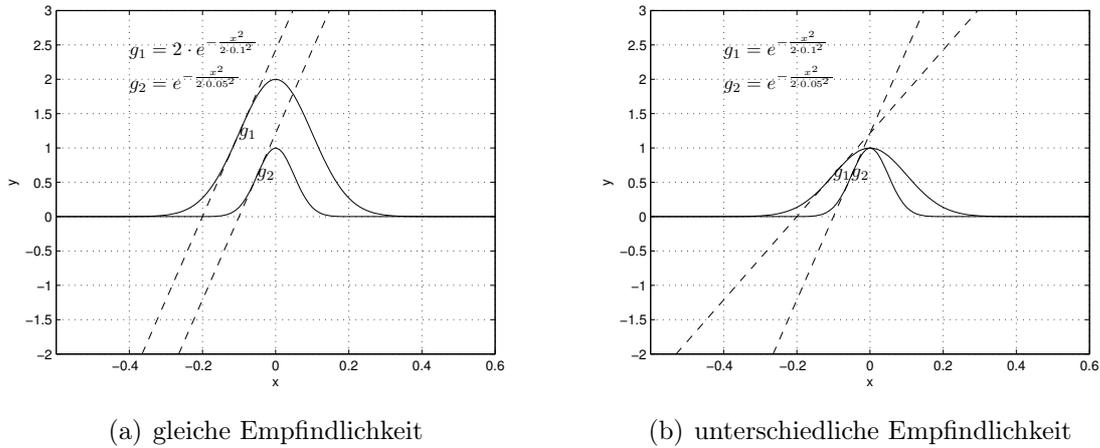
beschreiben, wobei  $R_N$  das Restglied bezeichnet.

Da Rauschen als eine additive Störung aufgefasst werden kann und nur sehr kleine Störungen im Eingang hervorruft, können die Terme höherer Ordnung aus (4.60) vernachlässigt werden. Deshalb kann das System durch eine Tangente beschrieben werden, wobei die Steigung die aktuelle Verstärkung bezeichnet. Somit bestimmt diese Steigung die Empfindlichkeit des Systems und somit des Neurons und führt zu großen Änderungen im Ausgang für den Fall hoher Verstärkungen. Somit ist diese Analyse mit der Kleinsignalanalyse zu vergleichen [Raz00].

Hierdurch kann die Differenz zwischen dem korrekten Ausgangswert und dem durch Rauschen gestörten Wert berechnet werden, wobei gilt

$$\Delta g_k = g_k(\mathbf{x}) - g_k(\mathbf{x} + \mathbf{h}) = -\nabla g_k(\mathbf{x})^T \mathbf{h} \quad \forall k = 1 \dots m. \quad (4.61)$$

Durch die Tangente kann der absolute Fehler im Ausgang bei kleinen Abweichungen im Eingang berechnet werden. Als Ergebnis von (4.61) trägt jenes Neuron am meisten zu dem absoluten Fehler im Ausgang bei, dessen Gradient die größte Länge besitzt und somit die Tangente in einem beliebigen Punkt die größtmögliche Steigung besitzt. Es wird angenommen, dass das zusätzliche Rauschen  $\mathbf{h}$  in den Eingängen für alle Neuronen aus (4.56) gleich ist. Diese Annahme ist gerechtfertigt, solange die Eingänge selbst durch das Rauschen gestört werden und nicht die Verbindungen zwischen den Eingängen und den Neuronen. Dadurch kann das Neuron als die empfindlichste Basisfunktion identifiziert werden, dessen Tangente in einem Punkt des Eingangsraumes den maximalen Wert annimmt.



**Abbildung 4.7:** Definition der Sensibilität von Neuronen eines RBF-Netzes

Abbildung 4.7 zeigt die Approximation einer Gaußfunktion, einer Basisfunktion, jeweils in dem Punkt, in dem die Tangente die maximale Steigung besitzt. In Abbildung 4.7(a) sind beide Steigungen identisch, wodurch eine gleiche Empfindlichkeit gegenüber Rauschen erreicht wird, während in Abbildung 4.7(b) die beiden Neuronen unterschiedliche Empfindlichkeiten gegenüber Rauschen zeigen. Wie aus den beiden Abbildungen entnommen werden kann, spielen besonders das Ausgangsgewicht und die Breite der Basisfunktion eine zentrale Rolle bei der Empfindlichkeit der Neuronen.

Für die Definition der Empfindlichkeit eines Neurons kann der Gradient mit Hilfe von (4.56) zu

$$\nabla^T g_k(\mathbf{x}) = \begin{pmatrix} \frac{\partial g_k}{\partial x_1} \\ \vdots \\ \frac{\partial g_k}{\partial x_n} \end{pmatrix} = -\frac{\alpha_k}{\sigma_k^2} \exp(\cdot) \begin{pmatrix} x_1 - c_{k1} \\ \vdots \\ x_1 - c_{kn} \end{pmatrix} \quad (4.62)$$

bestimmt werden, wobei für die Länge des Gradienten folgt

$$\|\nabla g_k(\mathbf{x})\|^2 = \sum_{j=1}^n \left( \frac{\alpha_k}{\sigma_k^2} \exp(\cdot) (x_j - c_{kj}) \right)^2. \quad (4.63)$$

Für das Maximum von (4.63) bzgl. eines Einganges  $x_i$  gilt dann

$$\frac{\partial \|\nabla g_k(\mathbf{x})\|^2}{\partial x_i} = 0 \quad (4.64)$$

$$\Rightarrow x_i = c_{ki} \pm \sigma_k \quad (4.65)$$

$$x_j = c_{kj} \quad \forall j = 1 \dots n, j \neq i. \quad (4.66)$$

Für (4.65) und (4.66) kann nachgewiesen werden, dass an diesen Punkten tatsächlich ein Maximum der Ableitung existiert. Damit besitzt die Länge des Gradienten mehrere Maximastellen, wobei diese Stellen dann als die Empfindlichkeit eines Neurons definiert werden können:

**Definition 7** (Empfindlichkeit eines Neurons). *Es sei  $g_k(\mathbf{x})$  die Aktivierungsfunktion eines Neurons  $k$  aus einem RBF-Netz, und  $\alpha_k$ ,  $\sigma_k$  und  $\mathbf{c}_k$  seien die freien Parameter dieser Gaußfunktion. Dann bezeichnet*

$$\max \{ \|\nabla g_k(\mathbf{x})\|^2 \} = \frac{\alpha_k^2}{\sigma_k^2} e^{-1} \quad (4.67)$$

die Empfindlichkeit des Neurons  $k$ .

Für die Empfindlichkeit<sup>5</sup> eines Neurons wird somit das Maximum der Länge des Gradienten verwendet. Mit (4.67) können die empfindlichen Neuronen bzw. deren Empfindlichkeit gegenüber Rauschen durch den Quotienten aus Ausgangsgewicht und Varianz einer Basisfunktion bestimmt werden. Dadurch können die empfindlichen Neuronen eines Netzes durch

$$\max_{i=1\dots m} \left\{ \frac{|\alpha_k|}{\sigma_k} \right\} \quad (4.68)$$

berechnet werden, wobei  $\sigma_k \neq 0$  gelten muss, also

$$\sigma_k \in \mathbb{R} \setminus \{0\},$$

was durch die Definition der RBF (2.28) sichergestellt ist.

Wie man in Abbildung 4.7 erkennen kann, ist der Quotient aus (4.68) für beide Gaußfunktionen in Abbildung 4.7(a) identisch, während die Darstellung 4.7(b) unterschiedliche Empfindlichkeit der Gaußfunktionen gegenüber Rauschen zeigt, da beide unterschiedliche Quotienten des Ausgangsgewichtes und ihrer Varianz besitzen.

Mit Hilfe dieser einfachen Berechnungsvorschrift lässt sich die Sensibilität der einzelnen Gaußfunktionen unabhängig vom verwendeten Trainingsverfahren berechnen. Aufgrund der Einfachheit ist eine Bestimmung während der Laufzeit ohne hohe Anforderungen an die Rechenleistung möglich.

Obwohl die Definition der Empfindlichkeit von Neuronen bzgl. Rauschens in den Eingängen aufgestellt wurde, kann das Maß aus (4.68) ebenfalls für Rauschen in den Gewichten verwendet werden, um hier empfindliche Neuronen zu bestimmen, da, wie schon in Abschnitt 3.1 und 4.2.1, die Gewichte und Varianzen den größten Einfluss haben.

<sup>5</sup> Synonym zur Empfindlichkeit wird im Weiteren auch der Begriff *Sensitivität* verwendet.

Das Ergebnis aus (4.68) ist konsistent zu den simulativen Ergebnissen aus [PC01]. Dort wurde beobachtet, dass die Qualität der Approximation und die Empfindlichkeit proportional zu den absoluten Werten der Ausgangsgewichte verlaufen. Zusätzlich ist das Ergebnis aus (4.68) vergleichbar mit den *weight-based feature saliency measures*, bei denen die zentrale Idee ist, große Ausgangsgewichte bei sehr wichtigen und häufig auftretenden Merkmalen zu verwenden [BAG00, VB02].

*Beispiel.* Bei der Identifikation der empfindlichen Basisfunktionen müssen für das RBF-Netz mit 10 Gaußfunktionen zur Approximation der si-Funktion nur die Gewichte und Varianzen betrachtet werden. Somit müssen nicht insgesamt alle Parameter des Netzes zur Bestimmung der größten Rauschquelle ausgewertet werden, was hier einer Anzahl von 31 Rauschquellen bedeuten würde. Mit höherer Dimension fällt der Unterschied entsprechend größer aus. Eine Betrachtung des RBF-Netzes ergibt, dass die empfindlichsten Neuronen auch den größten Auswirkungen bei einem Ausfall entsprechen. Dies liegt an der besonderen Form des neuronalen Netzes, da hier für alle Gaußfunktionen die gleichen Varianzen verwendet werden. So ergibt sich sowohl für die Bedeutung als auch für die Empfindlichkeit eine Proportionalität zum Ausgangsgewicht der Gaußfunktion.

### 4.3 Robustes und fehlertolerantes Lernen

Das Trainieren der neuronalen Netze bestimmt die freien Parameter der Netzwerke, wie z.B. die Ausgangsgewichte, die Varianzen der Gaußfunktionen oder die Schwellenwerte und Steigungen der Sigmoiden sowie die Gewichtungen der Verbindungen der Neuronen untereinander. Welche Parameter bestimmt werden müssen, hängt vom Typ des neuronalen Netzes ab. Dabei soll das neuronale Netz die gestellte Aufgabe<sup>6</sup> möglichst gut erfüllen und durch diesen Lernprozess seine Eigenschaften verbessern [Hay99].

Die Auswirkungen und Verbesserungen des Lernens werden durch eine Kostenfunktion erfasst, wobei das Ziel des Lernprozesses die Minimierung dieser Funktion darstellt. Oft bestimmt diese Kostenfunktion die Diskrepanz zwischen der gewünschten Ausgabe des neuronalen Netzes und der Erzeugten, was meist durch den mittleren quadratischen Fehler erfasst wird [Hay99, Seite 89 ff.]. Es existiert in der Literatur eine Vielzahl von verschiedenen Lernverfahren, die unterschiedliche Methoden verwendet und für die verschiedenen Netze einsetzbar sind, so z.B. beim

**MLP** Backpropagation als populärer Vertreter des Lernens [RHW86a] sowie diverse andere Möglichkeiten<sup>7</sup> [CP03, RB93, Bis93]

<sup>6</sup> Als Aufgabe kann z.B. die Approximation von Funktionen, die Klassifikation von Daten oder die Zeitreihenvorhersage gewünscht sein.

<sup>7</sup> Hier sei nur beispielhaft auf einige mögliche Verfahren verwiesen.

**RBF** Initialisieren der Zentren und Bestimmen der Varianzen sowie anschließendes Lösen des LGS mit Hilfe des *Orthogonal least squares* Verfahren [CCG91] oder der *Orthogonal triangular* Zerlegung [GLL96] sowie diverse andere Methoden [PC01, MH05, HSS05, WTK05]

Als Kostenfunktion wird i.A. der mittlere quadratische Fehler zwischen den geforderten Ausgangswerten des Netzes und der durch die Trainingsdaten Erzeugten berechnet. So wird der Lernprozess mathematisch durch ein Optimierungsproblem beschrieben [HUL93], welches hinsichtlich der freien Parameter minimiert werden soll

$$\min F(\mathbf{w}) = \min \frac{1}{M} \sum_{i=1}^M (f_m(\mathbf{x}^i, \mathbf{w}) - y^i)^2 \quad \mathbf{w} \in \mathbb{R}^{m(n+2)}, \quad (4.69)$$

wobei  $\mathbf{w}$  den Parametervektor, der alle freien Parameter des neuronalen Netzes enthält, und  $M$  die Anzahl der Testdaten bezeichnet. Für ein RBF-Netz nach (2.28) sind hier als freie Parameter die Ausgangsgewichte, die Zentren und die Varianzen der Gaußfunktionen möglich. Ebenfalls für das LCNN oder ein anderes MLP kann (4.69) zur Bestimmung der freien Parameter verwendet werden. Die Art der Parameter ist vom Netztyp abhängig.

### 4.3.1 Robustheit als Zielfunktion

Die Kostenfunktion des Lernverfahrens erfasst zumeist nur eine bestimmte Eigenschaft des neuronalen Netzes, für welche dieses optimiert werden soll. Daneben kann es wünschenswert sein, noch weitere Eigenschaften des neuronalen Netzes zu verbessern. Als weitere Eigenschaften neben der Approximationsgüte aus (4.69) können eine hohe Robustheit des Netzes oder eine gute Fehlertoleranz gewünscht sein. Da alle diese Eigenschaften gleichberechtigte Ziele darstellen, entsteht in Analogie zu dem Optimierungsproblem aus (4.69) ein Mehrzieloptimierungsproblem (MOP) der folgenden Form [SNT85, CS04b, ER06b].

$$\text{minimiere} \quad \mathbf{y} = \mathbf{F}(\mathbf{w}) = \begin{pmatrix} F_1(\mathbf{w}) \\ F_2(\mathbf{w}) \\ \vdots \\ F_\Omega(\mathbf{w}) \end{pmatrix} \quad (4.70)$$

$$\text{mit} \quad \mathbf{g}(\mathbf{w}) = (g_1(\mathbf{w}), g_2(\mathbf{w}), \dots, g_k(\mathbf{w}))^T \leq \mathbf{0}$$

$$\mathbf{h}(\mathbf{w}) = (h_1(\mathbf{w}), h_2(\mathbf{w}), \dots, h_p(\mathbf{w}))^T = \mathbf{0},$$

wobei  $\Omega$  die Anzahl der Zielfunktionen darstellt und  $\mathbf{g}(\mathbf{w})$  bzw.  $\mathbf{h}(\mathbf{w})$  die Randbedingungen in Form von Ungleichungen und Gleichungen beschreiben. Der Vektor  $\mathbf{w}$  enthält alle freien Parameter des Netzes, also z.B. die Zentren, die Varianzen und die Ausgangsgewichte im Falle eines RBF-Netzes. Die Randbedingungen beschreiben

dabei die mögliche Lösung im Urbildraum und schränken die Menge der möglichen Lösungen für  $\mathbf{w}$  ein. Die Zielfunktionen besitzen häufig zueinander gegenläufige Trends bzw. Auswirkungen.

Da in (4.70) keine der Zielfunktionen ein primäres Ziel darstellt, existieren mehrere mögliche Lösungen des Mehrzieloptimierungsproblems, die zusammen eine Pareto-front im Urbildraum beschreiben. Für weitere Informationen über die Definition der dominanten Punkte, pareto-optimaler Mengen und Punkte sei auf die Literatur verwiesen [SNT85, CS04b].

Zunächst soll die Robustheit des neuronalen Netzes als zusätzliche Zielfunktion in den Lernprozess einfließen. Die Untersuchung wird beispielhaft für ein RBF-Netz durchgeführt, wobei sich für das LCNN und die Basisfunktionsnetze mit Produktstabilisatoren eine analoge Vorgehensweise ergibt. Für die Robustheit werden die in Kapitel 3.1 und 4.2.1 eingeführten Maße verwendet. So ergibt sich ein MOP, was durch

$$\text{minimiere } \mathbf{y} = \mathbf{F}(\mathbf{w}) = (F_1(\mathbf{w}), F_2(\mathbf{w}))^T \quad (4.71)$$

$$\text{mit } \mathbf{g}(\mathbf{w}) = (g_1(\mathbf{w}), g_2(\mathbf{w}), \dots, g_k(\mathbf{w}))^T \leq \mathbf{0} \quad (4.72)$$

$$\mathbf{h}(\mathbf{w}) = (h_1(\mathbf{w}), h_2(\mathbf{w}), \dots, h_p(\mathbf{w}))^T = \mathbf{0} \quad (4.73)$$

$$\text{wobei } F_1(\mathbf{w}) = \frac{1}{M} \sum_{i=1}^M \left( y^i - \sum_{\nu=1}^m \alpha_\nu \exp \left( \frac{-\|\mathbf{x}^i - \mathbf{c}_\nu\|^2}{2\sigma_\nu^2} \right) \right)^2$$

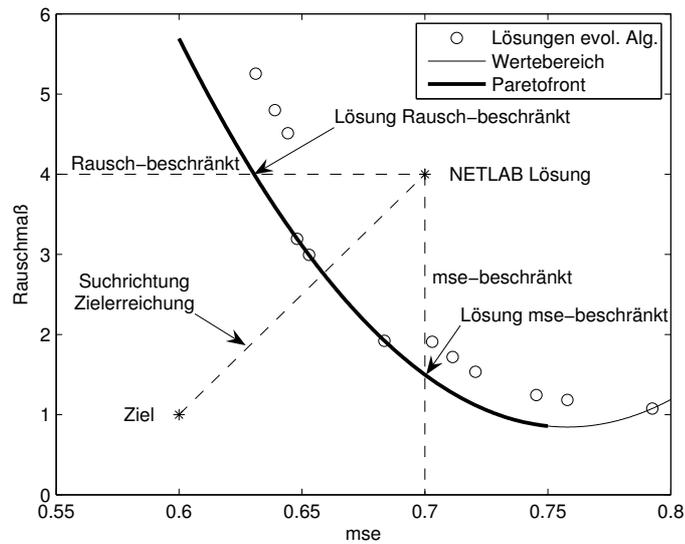
$$F_2(\mathbf{w}) = n^3 \left( \sum_{\nu=1}^m \left| \frac{\alpha_\nu}{\sigma_\nu} \right| \right)^2 \quad (4.74)$$

$$\text{oder } F_2(\mathbf{w}) = \frac{1}{M} \sum_{i=1}^M \sum_{j=1}^n \left( \sum_{\nu=1}^m -\alpha_\nu \frac{(x_j^i - c_{\nu j})}{\sigma_\nu^2} e^{-\frac{\|\mathbf{x}^i - \mathbf{c}_\nu\|^2}{2\sigma_\nu^2}} \right)^2 \quad (4.75)$$

beschrieben werden kann. Hier stellt  $M$  die Anzahl der Trainingsvektoren  $(\mathbf{x}^i, y^i)$  mit  $i = 1 \dots M$  dar, mit denen das neuronale Netz trainiert wird. Bei diesen bezeichnet  $x_j^i$  den  $j$ -ten Eintrag bzw. Eingang des aktuellen ( $i$ ) Trainingsvektors. Der Spaltenvektor  $\mathbf{w}$  enthält zuerst die  $m$  Zentren der Gaußfunktionen, dann die Varianzen und anschließend die Ausgangsgewichte des Netzes.

Die Randbedingungen aus (4.72) und (4.73) formen im Definitionsbereich die möglichen Lösungen des MOPs. Diese können z.B. die oberen und unteren Grenzen der Parameter bezeichnen, wie  $\sigma_i^2 > 0 \forall i = 1 \dots m$ . Zur Lösung des MOPs aus (4.71) können verschiedene mathematische Methoden verwendet werden. Hier soll kurz auf mögliche Vorgehensweisen eingegangen werden. Für eine genaue Übersicht sei auf [CS04b] verwiesen.

Das MOP kann in ein skalares Problem transformiert werden, wobei jeder Zielfunktion ein Gewicht zugeordnet wird und anschließend diese Funktionen summiert werden. Das skalare Problem kann dann mit klassischen Verfahren, wie z.B. durch



**Abbildung 4.8:** Verfahren zur Lösung des Mehrzieloptimierungsproblems

Gradientenverfahren [HUL93], gelöst werden. Die korrekte Bestimmung der Gewichte und die Notwendigkeit eines konvexen Wertebereiches sind die Nachteile dieses Verfahrens. Neben der Summenbildung kann ebenfalls das Produkt gebildet werden, was als *Keeney-Raiffa* Methode bekannt ist.

Die Anwendung einer Summenbildung für das MOP in (4.71) ähnelt der Methode des Trainings mittels Einfügen von Rauschen in die Lerndaten. Das zusätzliche Rauschen verbessert die Generalisierungseigenschaften des Netzes und bewirkt für die Fehlerfunktion das Einfügen eines zusätzlichen Regularisierungsparameters, der durch die Varianz des Rauschens bestimmt wird [Bis95b, Seite 346]. Dieser zusätzliche Parameter in der Zielfunktion ist konsistent mit den Ergebnissen aus Abschnitt 4.2.1.

Weiter kann die Methode der Zielerreichung (*goal attainment*) zum Finden einer Lösung angewendet werden. Bei diesem Vorgehen wird ein Startvektor mit gültigen<sup>8</sup> Werten der Zielfunktion gewählt und eine Suchrichtung im Bildraum bestimmt, wobei ein skalarer Koeffizient  $\lambda$  minimiert werden soll, der den Abstand zur Startlösung darstellt (vgl. Abbildung 4.8). Anschaulich wird hier der pareto-optimale Punkt bestimmt, der auf der Geraden liegt, die durch den Startvektor und die Suchrichtung beschrieben wird. Der Wertebereich muss hierfür nicht notwendigerweise konvex sein.

Daneben können  $M - 1$  Zielfunktionen in Randbedingungen transformiert werden. Diese Funktionen können als zusätzliche Randbedingungen in Gleichungs- oder in Ungleichungsform eingesetzt werden. Das Optimierungsproblem besteht dann nur noch aus einer einzigen Zielfunktion mit erweiterten Randbedingungen, wie in Ab-

<sup>8</sup> Für den Startwert sollte eine „gute“ Lösung ausgewählt werden.

bildung 4.8 dargestellt, wobei zum Finden des Minimums Verfahren für den eindimensionalen Fall eingesetzt werden können.

Die bisher beschriebenen Methoden bestimmen nur einen Punkt auf der pareto-optimalen Front. Um diese komplett zu bestimmen, müssen spezielle Lösungsverfahren angewendet werden. Eine stochastische Suche kann verwendet werden, um die möglichen Minima des MOPs zu finden. Hier werden meist evolutionäre Algorithmen eingesetzt. Für die genaue Vorgehensweise und die Eigenschaften dieser Verfahren sei auf die Literatur [Mos04, CFPPF94] verwiesen. Weiter können auch mathematisch exakte Verfahren [DSH05] angewendet werden, die zur Lösung solcher MOPs entworfen wurden.

Wenn es sich bei dem MOP um ein konvexes Problem handelt, lässt sich die Lösung sehr einfach bestimmen, und man benötigt keine evolutionären Algorithmen zur Bestimmung der Lösung. Wenn alle Zielfunktionen und die mögliche Menge im Definitionsbereich konvex sind, so existiert ein konvexes MOP [CS04b, HUL93]. Das MOP aus (4.71) besitzt zwar eine konvexe Definitionsmenge, jedoch erfüllt die Zielfunktion  $F_1(\mathbf{w})$  nicht die Eigenschaft einer konvexen Funktion. Somit ist das Verfahren der gewichteten Summe zur Lösung dieses Problems ungeeignet. Der Nachweis, dass es sich bei der Zielfunktion  $F_1(\mathbf{w})$  um eine nicht konvexe Funktion handelt, kann durch ein Gegenbeispiel geführt werden. Dabei folgt für einen einzelnen Testdatensatz  $x^1 = 0, y^1 = 1$  und für festgelegte Parameter des Netzes  $m = 1, \alpha_1 = -1, c_1 = 1$  bei der Zielfunktion  $F_1(\sigma)$

$$F_1(\sigma) = \left( 1 + e^{-\frac{(0-1)^2}{2\sigma^2}} \right)^2. \quad (4.76)$$

Dies ist aber keine konvexe Funktion im Intervall  $(0; 10]$ , was man leicht am Funktionsgraphen verifizieren kann. Damit handelt es sich nicht um ein konvexes MOP.

Zur Lösung des MOPs aus (4.71) werden die anderen vorgestellten Methoden, Zielerreichung und Transformation der Zielfunktionen, verwendet. Für die Methode der Zielerreichung werden dabei ein Startvektor und eine Suchrichtung benötigt. Um mit einer „guten“ Lösung zu starten, wird diese mit Hilfe der NETLAB-Werkzeuge für MATLAB [Nab02] erzeugt. Für die Suchrichtung wird der Richtungsvektor verwendet, der durch den Ursprung und die Startlösung vorgegeben wird. So wird in beiden Zielen die gleiche relative Über- und Unterbestimmung gewährleistet.

Daneben wird die Zielfunktion  $F_2(\mathbf{w})$  in einem weiteren Verfahren als zusätzliche Randbedingung in Ungleichungsform eingesetzt, und als obere Schranke wird das Rauschmaß der NETLAB-Lösung nach (4.74) oder (4.75) verwendet. Dieses Optimierungsproblem und dessen Lösung werden im Folgenden als *Rausch*-beschränktes Problem bezeichnet. Weiter kann auch die Zielfunktion  $F_1(\mathbf{w})$  als Randbedingung verwendet werden. Dabei wird auch hier die durch NETLAB ermittelte Lösung als obere Schranke verwendet. So entsteht ein *mse*-beschränktes Problem, wobei der

mittlere quadratische Fehler des Netzes kleiner als derjenige der NETLAB-Lösung sein soll.

Bei allen Verfahren wird zuerst das neuronale Netz mittels NETLAB trainiert<sup>9</sup>. In wieweit diese Lösung sich verbessern lässt, hängt von der Art der Lösung ab. Handelt es sich bei dieser um einen pareto-optimalen Punkt im Bildraum, so kann per Definition kein Punkt gefunden werden, der in einer der beiden Zielfunktionen besser ist. Bei vielen Trainingsverfahren wird jedoch die Robustheit des neuronalen Netzes nicht näher betrachtet, sondern lediglich die Approximations- oder Klassifizierungseigenschaften optimiert. So sollte es für eine Vielzahl von Lernverfahren möglich sein, die gefundene Lösung hinsichtlich ihrer Empfindlichkeit gegenüber Rauschen zu verbessern.

Das Verfahren kann als eigenständiges Training eingesetzt werden, wenn (4.71) durch evolutionäre Algorithmen oder andere mathematische Verfahren gelöst wird. Welcher Punkt der Paretofront mit entsprechenden Parametern dann für das Netz gewählt wird, ist von der Anwendung oder dem Entwickler abhängig.

Zur simulativen Verdeutlichung der verschiedenen Verfahren und der Wirksamkeit der Robustheit des neuronalen Netzes als zusätzliche Zielfunktion wird im Folgenden das MOP für das trainierte RBF-Netz gelöst. Zur Approximation werden drei Testfunktionen aus [GRO<sup>+</sup>03] verwendet

$$f_1(x) = 3x(x-1)(x-1.9)(x+0.7)(x+0.8) \quad x \in [-2,1; 2,1] \quad (4.77)$$

$$f_2(x,y) = 42.659(0.1 + x(0.05 + x^4 - 10x^2y^2 + 5y^4)) \quad x,y \in [-0,5; 0,5] \quad (4.78)$$

$$f_3(x,y) = 1.9(1.35 + e^x \sin(13(x-0.6)^2) e^y \sin(7y)) \quad x,y \in [0; 1]. \quad (4.79)$$

Zur Bestimmung einer Anfangslösung wird die Lösung von NETLAB verwendet. Für das Trainieren des Netzes wird eine Menge mit 1000 Vektoren sowie eine Testmenge mit 5000 Vektoren für den Fall einer zweidimensionalen Testfunktion innerhalb des Definitionsbereiches erstellt. Für den eindimensionalen Fall werden eine Trainingsmenge mit 100 Punkten und eine Menge mit 1000 Punkten zum Testen erstellt. Alle Testvektoren werden aus dem Definitionsbereich der jeweiligen Funktion auf Basis einer Gleichverteilung entnommen. Die Anzahl von Gaußfunktionen wird exemplarisch aus den Ergebnissen aus [GRO<sup>+</sup>03] gewählt. Zur Ermittlung der Lösung des Verfahrens mit zusätzlichen Randbedingungen werden Standardverfahren der Mathematik genutzt, so z.B. eine sequenzielle, quadratische Methode, bei der die Hessematrix in jedem Iterationsschritt geschätzt wird [The92].

Tabelle 4.7 zeigt die Ergebnisse der weiteren Verbesserung einer Lösung des ursprünglichen Trainingsverfahrens, wobei als zusätzliche Zielfunktion oder Randbedingung die Empfindlichkeit der Eingänge des Netzes nach (4.75) verwendet wird.

<sup>9</sup> Neben dem Algorithmus aus NETLAB kann jedes beliebige Trainingsverfahren verwendet werden.

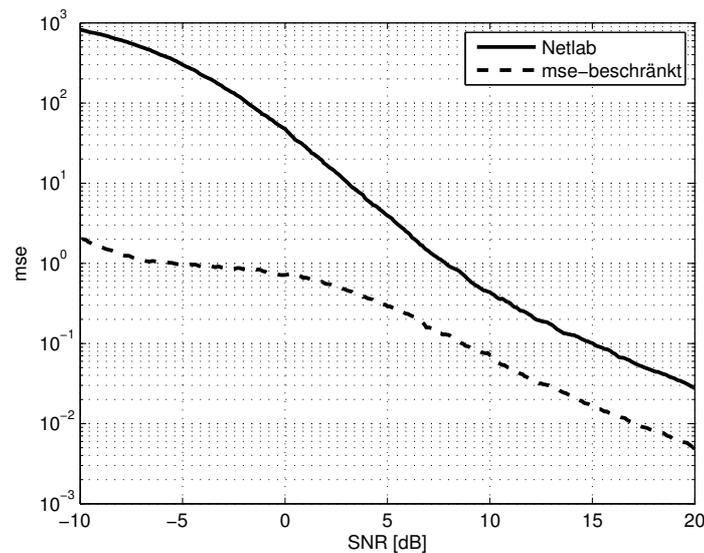
**Tabelle 4.7:** Lösungen des Mehrzieloptimierungsproblems mit Hilfe verschiedener Verfahren. Als zweite Zielfunktion wird die Empfindlichkeit der Eingänge nach (4.75) verwendet.

Funktion	m	Verfahren	mse Training	mse Test	Rauschen
$f_1$	4	NETLAB	37,0388	43,5619	$3,37 \cdot 10^4$
		Zielerreichung	36,2133	43,1464	$3,29 \cdot 10^4$
		Rausch-beschränkt	36,4030	42,9766	$3,37 \cdot 10^4$
		mse-beschränkt	37,0388	44,6148	$3,22 \cdot 10^4$
$f_2$	8	NETLAB	0,6161	0,5829	981,39
		Zielerreichung	0,5403	0,6083	640,26
		Rausch-beschränkt	0,2729	0,3316	944,57
		mse-beschränkt	0,6161	0,5829	583,26
$f_3$	6	NETLAB	0,6312	0,6786	5,2542
		Zielerreichung	0,4605	0,5155	3,8333
		Rausch-beschränkt	0,4536	0,5041	5,2542
		mse-beschränkt	0,6312	0,7095	1,2279

Die Werte für das Rauschmaß sind in einer Pseudoeinheit angegeben, da die Empfindlichkeit und das Rauschen im Ausgang von der Rauschleistung im Eingang abhängen. Der sich einstellende mittlere quadratische Fehler ist mit den Ergebnissen aus [GRO<sup>+</sup>03] vergleichbar, da dort ein normiertes Fehlermaß benutzt wird.

Die Tabelle 4.7 zeigt, dass sich die ursprünglichen Lösungen von NETLAB bedeutend sowohl in den Approximationsfähigkeiten als auch in den Rauscheigenschaften verbessern lassen. So zeigt sich für die Testfunktionen  $f_2$  und  $f_3$  in beiden Zielfunktionen eine deutliche Verbesserung gegenüber der NETLAB-Lösung. Dies ist möglich, wenn es sich bei der Startlösung nicht um eine pareto-optimale Lösung handelt. Im Falle der Testfunktion  $f_1$  zeigt sich, dass die gefundenen Lösungen nur unwesentlich von der Ursprünglichen abweichen. Hier ist schon ein Punkt nahe der Paretofront gefunden worden, so dass keine weiteren Verbesserungen möglich sind. Für den Fall, dass die beiden Lösungen für das Optimierungsproblem keine pareto-optimalen Punkte darstellen, kann die Empfindlichkeit des Netzes gegenüber Rauschen um den Faktor zwei für die Testfunktion  $f_2$  und um den Faktor vier für die Testfunktion  $f_3$  verbessert werden. Mit der Methode der Zielerreichung lassen sich beide Zielfunktionen verbessern.

Welche der Methoden den größten Nutzen für den Entwurf bringt, hängt stark von der Gestalt der Wertemenge ab, die durch die beiden Zielfunktionen beschrieben wird. Für konvexe Mengen lässt sich mit dem Verfahren der transformierten Zielfunktionen das globale Minimum bestimmen. Die Methode der Zielerreichungen liefert auch für nicht konvexe Mengen einen pareto-optimalen Punkt. Allerdings lassen sich auch hier Formen im Bildraum konstruieren, bei denen die Güte der Lösung stark von der Suchrichtung und dem Startwert abhängt [CS04b, Seite 65].



**Abbildung 4.9:** Verlauf des mittleren quadratischen Fehlers für zwei neuronale Netze in Abhängigkeit des Rauschens am Eingang für die Testfunktion  $f_3$

Abbildung 4.9 zeigt den mittleren quadratischen Fehler aufgrund Rauschens in den Eingängen für die Testfunktion  $f_3$ . Dabei werden die Eingänge mit unterschiedlichen Rauschleistungen belegt, die durch ihr Signal-Rausch-Verhältnis (SNR) bestimmt werden. Hier wird das Verhalten zweier unterschiedlich trainierter Netze dargestellt, zum einen ein Netz mit den Parametern der NETLAB-Lösung und zum anderen von der Lösung des MOPs. Es zeigt sich, dass sich bei gleicher Topologie des Netzes mit zwei unterschiedlichen Parametersätzen verschiedene Antworten auf das Rauschen ergeben können. Die NETLAB-Lösungen zeigen bei Rauschen in den Eingängen einen um zwei Größenordnungen höheren, mittleren quadratischen Fehler. Beide Netze besitzen für die Test- und die Trainingsmenge einen vergleichbaren Approximationsfehler.

Tabelle 4.8 zeigt die Resultate für ein Mehrzieloptimierungsproblem mit dem mittleren quadratischen Fehler aufgrund rauschender Eingänge als zusätzliche Zielfunktion (4.74). Für die Testfunktion  $f_2$  stellt die NETLAB-Lösung einen Punkt nahe der Paretofront dar. Es lassen sich mit einem zusätzlichen Optimierungsschritt keine Verbesserungen in einer der beiden Zielfunktionen erreichen. Dass in diesem Fall bei der Testfunktion  $f_2$  ein pareto-optimaler Punkt mittels NETLAB gefunden wird, ist zufällig bedingt und nicht von der Wahl der zweiten Zielfunktion abhängig. Der NETLAB-Algorithmus wählt bei jeder Iteration unterschiedliche Zentren und Varianzen, wobei die Ausgangsgewichte dann mittels der Pseudoinversen bestimmt werden. So ergeben sich verschiedene Lösungen im Bildraum.

Die ursprünglichen Lösungen für die Testfunktionen  $f_1$  und  $f_3$  können dagegen in beiden Zielfunktionen verbessert werden, so dass als Schlussfolgerung die initiale

**Tabelle 4.8:** Lösungen des Mehrzieloptimierungsproblems mit Hilfe verschiedener Verfahren. Als zweite Zielfunktion wird der mittlere quadratische Fehler aufgrund verrauschter Eingänge nach (4.74) verwendet.

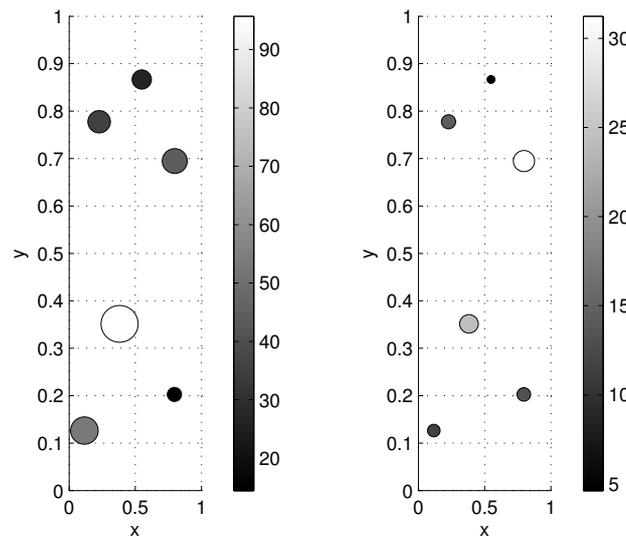
Funktion	m	Verfahren	mse Training	mse Test	Rauschen
$f_1$	4	NETLAB	38,4554	55,5408	$6,04 \cdot 10^9$
		Zielerreichung	28,2350	41,0862	$4,44 \cdot 10^9$
		Rausch-beschränkt	30,3200	43,7137	$5,01 \cdot 10^9$
		mse-beschränkt	38,4535	56,4501	$0,08 \cdot 10^9$
$f_2$	8	NETLAB	0,6047	0,6485	$1,42 \cdot 10^9$
		Zielerreichung	0,6047	0,6485	$1,42 \cdot 10^9$
		Rausch-beschränkt	0,6047	0,6485	$1,42 \cdot 10^9$
		mse-beschränkt	0,6047	0,6485	$1,42 \cdot 10^9$
$f_3$	6	NETLAB	0,6426	0,6336	$7,13 \cdot 10^5$
		Zielerreichung	0,1761	0,1708	$1,94 \cdot 10^5$
		Rausch-beschränkt	0,6473	0,6422	$7,13 \cdot 10^5$
		mse-beschränkt	0,6473	0,6378	535,48

Lösung keinen pareto-optimalen Punkt darstellt. Die Methode der Zielerreichung kann in den beiden Zielfunktionen eine Verbesserung für die Funktionen  $f_1$  und  $f_3$  erreichen. Auch die beiden anderen Verfahren minimieren die jeweilige Zielfunktion. Bei der Testfunktion  $f_3$  zeigt sich bei dem rausch-beschränkten Problem, dass die Methode in lokalen Minima festsitzen kann. Wie schon gezeigt, stellt die Zielfunktion  $F_1(\mathbf{w})$  keine konvexe Funktion dar, und so muss die Methode nicht notwendigerweise das globale Minimum finden, sondern kann ein lokales Minimum erreichen.

Die Effizienz der Lösung des MOPs hängt stark von der anfänglichen Lösung ab. Stellt diese einen pareto-optimalen Punkt dar, so kann per Definition keine der beiden Zielfunktionen verbessert werden, und die vorgestellten Methoden finden keine besseren Parametersätze. Andernfalls lassen sich beide Zielfunktionen durch die Methode der Zielerreichung verbessern. Die beiden anderen Methoden hängen von der Form der möglichen Lösung im Bildraum ab. Da es sich aber nicht um konvexe Funktionen handelt, besteht die Gefahr in einem lokalen Minimum hängen zu bleiben.

#### 4.3.1.1 Auswirkungen auf die Parameter

Durch eine zweite Zielfunktion beim Trainieren des neuronalen Netzes kann das Verhalten des Netzes gegenüber Rauschen verbessert werden. Dabei sind die beiden Zielfunktionen zueinander gegenläufig, da zur Minimierung des Rauscheinflusses die Steilheit der Neuronen verringert werden muss, wobei für gute Approximationseigenschaften das Netz dem Verlauf der Funktion auch bei großen Steigungen folgen muss.



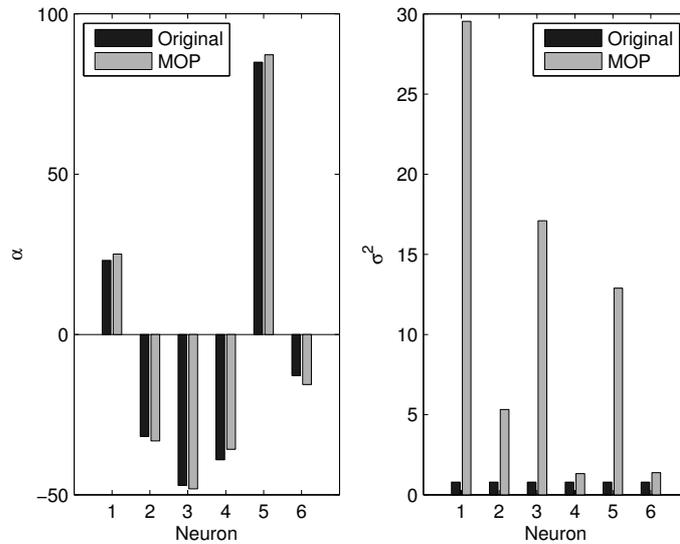
**Abbildung 4.10:** Verteilung der Zentren und der Quotient  $\alpha/\sigma$  für das ursprüngliche RBF-Netz (links) und nach dem MOP (rechts)

Hier sollen die Auswirkungen auf die Parameter beispielhaft untersucht werden. Da in dieser Arbeit besonders die Robustheit der Netze im Mittelpunkt steht, wird bei der Analyse auf die mse-beschränkte Lösung eingegangen. Bei diesem Problemfall können tendenzielle Aussagen über den Einfluss auf die Parameter des Netzes getroffen werden, während zur Verbesserung der Approximationseigenschaften ein komplexer Zusammenhang besteht. Zum einen handelt es sich um ein nichtkonvexes Problem, so dass das Finden des globalen Minimums vom Optimierungsverfahren abhängig ist und die Gefahr besteht, in lokalen Minima hängen zu bleiben. Zum anderen sind die Größe des Netzes und damit ebenfalls dessen Parameter von der zu approximierenden Funktion abhängig. So können lediglich für eine Beispielfunktion Aussagen getroffen werden. Da beide Zielfunktionen zusammen betrachtet werden müssen, können somit keine allgemeingültigen Aussagen getroffen werden.

Hinsichtlich des Einflusses der zweiten Zielfunktion, dem Einfluss des Rauschens, können allerdings tendenzielle Aussagen getroffen werden. Aufgrund der Abhängigkeit in (4.74) bzw. (4.75) von dem Quotienten der Ausgangsgewichte und der Breite der Gaußfunktion wird bei einer Verwendung der Approximationseigenschaften als zusätzliche Randbedingung versucht werden, jene zu verringern und die Zentren entsprechend zu verschieben.

Zur beispielhaften Analyse wurde aus Tabelle 4.8 die mse-beschränkte Lösung der Funktion  $f_3$  verwendet. Dabei werden im Folgenden besonders die Lagen der Zentren, die Höhe und die Breite der Gaußfunktionen betrachtet.

Abbildung 4.10 zeigt die Verteilung der Zentren der Basisfunktionen in der  $xy$ -Ebene. Der Flächeninhalt und die Farbe der Kreise sind linear von dem Quotienten



**Abbildung 4.11:** Höhe und Breite der Gaußfunktionen bei Robustheit

$\alpha/\sigma$  abhängig, so dass ein robusteres Netz nach (4.74) kleinere Kreise besitzt. Die Fläche erlaubt dabei einen Vergleich der beiden Netze untereinander, da die gleiche Skalierung für beide Netze gewählt wurde. Die Farbe dagegen zeigt die relative Größe des Quotienten für jedes Netz separat, so dass hierdurch die Basisfunktion mit der höchsten Empfindlichkeit für jedes Netz bestimmt werden kann.

Durch die zweite Zielfunktion wird der Quotient aus dem Ausgangsgewicht und der Varianz für alle Basisfunktionen reduziert, so dass sich eine robustere Approximation durch das Netz ergibt. Die Lagen der Gaußfunktionen bleiben dabei unverändert, um weiterhin die gleichen Approximationseigenschaften zu erhalten. Die Lagen der Zentren werden für diesen Fall durch die angewendete Methode (Erwartungsmaximierung eines gaußschen Mischmodells) von NETLAB optimal gewählt. Das robustere Netz wird lediglich durch eine Veränderung der Höhe und der Breite der Gaußfunktionen erreicht. Um die Anforderungen an die Optimierung zu reduzieren, können somit die Zentren aus diesem Prozess herausgenommen werden, was die Komplexität des Verfahrens erheblich reduziert. Das Optimierungsproblem erfährt eine Reduktion aus dem  $\mathbb{R}^{(n+2)\cdot m}$  in den  $\mathbb{R}^{2m}$  und kann so i.A. effizienter hinsichtlich seiner Konvergenz und der Art des Minimums arbeiten.

Abbildung 4.11 veranschaulicht die Gewichte und die Breite der Gaußfunktionen für das ursprüngliche Netz und nach dem MOP. Hier zeigt sich, dass in diesem Beispiel die Gewichte nur minimal gegenüber dem ursprünglichen Netz verändert werden. Hingegen werden größere Breiten bei der Approximation verwendet, was zu einem robusteren Netz führt. Weiterhin führen größere Varianzen zu einer besseren Konditionszahl der einzelnen Gaußfunktion, was sich positiv auf Rundungsfehler auswirkt (vgl. Abschnitt 5.1).

Die Berücksichtigung einer zweiten Zielfunktion führt sogar zu größeren Ausgangsgewichten, was zunächst für die Robustheit des Netzes nicht förderlich ist. Durch die höheren Breiten kann diese Verschlechterung in der zusätzlichen Zielfunktion kompensiert und verbessert werden. Allerdings wird durch die größeren Gewichte und die höhere Varianz die Fehlertoleranz des Netzes direkt beeinflusst, und diese verschlechtert sich aufgrund der größeren Breite der Basisfunktionen (siehe Abschnitt 3.2). Damit stellen Robustheit und Fehlertoleranz in einem bestimmten Maße gegenläufige Ziele dar (vgl. (3.16) bzw. (4.48) und (4.8)). Eine gemeinsame Verbesserung kann nur durch kleinere Ausgangsgewichte erreicht werden, da bei der Veränderung der Varianzen jeweils eine Eigenschaft des Netzes, Fehlertoleranz oder Robustheit, herabgesetzt wird (siehe Abschnitt 4.1).

### 4.3.2 Fehlertoleranz als Zielfunktion

Neben der Robustheit eines neuronalen Netzes ist auch dessen Fehlertoleranz von Bedeutung. Um besonders wichtige Neuronen des Netzes zu identifizieren, wurde in Abschnitt 4.1 ein Maß für die Bedeutung der Neuronen bei einem RBF-Netz und einem LCNN eingeführt. Dieser Bewertungsmaßstab kann ebenso wie die Robustheit im vorangegangenen Abschnitt als zusätzliche Zielfunktion beim Trainieren eingesetzt werden.

So erhält man analog zu (4.71) ein Mehrzieloptimierungsproblem mit der Fehlertoleranz als weitere Zielfunktion

$$\text{minimiere} \quad \mathbf{y} = \mathbf{F}(\mathbf{w}) = (F_1(\mathbf{w}), F_2(\mathbf{w}))^T \quad (4.80)$$

$$\text{mit} \quad \mathbf{g}(\mathbf{w}) = (g_1(\mathbf{w}), g_2(\mathbf{w}), \dots, g_k(\mathbf{w}))^T \leq \mathbf{0} \quad (4.81)$$

$$\mathbf{h}(\mathbf{w}) = (h_1(\mathbf{w}), h_2(\mathbf{w}), \dots, h_p(\mathbf{w}))^T = \mathbf{0} \quad (4.82)$$

$$\text{wobei} \quad F_1(\mathbf{w}) = \frac{1}{M} \sum_{i=1}^M \left( y^i - \sum_{\nu=1}^m \alpha_{\nu} \exp \left( \frac{-\|\mathbf{x}^i - \mathbf{c}_{\nu}\|^2}{2\sigma_{\nu}^2} \right) \right)^2$$

$$F_2(\mathbf{w}) = \sum_{k=1}^m s_k + b^2, \quad (4.83)$$

wobei  $s_k$  die Bedeutung der Gaußfunktionen nach (4.8) darstellt. Für die zweite Zielfunktion wird die Summe über alle Bedeutungen der Neuronen gebildet und somit werden die ursprünglichen  $m$  zusätzlichen Zielfunktionen<sup>10</sup> über diese gewichtete Summe zu einer einzigen Zielfunktion zusammengefasst. Dieses Vorgehen ist zulässig (siehe Erklärungen vorherigen Abschnitt) und besitzt für konvexe Zielfunktionen keine gravierenden Nachteile [CS04b]. Im Fall der Gleichverteilung der

<sup>10</sup>Das Ziel der Optimierung ist die Bedeutung eines jeden Neurons so gering wie möglich zu halten. So entstehen  $m$  zusätzliche Zielfunktionen.

**Tabelle 4.9:** Lösungen des Mehrzieloptimierungsproblems aus (4.80) mit Hilfe verschiedener Verfahren

Funktion	m	Verfahren	mse Training	mse Test	$F_2(\mathbf{w})$
$f_1$	4	NETLAB	42,3817	42,6380	$5,84 \cdot 10^{10}$
		Zielerreichung	26,7468	28,2028	$3,70 \cdot 10^{10}$
		Fehler-beschränkt	43,3665	44,1899	$5,84 \cdot 10^{10}$
		mse-beschränkt	42,3441	42,3469	$5,94 \cdot 10^{10}$
		$\sum_i  \alpha_i ^2$	41,7876	42,0967	$5,90 \cdot 10^{10}$
$f_2$	8	NETLAB	0,5393	0,5159	$4,02 \cdot 10^8$
		Zielerreichung	0,3556	0,3552	$5,90 \cdot 10^7$
		Fehler-beschränkt	0,5393	0,5159	$4,02 \cdot 10^8$
		mse-beschränkt	0,5375	0,4809	$4,23 \cdot 10^7$
		$\sum_i  \alpha_i ^2$	0,2783	0,2765	$1,84 \cdot 10^8$
$f_3$	6	NETLAB	0,6562	0,6804	$1,51 \cdot 10^4$
		Zielerreichung	0,2743	0,3305	$6,04 \cdot 10^3$
		Fehler-beschränkt	0,2227	0,3132	$1,34 \cdot 10^3$
		mse-beschränkt	0,6562	0,6972	7,63
		$\sum_i  \alpha_i ^2$	0,2437	0,2603	$3,05 \cdot 10^5$

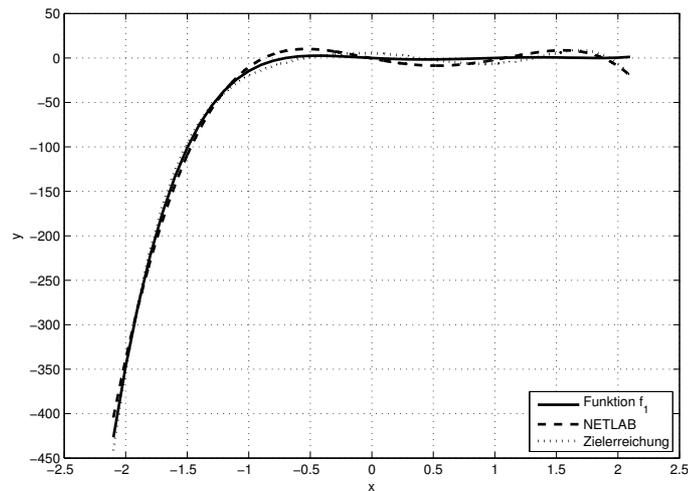
Eingänge und der Vereinfachung der Bedeutung nach (4.17) ergeben sich konvexe Funktionen für die Bedeutung eines Neurons.

Bei einem RBF-Netz nach (2.28) wird zumeist ein zusätzlicher, konstanter Term  $b$  im Ausgangsneuron addiert, um so den Gleichanteil bei der Funktionsapproximation abbilden zu können. Dieser wurde bei der Bestimmung der Bedeutung in Abschnitt 4.1 nicht berücksichtigt. Wenn dieser konstante Term durch einen Fehler entfällt, entsteht ein mittlerer quadratischer Fehler  $mse = b^2$ . Um diesen möglichst klein zu halten, müssen kleine Gleichanteile verwendet werden. Deswegen wird dieser Einfluss mit in der Zielfunktion  $F_2(\mathbf{w})$  berücksichtigt.

Neben der Zielfunktion (4.83) kann auch die Summe der Betragsquadrate der Ausgangsgewichte und des Gleichanteils als alternative Zielfunktion verwendet werden, da die Bedeutung der Neuronen stark von dem Ausgangsgewicht (vgl. Abschnitt 4.1) abhängig ist. Diese alternative Zielfunktion versucht somit den Betrag der Gewichte zu minimieren, was im Gegensatz zu dem Approximationsfehler steht.

Tabelle 4.9 zeigt die Ergebnisse verschiedener Lösungen des MOPs nach (4.80). Dabei gelten die gleichen Randbedingungen und Anmerkungen wie im vorangegangenen Abschnitt. Daneben wurde das Verfahren der Zielerreichung auf die alternative Zielfunktion angewendet, bei der die Summe über die Beträge der Gewichte gebildet wird. Allerdings wurde für dieses Verfahren in Tabelle 4.9 der Funktionswert von  $F_2(\mathbf{w})$  aus (4.83) zum besseren Vergleich angegeben.

Die Tabelle zeigt, dass die NETLAB-Lösungen keine pareto-optimalen Punkte darstellen. Es lassen sich Verbesserungen in beiden Zielfunktionen für alle Testfunktio-



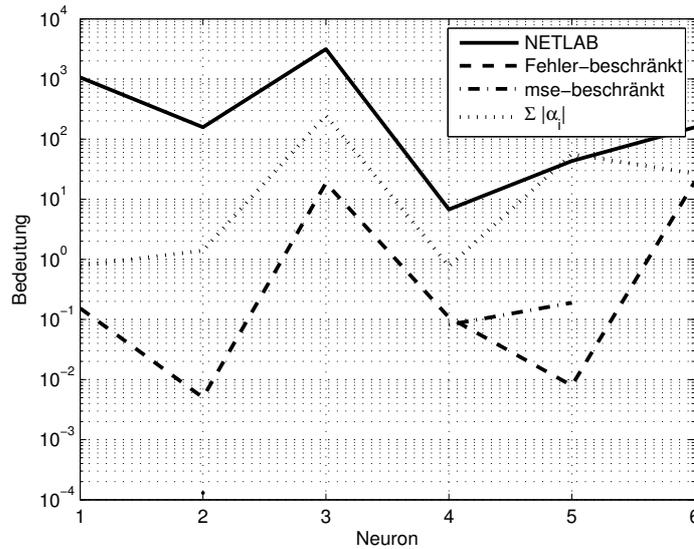
**Abbildung 4.12:** Approximation der Funktion  $f_1$  mit Fehlertoleranz als zusätzliche Randbedingung

nen erreichen. Da es sich auch bei diesem Mehrzieloptimierungsproblem nicht um konvexe Zielfunktionen handelt, können die Methoden in lokalen Minima festsitzen, wie z.B. für die Fehler-beschränkte oder mse-beschränkte Lösung der Funktion  $f_1$ .

Die alternative Zielfunktion über die Summe der Betragsquadrate stellt dabei einen guten Kompromiss für ein späteres Lernverfahren zur Laufzeit dar. Dass bei der Verwendung der alternativen Zielfunktion auch schlechtere Werte für die Fehlertoleranz entstehen, ist dadurch bedingt, dass in Tabelle 4.9 die Funktionswerte der Zielfunktion  $F_2(\mathbf{w})$  aus (4.83) angegeben werden. Die Optimierung erfolgt aber hinsichtlich einer anderen Zielfunktion, so dass nicht notwendigerweise die alternative Zielfunktion zu geringeren Werten als die ursprüngliche Zielfunktion führen muss. Aufgrund der geringen Anforderungen an die Rechenleistung und einer konvexen Zielfunktion ist die alternative Zielfunktion mit der Summe der Betragsquadrate eine gute Wahl für ein Verfahren zur Laufzeit.

Abbildung 4.12 zeigt die Approximation eines Netzes mit der NETLAB-Lösung und einer optimierten Lösung (Zielerreichung) für die Testfunktion  $f_1$ . Die ursprüngliche Lösung kann in beiden Zielfunktionen verbessert werden (vgl. Tabelle 4.9), falls leicht unterschiedliche Zentren und kleinere Gewichte verwendet werden. Dadurch kann das Überschwingen der NETLAB-Lösung verringert werden, was den mittleren quadratischen Fehler verkleinert. Als Konsequenz eines kleineren Gewichtes verbessert sich auch die Fehlertoleranz des Netzes.

Abbildung 4.13 zeigt die Bedeutung der Neuronen nach dem Lösen des MOPs für die Testfunktion  $f_3$ . Für das mse-beschränkte Netz mit maximaler Fehlertoleranz ist die Bedeutung der ersten drei Neuronen kleiner als die Skalierung der Abbildung, ebenso für das letzte Neuron. Die Bedeutung dieser Neuronen besitzt sehr kleine



**Abbildung 4.13:** Bedeutung der Neuronen nach Bestimmung der Lösung von (4.80) für die Testfunktion  $f_3$

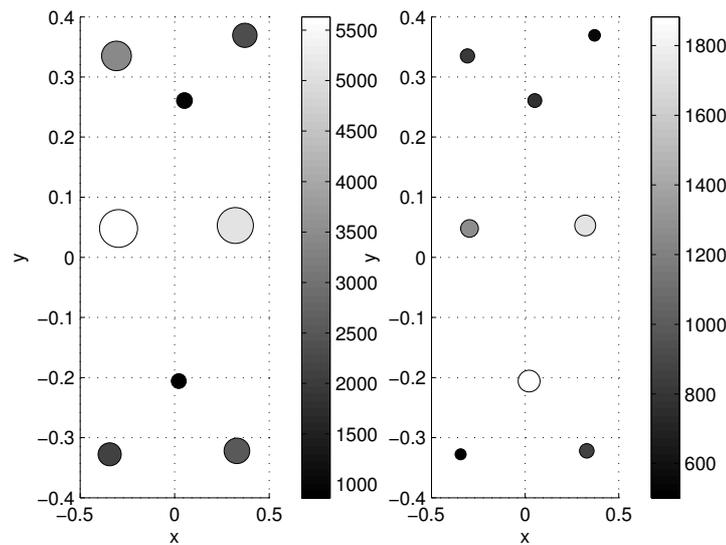
Werte und verschwindet. Allerdings ergibt sich bei dieser Testfunktion aufgrund seiner Form die Gefahr, ungünstige Paretopunkte auszuwählen. Wie man an Tabelle 4.9 und an Abbildung 4.13 erkennt, bietet die mse-beschränkte Lösung eine hohe Fehlertoleranz. Dies ist dadurch bedingt, dass in diesem Fall die Ausgabe des Netzes eine konstante Funktion mit  $f(\mathbf{x}) \approx 0$  beschreibt. Für diesen Fall ergeben sich sehr kleine Gewichte und eine hohe Fehlertoleranz. Aufgrund der Form der Testfunktion ist der mittlere quadratische Fehler jedoch gering und in der Größe der ursprünglichen Lösung. Diese pareto-optimale Lösung ist jedoch nicht wünschenswert, da das neuronale Netz nichts über die Form der Funktion aussagt. Um solche Punkte von der Lösung anzuschließen, muss das MOP in (4.80) um entsprechende Randbedingungen erweitert werden. Diese können z.B. eine untere Grenze für die Varianzen der Gaußfunktionen oder

$$\text{Var}(f_m(\mathbf{x})) = \text{Var}(y) \quad (4.84)$$

umfassen, wobei  $y$  die Ausgabe der Trainingsdaten und  $\text{Var}(\cdot)$  die Varianz bezeichnet. Durch die Randbedingung der Varianz wird eine konstante Funktion ausgeschlossen. Durch diese zusätzlichen Einschränkungen verringert sich der mögliche Lösungsraum, da nicht sinnvolle Lösungen, wie z.B. eine konstante Funktion, ausgeschlossen werden.

#### 4.3.2.1 Auswirkungen auf die Parameter

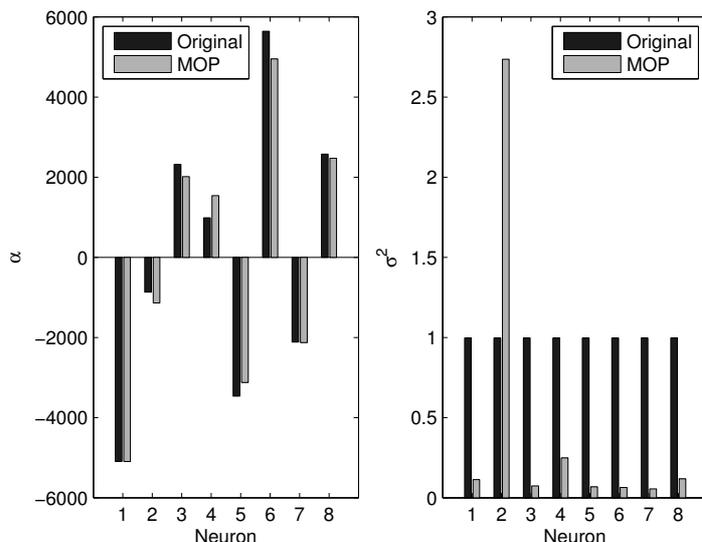
Wie schon zuvor für die Robustheit des Netzes soll an einem Beispiel der Einfluss der Fehlertoleranz als zusätzliche Zielfunktion bei der Optimierung betrachtet



**Abbildung 4.14:** Verteilung der Zentren und Darstellung  $\alpha \cdot \sigma$  für das ursprüngliche RBF-Netz (links) und nach dem MOP (rechts)

werden. Hier gelten analoge Ausführungen wie zuvor bei der Robustheit. Da im Allgemeinen kein konvexes Problem vorliegt, hängt die Bestimmung des Minimums von dem Optimierungsverfahren ab. Es besteht die Möglichkeit, nur lokale Minima zu finden. Da ebenfalls der Lösungsraum von den Randbedingungen sowie von der zu approximierenden Funktion abhängig ist, können somit keine allgemeingültigen Aussagen getroffen werden. Beispielhaft sollen die Effekte auf die Parameter anhand der Funktion  $f_2$  aus Tabelle 4.9 untersucht werden, wobei ebenfalls die mse-beschränkte Lösung betrachtet wird. Diese wird aufgrund des Fokus der Arbeit auf fehlertoleranten Netzen ausgewählt.

Abbildung 4.14 zeigt die Auswirkung der MOP auf die Verteilung der Zentren sowie die Fehlertoleranz, wobei als Maß das Produkt  $\alpha \cdot \sigma$  (vgl. Abschnitt 4.1) angewendet wird. Die Größe der Kreise erlaubt einen Vergleich der beiden Netze miteinander, während die Farbe einen Vergleich der Neuronen innerhalb eines Netzes ermöglicht. Hier zeigt sich, dass die Position der Zentren nicht durch den zusätzlichen Optimierungsschritt verändert wird. So kann auch hier eine Reduktion des Optimierungsproblems wie zuvor vorgenommen werden, indem auf eine Optimierung hinsichtlich der Zentren verzichtet wird. Allerdings ergibt sich eine bessere Fehlertoleranz des Netzes durch das Absenken des Produkts  $\alpha \cdot \sigma$ . Bei diesem Optimierungsschritt ändert sich die zusätzlich die Reihenfolge der Bedeutung einzelner Basisfunktionen. Eine Gaußfunktion mit einer zuvor geringen Bedeutung wird nach dem Training zu einer signifikanten Funktion. Somit ist diese Methode nicht treu hinsichtlich der Reihenfolge in der Bedeutung.



**Abbildung 4.15:** Höhe und Breite der Gaußfunktionen bei Fehlertoleranz

Abbildung 4.15 zeigt die Höhe und die Breite der Gaußfunktion des ursprünglichen Netzes vor und nach dem MOP. Auch hier zeigt sich, dass die Verbesserung der Fehlertoleranz hauptsächlich durch die Veränderung der Breite einer Gaußfunktion erreicht wird, während die Gewichte der Neuronen nur innerhalb kleiner Bereiche verändert werden. Aufgrund der multiplikativen Verknüpfung der Höhe und der Breite mit der Fehlertoleranz bzw. der Bedeutung ergibt sich eine bessere Fehlertoleranz des gesamten Netzes. Als Konsequenz aus diesem Beispiel muss für die Varianz der Gaußfunktionen eine hinreichend große untere Grenze festgelegt werden, um bei einer hohen Anzahl von Gaußfunktionen die Trainingsdaten nicht überangepasst zu approximieren. Das MOP versucht die Varianz der Gaußfunktionen zu minimieren, wobei im Extremfall jeder (Trainings-) Punkt durch eine Basisfunktion mit minimaler Breite dargestellt wird. So entsteht ein überangepasstes Modell mit schlechten Generalisierungseigenschaften, welches durch eine untere Schranke in den Varianzen vermieden wird. Die Wahl dieser Schranke hängt von der jeweiligen Funktion und der Größe des Netzes (Anzahl von Neuronen) ab.

### 4.3.3 Signifikanz als Randbedingung

Im vorherigen Abschnitt wurde zur Bestimmung der pareto-optimalen Punkte die Bedeutung der Neuronen unter anderem als zusätzliche Randbedingung verwendet. So konnte die Fehlertoleranz des Netzes einer ursprünglichen Lösung verbessert werden. Daneben kann die Fehlertoleranz auch schon beim initialen Training als

Randbedingung verwendet werden. Hierfür ergibt sich analog ein Optimierungsproblem

$$\min F(\mathbf{w}) = \min \frac{1}{M} \sum_{i=1}^M (f_m(\mathbf{x}^i, \mathbf{w}) - y^i)^2 \quad \mathbf{w} \in \mathbb{R}^{m(n+2)} \quad (4.85)$$

$$s_k(\mathbf{w}) - b_k \leq 0 \quad \text{für } k = 1 \dots m, \quad (4.86)$$

wobei  $s_k$  die Bedeutung nach (4.8) bezeichnet und  $b_k$  einen maximalen Wert für die Bedeutung beschreibt. So entstehen neben den anderen Randbedingungen  $m$  zusätzliche Randbedingungen für das Optimierungsproblem.

Allerdings sind die absoluten Werte der Bedeutung vor dem Training nicht bekannt, so dass die Auswertung der Randbedingungen nach (4.86) nur schwer möglich ist. Jedoch werden beim Lernen die Lagen der Zentren sowie die einzelnen Varianzen der Basisfunktionen zumeist mit Hilfe anderer Algorithmen bestimmt. So können z.B. die Lagen der Zentren durch den *k-means* Algorithmus bestimmt werden und die Werte der Varianzen durch die Distanzen zwischen den Zentren festgelegt werden, vgl. die Vorgehensweise von *NETLAB* [Nab02]. Sind diese Parameter nun durch andere Algorithmen bestimmt worden, bleiben als Variablen für das Optimierungsproblem aus (4.85) nur noch die Ausgangsgewichte übrig, und es ergibt sich ein lineares Optimierungsproblem. Zusätzlich sind die Randbedingungen aus (4.86) nur noch linear von den freien Variablen abhängig. Damit resultiert die Randbedingung (4.86) in eine obere Grenze für die Ausgangsgewichte, was zusätzlich die gleichgradige Stetigkeit und damit die Robustheit des Netzes zur Folge hat, vgl. Kapitel 3.1. Werden die  $m$  zusätzlichen Randbedingungen über die gewichtete Summe zusammengefasst, so ergibt sich ebenfalls eine obere Grenze für die Gewichte, da die Summe beschränkt sein muss.

Wird das Optimierungsproblem aus (4.85) für alle Parameter des Netzes gelöst, so ergibt sich ebenfalls eine obere Schranke für die Gewichte. Nach Abschnitt 4.1 gilt für die Bedeutung der Neuronen

$$s_k = E [d^q(y, \hat{y})] = |\alpha_k|^q \int_{\underline{X}} e^{-\frac{q \|\mathbf{x} - \mathbf{c}_k\|^2}{2\sigma_k^2}} p_{\underline{X}}(\mathbf{x}) d\mathbf{x}. \quad (4.87)$$

Da das Integral konvergiert (siehe Abschnitt 4.1), folgt direkt aus den Randbedingungen von (4.86), dass die Ausgangsgewichte des Netzes beschränkt sein müssen, um Fehlertoleranz des Netzes sicherzustellen.

Somit ist die Randbedingung zur Erreichung einer bestimmten Fehlertoleranz des Netzes gleichbedeutend mit dem Einführen einer oberen Grenze für die Gewichte, also äquivalent zu

$$|\alpha_k| \leq B \quad \text{für } k = 1 \dots m. \quad (4.88)$$

Diese Randbedingung kann zu dem Einfügen zusätzlicher Neuronen führen, da zum Erreichen einer bestimmten Fehlertoleranz die Randbedingungen nicht erfüllt werden können und so keine Lösung von (4.85) gefunden werden kann. So impliziert die gleichgradige Stetigkeit des Netzes ebenfalls ein fehlertolerantes Netz.

*Beispiel.* Als Beispiel soll die Funktion

$$g(x) = 5 e^{-x^2} + 2 e^{-(x-1)^2} \quad (4.89)$$

durch ein RBF-Netz approximiert werden. Für diese Aufgabe sind 2 Basisfunktionen mit Zentren  $c_1 = 0$  und  $c_2 = 1$  und Varianzen  $\sigma = \sqrt{1/2}$  ausreichend. Allerdings besitzen die beiden Neuronen unterschiedliche Bedeutung ( $q = 2$ )

$$s_1 \propto 5^2 \quad s_2 \propto 2^2 \quad (4.90)$$

und normiert auf das Quadrat des größten Ausgangsgewichts

$$s_1 \propto 1 \quad s_2 \propto \frac{4}{25}. \quad (4.91)$$

Wird nun die Randbedingung eingeführt, dass die relative Bedeutung jedes einzelnen Neurons  $s_k \leq 0,2$ , bezogen auf die maximale ursprüngliche Signifikanz, nicht übersteigen darf, folgt für die Gewichte  $\alpha_k \leq \sqrt{5}$ . Somit werden anstatt 2 Neuronen für eine gültige Lösung mindestens vier Neuronen benötigt, wobei dieses Vorgehen dem Einfügen von Neuronen entspricht. Ohne ein Einfügen von zusätzlichen Basisfunktionen und mit Hilfe zweier Gaußfunktionen lässt sich das Optimierungsproblem mit der Fehlertoleranz als Randbedingung nicht lösen.

*Bemerkung.* Bei den Untersuchungen zu dem robusten und fehlertoleranten Lernen zeigte sich, dass die in NETLAB verwendete Vorgehensweise zwar RBF-Netze mit guten Approximationseigenschaften liefert, jedoch diese Netze aufgrund ihrer Parameter schlechte Eigenschaften bzgl. Robustheit und Fehlertoleranz sowie der Umsetzung in Hardware besitzen. Beispielsweise sei das RBF-Netz zur Approximation der si-Funktion untersucht (siehe Abbildung 2.10). Der in NETLAB verwendete Algorithmus bestimmt die Lagen der Zentren basierend auf der Verteilungsdichte der Testdaten, in diesem Fall also gleichverteilte Zentren im Intervall  $[-10; 10]$ . Die Varianzen ergeben sich aus dem Maximum des quadratischen Abstandes zwischen den einzelnen Zentren. So entstehen in diesem Fall sehr große Breiten ( $\propto 10^2$ ) für die Gaußfunktionen. Dies bedingt für gute Approximationseigenschaften ebenfalls hohe Gewichte. Als Folge hieraus ergeben sich aufgrund des Zusammenhanges mit den Parametern (siehe Kapitel 3 und Abschnitt 4.1) eine niedrige Robustheit und eine schlechte Fehlertoleranz. Ebenfalls ist eine Umsetzung in Hardware sehr aufwendig, da sehr hohe Werte für die Parameter gespeichert werden müssen, was zum einen den Speicherplatz erhöht, aber auch z.B. in der digitalen Umsetzung die Komplexität der Recheneinheiten erhöht. Dies resultiert in einen deutlich höheren

**Tabelle 4.10:** Eigenschaften unterschiedlicher Trainingsverfahren

Verfahren	mse	Robustheit	Fehlertol.
NETLAB	$1,60 \cdot 10^{-3}$	$\propto 10^8$	$\propto 10^{21}$
NETLAB (mod.)	$1,09 \cdot 10^{-5}$	0,1792	0,1881
Gradient	$1,16 \cdot 10^{-2}$	0,6612	0,0073

Flächenbedarf [Rab96, BK80]. Tabelle 4.10 zeigt die Auswirkungen verschiedener Trainingsverfahren auf die Eigenschaften<sup>11</sup> des RBF-Netzes zur Approximation der  $\text{si}$ -Funktion. Dafür wurde das Verfahren aus NETLAB in seiner ursprünglichen Form und in einer modifizierten Variante sowie das Gradientenverfahren genutzt. Bei der modifizierten Variante werden die Varianzen der Gaußfunktionen einer Gleichverteilung im Intervall  $[10; 10]$  entnommen, so dass wesentlich kleinere Breiten und damit kleinere Ausgangsgewichte entstehen. Es zeigt sich, dass mit den anderen beiden Verfahren deutlich robustere und fehlertolerantere Netze entworfen werden können, wobei die Approximationseigenschaften ähnliche Güten besitzen.

## 4.4 Zusammenfassung und Diskussion

Das Kapitel 4 beschäftigte sich mit der Methodologie zur Verbesserung der Fehlertoleranz und Robustheit neuronaler Netze, wobei ein wesentlicher Vorteil der hier vorgestellten Methoden die Unabhängigkeit der Verfahren von dem verwendeten Trainingsalgorithmus ist. So ergibt sich eine allgemeine Anwendbarkeit der entwickelten Techniken.

Hinsichtlich der Fehlertoleranz wurden Methoden entwickelt, mit denen die Redundanz an geeigneten Stellen im Netz erhöht werden kann. Basierend auf dem statistischen Beitrag eines Neurons zu einem Fehler im Ausgang, der aufgrund unzuverlässiger Recheneinheiten entsteht, kann die Bedeutung eines Neurons definiert werden. Sowohl für RBF-Netze als auch das LCNN lässt sich ein solches Maß definieren. Simulationsergebnisse haben gezeigt, dass diese Verfahren diejenigen Neuronen erkennen, die bei einem Ausfall den größten Approximationsfehler im Ausgang erzeugen. So lassen sich die vorgestellten Verfahren aus Kapitel 3.3 zuerst für die Neuronen anwenden, welche eine hohe Bedeutung besitzen, und dadurch minimiert sich der zusätzliche Flächenbedarf. Neben der Verbesserung der Fehlertoleranz kann die Bedeutung der Neuronen auch zur Kontrolle der Modellkomplexität verwendet werden. Dies ist für gute Approximations- und Generalisierungseigenschaften eines

<sup>11</sup>Die Robustheit des Netzes wurde durch den Erwartungswert der Empfindlichkeit der Neuronen nach (4.67) bestimmt, und als Maß für die Fehlertoleranz wurde der Erwartungswert der Bedeutung aller Neuronen ermittelt.

neuronalen Netzes ein wesentlicher Bestandteil, da negative Effekte, wie überangepasste oder hoch komplexe Modelle, verhindert werden können.

Ähnliche Betrachtungen wurden für die Robustheit eines RBF-Netzes durchgeführt. Es lassen sich die Neuronen bzw. die Parameter mit der höchsten Empfindlichkeit gegenüber Rauschen identifizieren. Mit Hilfe dieser Maße können entsprechende Verfahren flächeneffizient angewendet werden, weil eine Reihenfolge in der Notwendigkeit der Verfahren festgelegt werden kann.

So können mit Hilfe beider Maße, für die Fehlertoleranz und die Robustheit, Aussagen über die optimalen Parameter eines Netzes getroffen werden. Für fehlertolerante Netze müssen möglichst kleine Ausgangsgewichte und schmale Breiten der lokalen Funktionen verwendet werden, während ein robustes Netz hohe Breiten erfordert. Hier zeigt sich der gegenläufige Trend in den Breiten der lokalen Funktionen, so dass die Breite für eine optimale Approximation ausgelegt werden sollte. Um möglichst kleine Gewichte zu erhalten, empfiehlt sich das Einfügen von identischen Neuronen. So wird eine Aufgabe von mehreren Neuronen übernommen, und die Fehlertoleranz des Netzes steigt. Die Maße zur Robustheit und Fehlertoleranz können in Entwurfswerkzeuge integriert werden, so dass diese automatisch bei Kenntnis der technologischen Randbedingungen das optimale System hinsichtlich Fehlertoleranz oder Robustheit bestimmen können.

Daneben wurden Verfahren vorgestellt, welche durch geschickte Wahl der Parameter versuchen, das Verhalten der Netze gegenüber Rauschen und dem Auftauchen von Fehlern zu verbessern. Die vorgestellten Maße können dabei als zusätzliche Ziele in einem Mehrzieloptimierungsproblem zum Trainieren der neuronalen Netze angewendet werden, um ohne eine Veränderung der Struktur bessere Netze zu erhalten. Eine allgemeine Aussage über die Auswirkungen auf die Parameter des Netzes kann nicht getroffen werden, da diese von der möglichen Lösung im Urbildraum abhängen, der wiederum von der zu approximierenden Funktion abhängig ist. Bei der Mehrzieloptimierung wurde nur eine zusätzliche Zielfunktion neben den Approximationseigenschaften der Netze betrachtet. Eine Anwendung aller Maße, Robustheit und Fehlertoleranz, ist jedoch denkbar. Zur weiteren Verbesserung der Eigenschaften können Verfahren aus der Literatur angewendet werden, wie z.B. die Hauptkomponentenanalyse [Pea01b, Hay99], die Unabhängigkeitsanalyse [HKO01, HHO98] oder Tests auf statische Unabhängigkeit von Merkmalen [FWV06]. Mit diesen Techniken können die Komplexität reduziert oder Störungen herausgefiltert werden.

# Technische Umsetzung

Dieses Kapitel betrachtet die Auswirkungen der Umsetzung neuronaler Verfahren in analoge und digitale Schaltungstechniken. Dazu werden zuerst numerische Betrachtungen der Verfahren durchgeführt, welche die Auswirkungen von Ungenauigkeiten während der Berechnung erfassen und so helfen, die Umsetzung der Methoden zu verbessern. Weiter wird eine parallele, rekonfigurierbare, digitale Struktur vorgestellt, die unterschiedliche Modelle von Neuronen und verschiedene Topologien von neuronalen Netzen abbilden kann. Analyseergebnisse der Umsetzung dieser Struktur auf einen aktuellen CMOS-Prozess und ein Vergleich eines analogen und digitalen Neurons, auch im Hinblick auf die Skalierung der CMOS-Technologie, finden sich am Ende des Kapitels.

## 5.1 Numerische Betrachtungen

Die Fehlerfortpflanzung bei der Auswertung von mathematischen Ausdrücken kann zu unerwünschten Effekten führen. Auch gibt es mehrere unterschiedliche Vorgehensweisen einen Ausdruck zu berechnen, wobei die einzelnen Verfahren sich unterschiedlich gut oder schlecht verhalten können. Im Folgenden sollen dabei die Auswirkungen von relativen Fehlern der Grundrechenarten auf den Ausgang der neuronalen Netze betrachtet werden, indem die Konditionierung und die numerische Stabilität der neuronalen Prinzipien untersucht werden.

### 5.1.1 Konditionierung der Gaußfunktionen

Bei der Darstellung von Zahlen und bei der Berechnung von Funktionen mit Hilfe von Recheneinheiten entstehen Fehler aufgrund von Rundungen und Fehlerfortpflanzungen. Hier können Konditionszahlen definiert werden, welche die relative Empfindlichkeit eines Ergebnisses gegenüber Störungen der Eingangsgrößen beschreibt. Jene beschreiben exakt das Verhalten der Koeffizientenempfindlichkeit aus Abschnitt 3.3, so dass diese Ergebnisse hier angewendet werden können. Die Empfindlichkeit ist allerdings bezüglich der Parameter definiert, während die Konditionierung sich auf die Störungen der Eingangsgrößen bezieht.

Die Konditionszahlen werden dabei wie folgt definiert [Sch98a]:

**Definition 8** (Kondition). Seien  $\|\cdot\|_X, \|\cdot\|_Y$  geeignete Normen auf  $X$  und  $Y$  sowie

$$\delta_x = \frac{\|\Delta x\|_X}{\|x\|_X} \quad \delta_y = \frac{\|\Delta y\|_Y}{\|y\|_Y} \quad (5.1)$$

die relativen Ein- bzw. Ausgabefehler. Dann heißt

$$\kappa = \frac{\delta_y}{\delta_x} \quad (5.2)$$

die Kondition des Problems  $f(x), f : X \rightarrow Y$ .

Das Problem ist gut konditioniert, wenn sich kleine Schranken für  $\kappa$  in dem Fall  $\delta_x \rightarrow 0$  ergeben.

Bei der Gaußfunktion als Basisfunktion folgt für die Kondition hinsichtlich der Eingangsvariable  $x_i$

$$\kappa_i = \frac{\alpha x_i (x_i - c_i) \exp(\cdot)}{\alpha \sigma^2 \exp(\cdot)} = \frac{x_i (x_i - c_i)}{\sigma^2}. \quad (5.3)$$

Als Ergebnis aus (5.3) ergibt sich eine gute Konditionierung des Problems für große Varianzen und für Werte der Eingänge nahe der auf die entsprechende Achse projizierten Zentren. Als Folge daraus sollten für geringe Empfindlichkeiten und eine gute Konditionierung der Gaußfunktionen große Varianzen eingesetzt werden, was konsistent zu den Ergebnissen aus Kapitel 3 ist. Der Nachteil breiter Gaußfunktionen besteht zum einen in dem großen Flächenbedarf, der zur Speicherung durch Kapazitäten oder Register benötigt wird. Zum anderen müssen bei hohen Varianzen auch große Gewichte für gute Eigenschaften der Approximation verwendet werden [Nab02]. Zudem lässt sich der Einsatz großer Breiten aufgrund der gewünschten Approximationseigenschaften nicht immer durchführen, und die Fehlertoleranz des Netzes nimmt ab (vgl. Abschnitt 3.2).

Die begrenzte Auflösung technischer Systeme führt zu einer oberen Schranke der Kondition. Hier gelte für die Differenz  $(x_i - c_i) \leq P$  und für die Eingänge  $x_i \leq M$ . So ergibt sich die obere Schranke zu

$$\kappa_i \leq T^2 MP. \quad (5.4)$$

Somit kann durch geschickte Wahl der Auflösung die Konditionierung des Problems beeinflusst und nach oben beschränkt werden. Aufgrund des großen Einflusses der Varianzen sollten diese nicht zu klein gewählt werden und bei einer digitalen Darstellung der Anteil des Bruches möglichst gering gehalten werden, was sich zudem positiv auf die benötigte Fläche auswirkt.

### 5.1.2 Polynomdarstellung und Horner-Schema

Bei der Approximation einer Basisfunktion durch ein Polynom  $q$ -ten Grades

$$y = p(x) = a_q x^q + a_{q-1} x^{q-1} + \dots + a_1 x + a_0 \quad (5.5)$$

werden als Parameter die Koeffizienten  $a_i, i = 0 \dots q$  benötigt. Da diese bei einer Speicherung quantisiert werden müssen, ergibt sich ein Fehler zu dem ursprünglichen Polynom.

Die absolute Konditionszahl für ein solches Interpolationspolynom ist durch [Sch04]

$$\kappa_{\text{abs}} = \max_{x \in [a; b]} \sum_{i=0}^n |L_i(x)| = \Lambda_q \quad (5.6)$$

gegeben, wobei  $L_i(x)$  das Lagrange-Polynom bezeichnet. Hierbei wird die absolute Kondition stark von der Wahl der Stützstellen beeinflusst. Werden äquidistante Stützstellen verwendet, so nimmt die Lebesgue-Konstante  $\Lambda_q$  ihr Maximum an, während sie bei einer Wahl nach Tschebyscheff minimiert wird [Sch04]. Dabei verbessert sich die Kondition des Problems je kleiner das Intervall der Approximation gewählt wird. Dies ist besonders für die Umsetzung mit Hilfe digitaler Rechenelemente in Abschnitt 5.2 wichtig.

Wird das Polynom (5.5) durch das Horner-Schema [BSM05] dargestellt, so ergibt sich für die absolute Konditionszahl

$$\kappa_{\text{abs}} = \max_{i=0, \dots, q} |a_i - \hat{a}_i| \sum_{i=0}^q |x^i|. \quad (5.7)$$

Weiter gilt

$$\sum_{i=0}^q |x^i| = \frac{x^{q+1} - 1}{x - 1}, \quad (5.8)$$

so dass innerhalb eines Intervalls  $[0; x_b]$  gilt

$$\kappa_{\text{abs}} \leq \max_{i=0, \dots, q} |a_i - \hat{a}_i| \frac{x_b^{q+1} - 1}{x_b - 1}. \quad (5.9)$$

Nun soll untersucht werden, welcher Effekt zu einer größeren Reduktion der Konditionszahl  $\kappa_{\text{abs}}$  führt: eine Verringerung der Intervallbreite von  $[0; x_b]$  auf  $[0; x_{b'}]$  oder eine Erhöhung der Auflösung zur Darstellung der Koeffizienten.

Bei der Quantisierung der Koeffizienten wird angenommen, dass diese zur nächsten Maschinenzahl gerundet werden, um möglichst geringe absolute Fehler zu erreichen.

Dabei gilt für die Auflösung und den maximalen Fehler durch die Speicherung der Koeffizienten mit begrenzter Genauigkeit

$$\max_{i=0,\dots,q} |a_i - \hat{a}_i| \leq \frac{2^{-l}}{2} = 2^{-(l+1)}, \quad (5.10)$$

wobei  $l$  die Anzahl der binären Stellen für den Bruch bei einer Festpunktdarstellung bezeichnet. Damit folgt für die absolute Konditionszahl bei einer Berechnung in Festpunktarithmetik

$$\kappa_{\text{abs}} \leq \frac{2^{-l} x_b^{q+1} - 1}{2 x_b - 1} = 2^{-(l+1)} \frac{x_b^{q+1} - 1}{x_b - 1}. \quad (5.11)$$

Für eine Erhöhung der Auflösung folgt somit

$$\hat{\kappa}_{\text{abs}} \leq \frac{2^{-(l+1)} x_b^{q+1} - 1}{2 x_b - 1}. \quad (5.12)$$

Ein Vergleich mit (5.11) zeigt, dass die obere Grenze um den Faktor  $1/2$  sinkt, wenn eine zusätzliche Stelle zur Darstellung benutzt wird. Analog ergibt sich für insgesamt  $p$  weitere Stellen der Auflösung eine Reduktion um den Faktor  $1/2^p$ .

Für eine Reduktion des Intervalls gilt dagegen

$$\hat{\kappa}_{\text{abs}} \leq 2^{-(l+1)} \frac{x_{b'}^{q+1} - 1}{x_{b'} - 1}. \quad (5.13)$$

Ein Vergleich von (5.13) zur ursprünglichen oberen Grenze liefert

$$\frac{\hat{\kappa}}{\kappa} = \frac{(x_{b'}^{q+1} - 1)(x_b - 1)}{(x_b^{q+1} - 1)(x_{b'} - 1)}. \quad (5.14)$$

Damit verbessert sich die Konditionszahl bei einer Verringerung der Intervallbreite gegenüber der Erhöhung der Auflösung, wenn gilt

$$\frac{(x_{b'}^{q+1} - 1)(x_b - 1)}{(x_b^{q+1} - 1)(x_{b'} - 1)} \leq \frac{1}{2}. \quad (5.15)$$

Ein Vorteil der Reduktion der Intervallbreite ist neben der Verbesserung der Konditionszahl auch die Verkleinerung der Fläche bei einer Umsetzung in Hardware, da sich eine geringere Auflösung positiv auf die Rechenzeit und die Fläche einer Multiplikation oder einer Addition auswirkt [BK80]. Bei einer Reduzierung des Intervalls kann die Auflösung der Signale verringert werden, weil durch eine geringere Intervallbreite kleinere absolute Zahlen dargestellt werden müssen. So lässt sich der ganzzahlige Anteil bei der Festpunktarithmetik verkleinern, ohne dass sich dies auf

den maximalen Fehler bei der Speicherung der Koeffizienten auswirkt. Dies führt dann zu geringen Wortbreiten bei der Addition und der Multiplikation, was sich in einer reduzierten Fläche bemerkbar macht. Dagegen vergrößert sich bei einer Erhöhung der Auflösung die Wortbreiten für die Addition und die Multiplikation, und die benötigte Fläche für die Rechenoperationen steigt an [BK80, Rab96].

*Beispiel.* Als Beispiel soll hier die Reduktion der Intervallbreite von  $[0; 3]$  auf  $[0; 2]$  betrachtet werden. Es folgt für den Vergleich der oberen Grenzen

$$\frac{2(2^{q+1} - 1)}{(3^{q+1} - 1)} \leq \frac{1}{2}, \quad (5.16)$$

was für  $q = 3$  erfüllt ist. Somit ist ab einem Polynom dritten Grades in diesem Fall die Verringerung der Intervallbreite hinsichtlich der Konditionierung des Problems günstiger als die Erhöhung der Bruchbreite bei der Festpunktarithmetik.

Ebenso kann gezeigt werden, dass sich eine Reduzierung des Polynomgrades positiv auf die Konditionierung der Berechnung mittels Horner-Schema für alle  $x \in \mathbb{R}$  auswirkt, und die absolute Konditionszahl besonders bei langen Polynomen erheblich reduziert werden kann. So verringert sich diese von  $597871 \cdot 2^{-(l+1)}$  auf  $66430 \cdot 2^{-(l+1)}$ , wobei  $q = 6$  bzw.  $q = 5$  und  $x_b = 9$  gewählt wurde. Dies entspricht einer Verringerung der Konditionszahl um den Faktor 9 bei gleicher Auflösung. Für noch kleinere Polynome führt dies zu einer weiteren Verbesserung, so dass jene aufgrund der Konditionierung bessere Approximationseigenschaften als Polynome höheren Grades besitzen können (vgl. Abschnitt 5.3.1).

Welches Vorgehen, Erhöhung der Auflösung, Verringerung der Intervallbreite oder Grad des Approximationspolynoms, den größten Einfluss auf die absolute Konditionszahl besitzt, ist von dem aktuellen Problem abhängig. Für dieses kann dann das optimale Vorgehen bestimmt werden. Bei dem Vergleich der direkten Auswertung eines Polynoms und der Anwendung des Horner-Schemas unter der Berücksichtigung von Rundungsfehlern ergibt sich das numerisch stabilere Verfahren für das Horner-Schema [Sch98a, Sch04]. Somit sollte zur Auswertung von Polynomen dieses Verfahren angewendet werden.

### 5.1.3 Numerische Stabilität

Das Vorgehen, die Ausgabe des neuronalen Netzes zu berechnen, besteht i.A. aus mehreren unterschiedlichen Funktionen, die nacheinander angewendet werden. Daneben entstehen bei der Berechnung Fehler, die sich in diesen mehrstufigen Verfahren fortpflanzen und einen Fehler in der Ausgabe produzieren. Um dieses Verhalten zu beschreiben, wird die numerische Stabilität [Sch98a, Pla04] eines Algorithmus betrachtet:

**Definition 9** (numerisch stabil). *Ein Algorithmus heißt numerisch stabil oder gutartig, wenn die durch die Rechnung erzeugten Fehler in der Größenordnung des durch die Kondition des Problems unvermeidbaren Fehlers bleiben.*

Ein Algorithmus ist numerisch stabiler als ein anderer, wenn der Gesamteinfluss der Rundungsfehler bei dem ersten kleiner ist als bei dem zweiten Verfahren.

Für die Analyse wird zunächst angenommen, dass die Funktion  $\exp(x)$  durch eine Schaltung in digitaler oder analoger Hardware berechnet werden kann. In realen Anwendungen dagegen wird diese Funktion mittels der Grundrechenarten berechnet, so dass auch der relative Fehler  $\epsilon_e$  durch die relativen Fehler der Grundrechenarten ausgedrückt werden kann.

Für die erste Variante, Algorithmus I<sup>1</sup> aus Abbildung 3.4, zur Berechnung der Basisfunktionen folgt für die einzelnen Abbildungen (siehe auch Abschnitt 3.2.1)

$$\begin{aligned} \varphi^{(0)}(\mathbf{x}) &= \begin{pmatrix} x_1 - c_1 \\ \vdots \\ x_n - c_n \end{pmatrix} & \varphi^{(1)}(\mathbf{x}) &= \begin{pmatrix} x_1^2 \\ \vdots \\ x_n^2 \end{pmatrix} & \varphi^{(2)}(\mathbf{x}) &= \sum_{i=1}^n x_i & \varphi^{(3)}(x) &= \frac{-x}{2\sigma^2} \\ \varphi^{(4)}(x) &= e^x & \varphi^{(5)}(x) &= \alpha x, \end{aligned} \tag{5.17}$$

die zur Berechnung der Zwischenergebnisse durchgeführt werden, und für die Zwischenresultate nach jedem Zwischenschritt gilt

$$\begin{aligned} \mathbf{x}^{(0)} &= \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} & \mathbf{x}^{(1)} &= \begin{pmatrix} x_1 - c_1 \\ \vdots \\ x_n - c_n \end{pmatrix} & \mathbf{x}^{(2)} &= \begin{pmatrix} (x_1 - c_1)^2 \\ \vdots \\ (x_n - c_n)^2 \end{pmatrix} \\ \mathbf{x}^{(3)} &= \sum_{i=1}^n (x_i - c_i)^2 & \mathbf{x}^{(4)} &= -\frac{\|\mathbf{x} - \mathbf{c}\|^2}{2\sigma^2} \\ \mathbf{x}^{(5)} &= \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}\|^2}{2\sigma^2}\right) & \mathbf{x}^{(6)} &= \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}\|^2}{2\sigma^2}\right) \end{aligned} \tag{5.18}$$

sowie für die Restabbildungen, welche noch auszuführen sind, um das Ergebnis zu erhalten

$$\begin{aligned} \Psi^{(5)} &= \alpha x & \Psi^{(4)} &= \alpha e^x & \Psi^{(3)} &= \alpha e^{-\frac{x}{2\sigma^2}} \\ \Psi^{(2)} &= \alpha e^{-\frac{\sum_i x_i}{2\sigma^2}} & \Psi^{(1)} &= \alpha e^{-\frac{\sum_i x_i^2}{2\sigma^2}}. \end{aligned} \tag{5.19}$$

<sup>1</sup> Dieses Vorgehen orientiert sich an der mathematischen Beschreibung des RBFs und bestimmt zunächst die euklidische Distanz zwischen Eingang und Zentrum.

Damit folgt für die einzelnen Rundungsfehler bei Algorithmus I

$$D\Psi^{(1)}(\mathbf{x}^{(1)}) a_1 = -\frac{\alpha\epsilon_-}{\sigma^2} \exp(\cdot) \|\mathbf{x} - \mathbf{c}\|^2 \quad (5.20)$$

$$D\Psi^{(2)}(\mathbf{x}^{(2)}) a_2 = -\frac{\alpha\epsilon_*}{2\sigma^2} \exp(\cdot) \|\mathbf{x} - \mathbf{c}\|^2 \quad (5.21)$$

$$D\Psi^{(3)}(\mathbf{x}^{(3)}) a_3 = -\frac{\alpha\epsilon_+}{2\sigma^2} \exp(\cdot) \|\mathbf{x} - \mathbf{c}\|^2 \quad (5.22)$$

$$D\Psi^{(4)}(\mathbf{x}^{(4)}) a_4 = -\frac{\alpha\epsilon_/}{2\sigma^2} \exp(\cdot) \|\mathbf{x} - \mathbf{c}\|^2 \quad (5.23)$$

$$D\Psi^{(5)}(\mathbf{x}^{(5)}) a_5 = \alpha\epsilon_e \exp(\cdot) \quad (5.24)$$

$$a_6 = \alpha\epsilon_\alpha \exp(\cdot), \quad (5.25)$$

wobei  $\epsilon_i, i \in \{-; +; *; /; e; \alpha\}$  den relativen Fehler bei der jeweiligen Abbildung bezeichnet.

Somit kann der für Algorithmus I entstehende absolute Fehler berechnet werden, der sich aufgrund der Fortpflanzung der ungenauen Berechnungen ergibt, und es folgt für diesen

$$\begin{aligned} \Delta y_I = Dg(\mathbf{x})\Delta\mathbf{x} &- \frac{\alpha\epsilon_-}{\sigma^2} \exp(\cdot) \|\mathbf{x} - \mathbf{c}\|^2 - \frac{\alpha\epsilon_*}{2\sigma^2} \exp(\cdot) \|\mathbf{x} - \mathbf{c}\|^2 \\ &- \frac{\alpha\epsilon_+}{2\sigma^2} \exp(\cdot) \|\mathbf{x} - \mathbf{c}\|^2 - \frac{\alpha\epsilon_/}{2\sigma^2} \exp(\cdot) \|\mathbf{x} - \mathbf{c}\|^2 + \alpha\epsilon_e \exp(\cdot) + \alpha\epsilon_\alpha \exp(\cdot). \end{aligned} \quad (5.26)$$

Ebenso kann für die zweite Variante, Algorithmus II<sup>2</sup> aus Abbildung 3.5, der Basisfunktionen der absolute Fehler bedingt durch die ungenauen Berechnungen bestimmt werden. Dabei gilt für die auszuführenden Abbildungen

$$\begin{aligned} \varphi^{(0)}(\mathbf{x}) &= \begin{pmatrix} x_1 - c_1 \\ \vdots \\ x_n - c_n \end{pmatrix} \quad \varphi^{(1)}(\mathbf{x}) = \begin{pmatrix} x_1^2 \\ \vdots \\ x_n^2 \end{pmatrix} \quad \varphi^{(2)}(\mathbf{x}) = \frac{-1}{2\sigma^2} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \\ \varphi^{(3)}(x) &= \begin{pmatrix} e^{x_1} \\ \vdots \\ e^{x_n} \end{pmatrix} \quad \varphi^{(4)}(x) = \prod_{i=1}^n x_i \quad \varphi^{(5)}(x) = \alpha x \end{aligned} \quad (5.27)$$

<sup>2</sup> Dieses Verfahren orientiert sich an dem biologischen Vorbild und bestimmt die multivariate Gaußverteilung aus der Multiplikation der eindimensionalen Verteilungen.

sowie für die einzelnen Zwischenwerte der Berechnungen

$$\begin{aligned}
\mathbf{x}^{(0)} &= \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} & \mathbf{x}^{(1)} &= \begin{pmatrix} x_1 - c_1 \\ \vdots \\ x_n - c_n \end{pmatrix} & \mathbf{x}^{(2)} &= \begin{pmatrix} (x_1 - c_1)^2 \\ \vdots \\ (x_n - c_n)^2 \end{pmatrix} \\
\mathbf{x}^{(3)} &= \frac{-1}{2\sigma^2} \cdot \begin{pmatrix} (x_1 - c_1)^2 \\ \vdots \\ (x_n - c_n)^2 \end{pmatrix} & \mathbf{x}^{(4)} &= \begin{pmatrix} e^{-\frac{(x_1 - c_1)^2}{2\sigma^2}} \\ \vdots \\ e^{-\frac{(x_n - c_n)^2}{2\sigma^2}} \end{pmatrix} \\
\mathbf{x}^{(5)} &= \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}\|^2}{2\sigma^2}\right) & \mathbf{x}^{(6)} &= \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}\|^2}{2\sigma^2}\right)
\end{aligned} \tag{5.28}$$

und für die Restabbildungen, welche noch ausgeführt werden müssen

$$\Psi^{(5)} = \alpha x \quad \Psi^{(4)} = \alpha \prod_{i=1}^n x_i \quad \Psi^{(3)} = \alpha e^{\sum_{i=1}^n x_i} \quad \Psi^{(2)} = \alpha e^{\sum_{i=1}^n \frac{-x_i}{2\sigma^2}} \quad \Psi^{(1)} = \alpha e^{-\frac{\sum_i x_i^2}{2\sigma^2}}. \tag{5.29}$$

Damit folgt für den absoluten Fehler aufgrund der Fehlerfortpflanzungen bei der zweiten Variante

$$\begin{aligned}
\Delta y_{\Pi} &= Dg(\mathbf{x})\Delta\mathbf{x} - \frac{\alpha\epsilon_-}{\sigma^2} \exp(\cdot)\|\mathbf{x} - \mathbf{c}\|^2 - \frac{\alpha\epsilon_/}{2\sigma^2} \exp(\cdot)\|\mathbf{x} - \mathbf{c}\|^2 \\
&\quad - \frac{\alpha\epsilon_*}{2\sigma^2} \exp(\cdot)\|\mathbf{x} - \mathbf{c}\|^2 + \epsilon_e n \alpha \exp(\cdot) + \alpha\epsilon_* \exp(\cdot) + \alpha\epsilon_\alpha \exp(\cdot). \tag{5.30}
\end{aligned}$$

Somit können beide Algorithmen zur Berechnung einer gaußschen Basisfunktion miteinander verglichen werden, für welchen Algorithmus sich ein kleinerer absoluter Fehler aufgrund der Fehlerfortpflanzung bei den Berechnungen ergibt. Ein Vergleich von (5.26) und (5.30) ergibt

$$\Delta y_{\text{I}} \leq \Delta y_{\Pi} \tag{5.31}$$

$$\begin{aligned}
\Rightarrow & \left| -\frac{\alpha\epsilon_*}{2\sigma^2} \exp(\cdot)\|\mathbf{x} - \mathbf{c}\|^2 - \frac{\alpha\epsilon_/}{2\sigma^2} \exp(\cdot)\|\mathbf{x} - \mathbf{c}\|^2 + \alpha\epsilon_e \exp(\cdot) \right| \\
& \leq \left| -\frac{\alpha\epsilon_/}{2\sigma^2} \exp(\cdot)\|\mathbf{x} - \mathbf{c}\|^2 + \alpha\epsilon_* \exp(\cdot) + \alpha n \epsilon_e \exp(\cdot) \right|. \tag{5.32}
\end{aligned}$$

Für die relativen Fehler  $\epsilon_i, i \in \{e, +, -, *, /\}$  kann angenommen werden, dass diese kleiner als die Maschinengenauigkeit  $\mathbf{eps}$  sind, also gilt jeweils

$$|\epsilon_i| \leq \mathbf{eps} \quad \text{für } i \in \{e, +, -, *, /\}. \tag{5.33}$$

Hiermit folgt für (5.32) bei relativen Fehlern kleiner der Maschinengenauigkeit

$$\Delta y_{\text{I}} \leq \Delta y_{\text{II}} \quad (5.34)$$

$$\text{eps} \left( \frac{\|\mathbf{x} - \mathbf{c}\|^2}{2\sigma^2} + \frac{\|\mathbf{x} - \mathbf{c}\|^2}{2\sigma^2} + 1 \right) \leq \text{eps} \left( \frac{\|\mathbf{x} - \mathbf{c}\|^2}{2\sigma^2} + n + 1 \right) \quad (5.35)$$

$$\|\mathbf{x} - \mathbf{c}\|^2 \leq \underbrace{2n\sigma^2}_{r^2}. \quad (5.36)$$

Dadurch ist nach (5.36) der absolute Fehler bei einer Berechnung mit Algorithmus I (vgl. Abbildung 3.4, Abschnitt 3.2.1) innerhalb einer Hypersphäre mit dem Radius

$$r = \sigma\sqrt{2n} \quad (5.37)$$

kleiner als der Fehler durch Algorithmus II (vgl. Abbildung 3.5, Abschnitt 3.2.1). Unterscheiden sich die relativen Fehler der einzelnen Berechnungen, so verändert sich der Radius dieser Hypersphäre zu

$$r = \sigma \sqrt{\frac{2(|\epsilon_e|(n-1) + |\epsilon_*|)}{|\epsilon_+|}}. \quad (5.38)$$

Somit ist Algorithmus I in direkter Umgebung um das Zentrum  $\mathbf{c}$  Algorithmus II vorzuziehen, der allerdings für Werte weit entfernt des Zentrums geringere absolute Fehler produziert. Da während des Betriebes ein Wechsel<sup>3</sup> der Berechnungsvorschrift nicht möglich ist, ist die Berechnung nach Algorithmus I aus Abbildung 3.4 vorzuziehen, da die Basisfunktionen aufgrund ihrer lokalen Übertragungseigenschaft im Bereich des Zentrums betrieben werden.

Weiter lässt sich für die beiden Algorithmen jeweils noch eine Variante betrachten, indem anstatt erst zu quadrieren zunächst durch die Varianz  $\sqrt{2\sigma^2}$  der Basisfunktionen geteilt wird. Für  $\sigma > 1$  wird für die Speicherung dieses Wertes eine geringere Auflösung benötigt, da kleinere absolute Werte gespeichert werden müssen. Für den Fall  $\sigma < 1$  wird allerdings bei einer Speicherung der Varianzen eine höhere Auflösung gefordert, weil kleinere absolute Werte aufgelöst werden müssen, und somit eine größere Wortbreite für den Bruch benötigt wird.

Hinsichtlich des absoluten Fehlers, der sich aufgrund der Fehlerfortpflanzung ergibt, sind beide Varianten schlechter als die in Abbildung 3.4 und 3.5 dargestellten

<sup>3</sup> Ein Wechsel des Verfahrens ist beispielsweise bei einem *Field Programmable Gate Array* möglich.

Berechnungsvorschriften. Für den absoluten Fehler der Variante von Algorithmus I folgt

$$\begin{aligned} \Delta y_{Iv} = & Dg(\mathbf{x})\Delta\mathbf{x} - \frac{\alpha\epsilon_-}{\sigma^2} \exp(\cdot)\|\mathbf{x} - \mathbf{c}\|^2 - \frac{\alpha\epsilon_/}{\sigma^2} \exp(\cdot)\|\mathbf{x} - \mathbf{c}\|^2 \\ & - \frac{\alpha\epsilon_+}{2\sigma^2} \exp(\cdot)\|\mathbf{x} - \mathbf{c}\|^2 - \frac{\alpha\epsilon_*}{2\sigma^2} \exp(\cdot)\|\mathbf{x} - \mathbf{c}\|^2 + \alpha\epsilon_e \exp(\cdot) + \alpha\epsilon_\alpha \exp(\cdot) \end{aligned} \quad (5.39)$$

und für die Variante von Algorithmus II

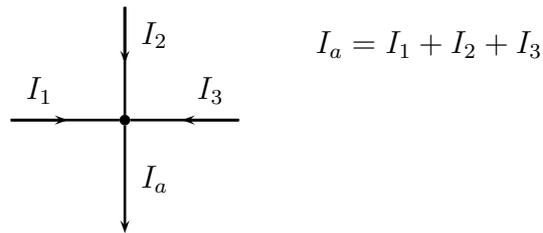
$$\begin{aligned} \Delta y_{IIv} = & Dg(\mathbf{x})\Delta\mathbf{x} - \frac{\alpha\epsilon_-}{\sigma^2} \exp(\cdot)\|\mathbf{x} - \mathbf{c}\|^2 - \frac{\alpha\epsilon_/}{\sigma^2} \exp(\cdot)\|\mathbf{x} - \mathbf{c}\|^2 \\ & - \frac{\alpha\epsilon_*}{2\sigma^2} \exp(\cdot)\|\mathbf{x} - \mathbf{c}\|^2 + \epsilon_e n \alpha \exp(\cdot) + \alpha\epsilon_* \exp(\cdot) + \alpha\epsilon_\alpha \exp(\cdot). \end{aligned} \quad (5.40)$$

Somit bleibt festzuhalten, dass Algorithmus I aus Abbildung 3.4 hinsichtlich des absoluten Fehlers die bessere Berechnung einer gaußschen Basisfunktion als durch Algorithmus II darstellt.

## 5.2 Entwurf neuronaler Systeme

In diesem Kapitel sollen die Eigenschaften eines Hardwareentwurfes für ein RBF-Netz und ein LCNN miteinander verglichen werden. Dafür sollen beide Algorithmen bzgl. ihrer Umsetzungsfähigkeit in analoge und digitale Hardware betrachtet werden. Algorithmus I aus Abbildung 3.4 benötigt gegenüber der Berechnungsvorschrift II aus Abbildung 3.5 weniger Fläche, da weniger arithmetische Einheiten benötigt werden. Tabelle 5.1 zeigt die Anzahl von benötigten Grundrechenarten für beide Berechnungsvorschriften für eine einzelne Basisfunktion  $g(\mathbf{x})$ . Dabei wird angenommen, dass Subtrahierer bzw. Dividierer durch Addierer und Multiplizierer ersetzt werden können. Es zeigt sich in Tabelle 5.1, dass die Anzahl von Recheneinheiten für Algorithmus II deutlich größer ist, was zu einem erhöhten Flächenbedarf führt. Allerdings benötigt ein Addierer innerhalb der Berechnungsvorschrift I insgesamt  $n$  Eingänge, die zur Berechnung der euklidischen Distanz benötigt werden. In digitaler Hardware kann diese Operation durch einen Baum von Addierern mit zwei Eingängen durchgeführt werden, wozu maximal  $n - 1$  Elemente benötigt werden. Somit ergibt sich hier eine Erhöhung der Anzahl der Addierer auf  $2n - 1$ . In analoger Schaltungstechnik jedoch kann mit Hilfe der Repräsentation durch Ströme diese Summation sehr einfach durch einen einzelnen Addierer durchgeführt werden, indem innerhalb eines Knotens die Ströme addiert werden (vgl. Abbildung 5.1).

Weiter kann auch noch eine Unterscheidung zwischen dem Flächenbedarf der einzelnen Recheneinheiten (Addition, Multiplikation und  $\exp$ -Funktion) vorgenommen werden, wobei eine Multiplikation i.A. mehr Fläche als eine Addition benötigt [BK80, Rab96].



**Abbildung 5.1:** Addition mittels Repräsentation durch Ströme

**Tabelle 5.1:** Vergleich der benötigten Anzahl von Grundrechenelementen für beide Berechnungsvorschriften

Algorithmus	Addierer	Multiplizierer	exp-Funktion
I	$n + 1/2n - 1$	$n + 2$	1
II	$n$	$3n$	$n$

Dabei ist noch nicht betrachtet worden, dass auch die exp-Funktion durch die Grundrechenarten approximiert werden muss, da es keine direkte Umsetzung in Hardware gibt<sup>4</sup>.

### 5.2.1 Digitaler Entwurf

Bei einem digitalen Entwurf kann das System in einer Hardwarebeschreibungssprache entworfen werden, so dass nachträglich die verwendete Auflösung, die Größe des Netzes und die Zieltechnologie mit kleinerem Aufwand gegenüber einer analogen Implementierung geändert werden können. Dadurch steigt die Wiederverwendbarkeit des Entwurfes, und zusätzlich kann die Entwicklungszeit verringert werden.

Bei der Umsetzung eines neuronalen Netzes in Hardware tritt der Effekt begrenzter Genauigkeit und Rauschen auf (siehe Kapitel 4). Weiter muss zur Funktionsapproximation die Übertragungsfunktion eines Neurons entsprechend umgesetzt werden. Da hier die neuronalen Netze als eigenständiges System mit elementaren Recheneinheiten (Neuronen) aufgefasst werden, wird aus diesem Grund die Realisierung der Übertragungsfunktion mit Hilfe einer Speicherung der Zuordnung nicht betrachtet. Ein Neuron würde in diesem Fall ausschließlich ein Gedächtnis besitzen, jedoch keine eigenständigen Operationen ausführen können. Auch eine Umsetzung durch geeignete boolesche Logikfunktionen wird nicht weiter betrachtet, da keine andere

<sup>4</sup> Zwar ist diese in analoger Schaltungstechnik umsetzbar, jedoch ist auch hier die Anzahl der Elemente der exp-Funktion für Algorithmus II deutlich höher.

Funktionalität bereitgestellt werden kann. Beide Verfahren entsprechen nicht dem biologischen Vorgehen, bei dem ein Neuron neben Speicher auch eine Grundfunktionalität besitzt. Die Genauigkeit dieser beiden Verfahren, der Umsetzung durch Speicher oder boolesche Funktionen, hängt von der Auflösung ab und wird durch den maximalen Quantisierungsfehler bestimmt. Für diesen folgt

$$Q = \frac{1}{2^{l+1}}, \quad (5.41)$$

wobei  $l$  die Auflösung des Bruchanteils bezeichnet.

### 5.2.1.1 Sigmoide Aktivierungsfunktion

Für das LCNN werden sigmoide Aktivierungsfunktionen benötigt. Dabei ist es für den mathematischen Existenzbeweis, dass jedes MLP eine Funktion beliebig genau approximieren kann, z.B. nur notwendig, eine nach oben und unten beschränkte und monoton steigende Funktion<sup>5</sup> als Übertragungsfunktion zu verwenden [HSW89]. So können verschiedene Übertragungsfunktionen verwendet werden, wobei die Logistikkfunktion eine sehr weite Verbreitung besitzt. Die Parameter des Netzes können dann mit dem Trainingsverfahren, z.B. mit dem (natürlichen) Gradientenabstieg, bestimmt werden.

In [MB99b] wurde eine vereinfachte Funktion anstelle der klassischen Sigmoidfunktion vorgeschlagen. Gegenüber der Logistikkfunktion ergibt sich ein maximaler absoluter Fehler von  $\Delta \leq 0,08$ , wobei der relative Fehler divergiert<sup>6</sup>. Ebenso wurde in [BPVL94] eine Realisierung des Sigmoids mit wenigen grundlegenden Rechenelementen in digitaler Schaltungstechnik vorgestellt, wobei diese auf der stückweise linearen Approximation der Aktivierungsfunktion basiert. Der absolute Fehler ist bei dieser Form durch 0,019 beschränkt, und der relative Fehler beträgt weniger als 2%, wie sich leicht nachprüfen lässt.

Der Vorteil der abschnittswiseen Approximation der Übertragungsfunktion liegt in dem kleineren Fehler der Approximation, wobei der Fehler für die Annäherung des Sigmoids die Effekte aufgrund der Quantisierung schon beinhaltet. Beide Verfahren sind annähernd gleich komplex und benötigen beide zwei Additionen, eine Multiplikation bzw. eine Division sowie einige Schiebe-Operationen. Aus diesen Gründen ist die approximierte Übertragungsfunktion aus [BPVL94] der vereinfachten Funktion aus [MB99b] vorzuziehen.

Neben diesen beiden Verfahren besteht die Möglichkeit, die Aktivierungsfunktion eines Neurons mit Hilfe der Taylorreihe oder Potenzreihen zu approximieren. Die

<sup>5</sup> In der Literatur finden sich viele verschiedene Existenzbeweise, die unterschiedliche Voraussetzungen an die Übertragungsfunktion stellen.

<sup>6</sup> Bei einem integrierten Lernverfahren kann der Fehler durch Adaption der Parameter minimiert werden. Wenn dieses jedoch fehlt, spielt der Fehler zwischen der eingesetzten und in der Theorie verwendeten Übertragungsfunktion eine Rolle.

Genauigkeit der Approximation richtet sich dann nach dem verwendeten Grad des Polynoms. Auf dieses Verfahren wird später in Abschnitt 5.2.1.3 näher eingegangen.

### 5.2.1.2 Gaußfunktion

In [MB99b] wurde ebenfalls eine vereinfachte Basisfunktion vorgestellt, die neben der Ähnlichkeit zur Gaußfunktion ebenfalls radialsymmetrisch um ihr Zentrum verläuft. Diese Annäherung basiert auf der Taylorreihe der Funktion  $\exp(x)$  mit Entwicklungspunkt  $x_0 = 0$ , welche nach dem linearen Term abgebrochen wird. Als Basisfunktion wird dann

$$g(x) = \frac{1}{1+x} \quad (5.42)$$

verwendet, wobei die Eingänge quadriert werden müssen, um eine Gaußfunktion zu erhalten. Allerdings wurden in [MB99b] die Auswirkungen auf den relativen und absoluten Fehler nicht betrachtet. Es kann leicht nachgewiesen werden, dass der relative Fehler bei dieser Abschätzung divergiert und der absolute Fehler eine obere Schranke von  $\Delta \leq 0,21$  besitzt. Der absolute Fehler wurde hierfür mittels des Newtonverfahrens ermittelt, da der absolute Fehler für  $x \rightarrow 0$  bzw.  $x \rightarrow \infty$  gegen Null konvergiert.

Der Nachteil dieser Implementierung in Hardware ist die geforderte Division, um eine gaußähnliche Funktion zu erhalten. Eine alternative Möglichkeit zur Approximation der Gaußfunktion ist die Approximation der Funktion  $\exp(-x)$  durch

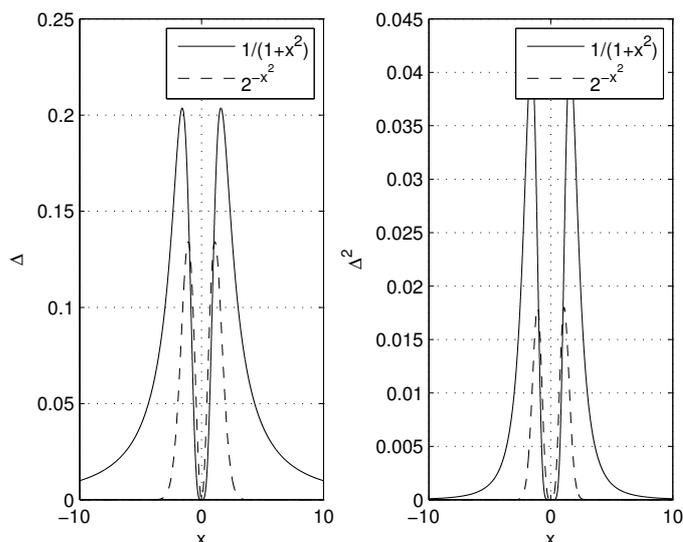
$$e^{-x} \approx 2^{-x} \quad \text{für } x \in \mathbb{R}^+. \quad (5.43)$$

Der Vorteil bei dieser Approximation für positive Eingänge ist der kleinere absolute Fehler<sup>7</sup>. Auch hier konvergiert der Fehler für  $x \rightarrow 0$  bzw.  $x \rightarrow \infty$  gegen Null. Das Maximum kann mit Hilfe des Newtonverfahrens zu 0,135 bestimmt werden. Der relative Fehler divergiert bei diesem Verfahren ebenfalls.

Um beide Verfahren miteinander zu vergleichen, soll der absolute Fehler, der sich bei beiden Approximationsarten ergibt, näher betrachtet werden. Dabei seien die Daten des Einganges symmetrisch um den Ursprung auf Basis einer Gleichverteilung mit unendlicher Breite angeordnet. Beide Approximationsarten besitzen den gleichen Erwartungswert für den absoluten Fehler  $\Delta$ , wobei das Verfahren über die Potenzfunktion eine kleinere Varianz aufweist. Für eine endliche Gleichverteilung der Eingangsdaten nimmt die Approximation über die Taylorreihe ebenfalls einen größeren Erwartungswert des Fehlers ein.

In Abbildung 5.2 sind der absolute und der quadrierte, absolute Fehler für beide Approximationsarten über den Eingangsbereich der Gaußfunktion dargestellt. Man kann deutlich erkennen, dass die Approximation über die Taylorreihe einen

<sup>7</sup> Für negative Eingabewerte divergiert der absolute Fehler.



**Abbildung 5.2:** Absoluter und quadrierter, absoluter Fehler bei Approximation der Gaußfunktion

höheren Fehler erzeugt und so der Näherung durch die Potenzfunktion unterlegen ist, obwohl sich gleiche Erwartungswerte und Varianzen bei einer unendlich breiten Gleichverteilung ergeben.

Hinsichtlich ihrer Umsetzbarkeit in digitaler Schaltungstechnik sind beide Verfahren annähernd gleich komplex. Beide Verfahren nähern die Exponentialfunktion, so dass für die Gaußfunktion die Eingaben zuerst quadriert werden müssen. Für die Verarbeitung der Daten werden dann für die Taylorreihe eine Addition und eine Multiplikation benötigt.

Die Umsetzung der Funktion  $g(x) = 2^x$  kann in digitaler Schaltungstechnik durch eine stückweise lineare Approximation umgesetzt werden [BPVL94]. Für diese Realisierung werden eine Schiebe-Operation und eine Multiplikation benötigt. Die (im Algorithmus verwendete) Addition kann über eine geeignete Zusammenfassung der Signale geschehen. Die stückweise, lineare Approximation erzeugt ebenfalls einen Fehler, wobei dieser zusätzlich noch von der verwendeten Auflösung abhängig ist. Bei einer Rasterung von 4 bit für die Bruchdarstellung ergibt sich ein maximaler Fehler von  $\Delta \leq 0,1610$ . Dieser wurde in der Simulation der (synthetisierbaren) VHDL-Beschreibung ermittelt und liegt nur unwesentlich höher als der maximale absolute Fehler bei Verwendung reeller Zahlen.

Da beide Verfahren in digitaler Schaltungstechnik ähnliche Komplexität aufweisen, ist die Approximation über die Potenzfunktion, bei der sich bessere Approximationseigenschaften ergeben, der Taylorreihe vorzuziehen. Da neben der Approximation der Gaußfunktion durch die Potenzfunktion auch eine sigmoide Übertragungsfunktion approximiert werden kann, wird für die Erweiterung der in Abschnitt 5.2.1.3

vorgestellten rekonfigurierbaren Architektur dieses Verfahren zur Umsetzung der Gaußfunktion verwendet. Neben diesen beiden Verfahren kann die Taylorreihe dazu benutzt werden, die Gaußfunktion zu approximieren. Dieses Vorgehen wird in Abschnitt 5.2.1.3 näher untersucht.

### 5.2.1.3 Rekonfigurierbare Struktur

Um die Wiederverwendbarkeit eines Systems zur Abbildung neuronaler Verfahren zu erhöhen, wurde eine einfache, rekonfigurierbare, neuronale Struktur (*Simple Reconfigurable Neural Structure*, SIRENS) entwickelt [EKR06]. Zur generellen Approximation einer Aktivierungsfunktion eines Neurons kann die Taylorreihe verwendet werden. Der Vorteil dieses Vorgehens ist, dass die Eigenschaften der Approximation anhand mehrerer Parameter bestimmt werden und so dem aktuellen Problem angepasst werden können. Der Entwicklungspunkt und die Anzahl der verwendeten Terme bestimmen die Genauigkeit des Vorgehens.

Im Folgenden bezeichne  $n$  die Anzahl der Terme der Taylorreihe. Nach dem Abbruch des  $n$ -ten Terms ergibt sich für das Restglied und somit für den Fehler der Approximation bei einem Entwicklungspunkt  $x_0$  [BSM05]

$$R_n(x) = \frac{g^{(n+1)}(\xi)}{(n+1)!} (x - x_0)^{n+1} \quad \text{mit } \xi \in [x; x_0]. \quad (5.44)$$

Für die Aktivierungsfunktion  $g(x) = \exp(-x)$  und die Gaußfunktion kann sehr einfach die Konvergenz der Reihe gezeigt werden. Stellvertretend folgt für die Gaußfunktion mit  $x_0 = 0$

$$\lim_{n \rightarrow \infty} R_n(x) = \lim_{n \rightarrow \infty} \frac{(-2)^{n+1} x^{2n+2}}{(n+1)!} e^{-\xi^2} = 0 \quad \text{mit } 0 \leq \xi \leq x \quad (5.45)$$

aufgrund der Monotonie der exp-Funktion. Da die Gaußfunktion und die Funktion  $g(x) = \exp(-x)$  ineinander durch Substitution  $a = x^2$  überführbar sind, wird im Folgenden nur noch die Funktion  $g(x)$  betrachtet.

Für den Approximationsfehler kann das Restglied der Taylorreihe abgeschätzt werden. Dabei folgt für das Restglied der Funktion  $g(x) = \exp(-x)$  mit  $x \in [0; b]$

$$|R_n(x)| = \left| \frac{(-x)^{n+1} e^{-\xi}}{(n+1)!} \right| \leq \left| \frac{x^{n+1}}{(n+1)!} \right| \quad \text{mit } 0 \leq \xi \leq b. \quad (5.46)$$

Diese Abschätzung erlaubt, die gewünschte Genauigkeit bei der Approximation mit Hilfe des Polynomgrades zu ermitteln. Indem das Restglied abgeschätzt wurde, lässt sich der Grad des Polynoms bestimmen, um eine gewünschte Genauigkeit in dem vorgesehenen Intervall zu erreichen. Für die Breite des Intervalls gilt dabei, je kleiner

das gewünschte Approximationsintervall gewählt wird, desto niedriger ist die Anzahl von Termen im Polynom. Aufgrund der strengen Monotonie der Gaußfunktion und der Funktion  $g(x)$  müssen für eine hohe Approximationsgüte eine große Anzahl von Termen in der Taylorreihe berücksichtigt werden. Um einen maximalen Fehler von 0,1 bei der Taylorreihe der Funktion  $g(x)$  im Intervall  $[0; 4]$  zu tolerieren, muss nach der Abschätzung ein Polynom 10. Grades verwendet werden. Der tatsächliche Fehler ergibt sich zu  $\Delta \leq 0,0967$ .

Die Approximationseigenschaften können verbessert werden, wenn die Symmetrie der Gaußfunktion ausgenutzt wird. Da diese eine gerade Funktion ist, ist es ausreichend, die Funktion in der positiven oder negativen Halbebene anzunähern. Dies kann ebenfalls durch die Taylorreihe geschehen bzw. durch die Verwendung eines Polynoms  $p(x)$  n-ten Grades

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0. \quad (5.47)$$

Durch die Koeffizienten kann die Art der Approximation eingestellt werden. Für diese können ebenfalls die Koeffizienten der Taylorreihe verwendet werden, so dass dieser Ansatz keine Einschränkung gegenüber der Approximation mittels Taylorreihe darstellt. Durch geschickte Wahl der Koeffizienten kann so der Fehler gegenüber einem Taylorpolynom verringert werden. Der Fehler bei einer Polynominterpolation lässt sich durch [Pla04]

$$f(x) - p(x) = \frac{w(x) f^{(n+1)}(\xi)}{(n+1)!} \quad (5.48)$$

mit  $w(x) = (x - x_0) \cdot \dots \cdot (x - x_n)$ , den Stützstellen der Interpolation, darstellen. Bei der Approximation mittels eines Polynoms ergeben sich andere Stützstellen als bei einer Interpolation der Trainingsdaten. Somit gilt i.A. [Pla04]

$$w(x) \leq x^{n+1}, \quad (5.49)$$

so dass ein kleinerer Fehler als bei der Taylorreihe erreicht wird, je nachdem wie die Stützstellen gewählt werden.

Die Koeffizienten des Polynoms können durch ein Optimierungsproblem gelöst werden. Hierbei wird der Abstand zwischen dem Polynom und der Aktivierungsfunktion minimiert, was mathematisch durch

$$\text{minimiere} \quad \max_{x \in [a; b]} \left( w(x) \cdot |g(x) - p(x)| \right) \quad (5.50)$$

ausgedrückt werden kann, wobei  $w(x)$  eine Gewichtsfunktion darstellt, mit der das Ziel der Optimierung vorgegeben werden kann. Das Optimierungsproblem aus (5.50) ist auch als *Minimax-Problem* [HUL93] bekannt.

**Tabelle 5.2:** Absoluter und relativer Fehler der Approximation der Aktivierungsfunktion mittels eines Polynoms und der Taylorreihe. Die Koeffizienten des Polynoms minimieren den relativen bzw. den absoluten Fehler.

Funk.	Meth.	Grad des Polynoms					
		2	3	4	5	6	7
$e^{-x}$	abs.	0,1634	0,0709	0,0267	0,0088	0,0025	0,0007
	Taylor	32,4999	89,0001	184,3749	307,7001	430,4124	518,5894
	rel.	0,9458	0,7582	0,4420	0,1896	0,0659	0,0196
$e^{-x^2}$	abs.	0,0777	0,0618	0,0208	0,0048	0,0037	0,0085
	Taylor	8,0001	-	32,4999	-	89,0001	-
	rel.	0,8421	0,3560	0,2438	0,1264	0,0197	0,0019

Durch die Wahl des Gewichtspolynoms lässt sich die Minimierung hinsichtlich der Art des Fehlers einstellen. Für  $w(x) = 1$  wird demnach der absolute Fehler zwischen der Aktivierungsfunktion und dem Polynom minimiert, während für  $w(x) = 1/g(x)$  der relative Fehler zwischen beiden Funktionen die Zielfunktion der Optimierung darstellt.

Tabelle 5.2 zeigt die Approximationsgenauigkeit für die exp-Funktion und die Gaußfunktion. Dafür wurde zum einen die Taylorreihe verwendet und zum anderen die Polynomapproximation, wobei die Koeffizienten durch die Lösung des Optimierungsproblems aus (5.50) mit dem absoluten und dem relativen Fehler<sup>8</sup> als Zielfunktion verwendet wurden. Als Eingangsbereich wurde für die Funktion  $g(x) = \exp(-x)$  das Intervall  $[0; 9]$  gewählt und für die Gaußfunktion  $[0; 3]$ . So decken beide Funktionen denselben Eingangsbereich ab. Ein Vorteil der Gaußfunktion ist dabei, dass keine zusätzliche Quadrierung der Eingänge erfolgen muss, sondern diese direkt verarbeitet werden können. Die Tabelle zeigt nur die Fehler, die aufgrund der Approximation entstehen. Zusätzlich müssen die Quantisierungsfehler noch berücksichtigt werden.

Wie man an der Tabelle 5.2 erkennen kann, ist die Polynomapproximation mit den Koeffizienten des Optimierungsproblems der Taylorreihe deutlich in der Approximationsgüte überlegen. Es lassen sich für Polynome mit einem geringen Grad kleine absolute und relative Fehler erreichen. Ein guter Kompromiss für diese beiden Funktionen ist ein Polynom mit einem Grad  $n \in \{5; 6; 7\}$ . Hier ergibt sich gegenüber einem Polynom vierter Ordnung eine deutliche Verbesserung der Approximationseigenschaften. Wie man an der Tabelle und an Gleichung (5.48) erkennen kann, verringert sich der Fehler mit steigendem Grad des Polynoms. Für eine digitale Implementierung ist dies nur bedingt geeignet, da sehr große/kleine Zwischenergebnisse gespeichert werden müssen und für eine hohe Anzahl von Termen ein großer

<sup>8</sup> Die Werte für den relativen Fehler in Tab. 5.2 entsprechen den Prozentangaben.

**Tabelle 5.3:** Anzahl von Operationen der Polynomdarstellung

	Polynom	Horner-Schema
Additionen	$n$	$n$
Multiplikationen	$\frac{n(n+1)}{2}$	$n$

Flächenbedarf besteht. Dies spricht gegen den Einsatz von Polynomen hohen Grades.

Die Approximation der Aktivierungsfunktion mit Hilfe eines Polynoms stellt somit eine gute Lösung zur Näherung unterschiedlicher Übertragungsfunktionen dar. Mit Hilfe der Koeffizienten können verschiedene Neuronenmodelle angenähert werden, wobei der Grad des Polynoms die Genauigkeit bestimmt. Das Schema aus (5.47) kann in seiner Effizienz verbessert werden, in dem das Horner-Schema [BSM05] angewendet wird. So ergibt sich für die Auswertung

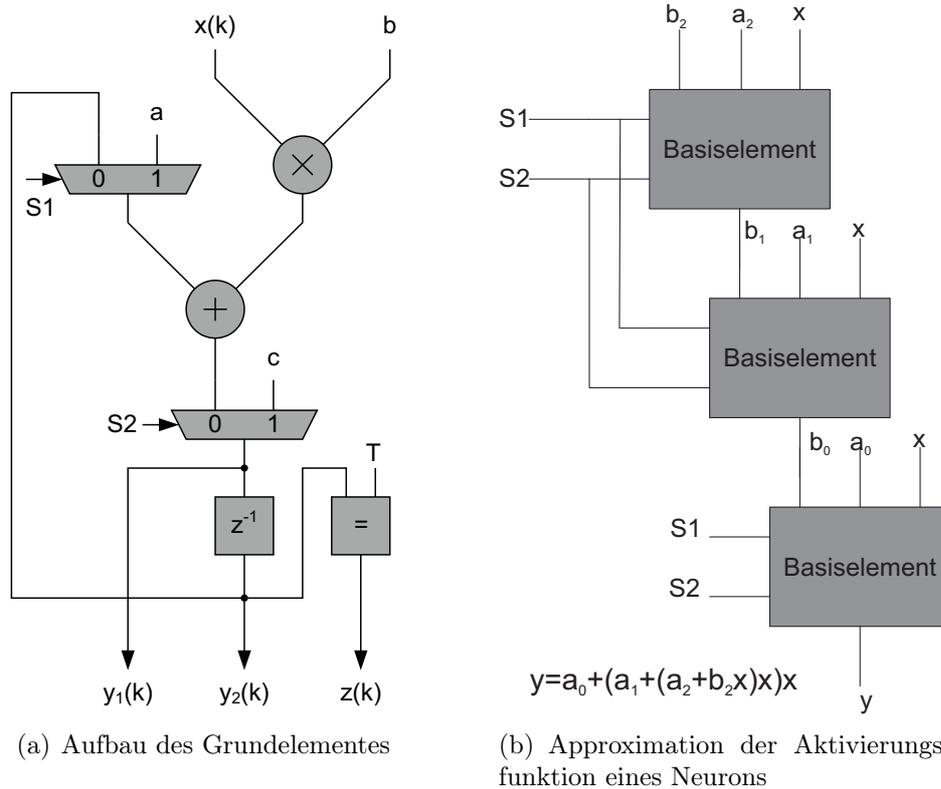
$$p(x) = a_0 + (a_1 + (a_2 + \dots + a_n x)x)x. \quad (5.51)$$

Zum einen wird dadurch die numerische Stabilität verbessert [Pla04], und zum anderen sinkt der Berechnungsaufwand für das Polynom. Tabelle 5.3 zeigt, dass die Komplexität der direkten Auswertung des Polynoms besonders bei der Multiplikation höher als beim Horner-Schema ist. Idealerweise werden Multiplikationen eingespart, welche bei digitalen Rechenelementen gegenüber einer Addition immer ein größeres Produkt aus Rechenzeit und Fläche aufweisen [BK80]. So kann der Aufwand des Flächenbedarfs und der Rechenzeit durch das Horner-Schema deutlich reduziert werden. Ein weiterer Vorteil des Horner-Schemas ist die einfache Berechnung der Ableitung an derselben Stelle wie die Auswertung des Polynoms. So kann durch eine wiederholte Anwendung der Wert der Ableitung ohne zusätzlichen Rechenaufwand bestimmt werden.

Mit Hilfe des Horner-Schemas entsteht somit ein Basiselement, mit welchem durch serielle Verschaltung der Funktionswert des Polynoms und durch wiederholte Anwendung ebenfalls der Wert der Ableitung bestimmt wird. Mit der Anzahl von Basiselementen dieser seriellen Verschaltung wird der Grad des Polynoms bestimmt. Für die Übertragungsfunktion des Elementes ergibt sich eine Multiplizier-Addier-Einheit (MAC-Einheit)

$$y(k) = a + b \cdot x(k). \quad (5.52)$$

Um eine hohe Genauigkeit durch die Polynomapproximation zu erreichen, muss ein hoher Grad verwendet werden. Ein Nachteil der seriellen Verschaltung der Grundelemente (MAC-Einheiten) ist die hohe Latenz aufgrund vieler Multiplikationen und Additionen. Somit ist es erforderlich, die Verarbeitung der Daten durch ei-



**Abbildung 5.3:** Aufbau der rekonfigurierbaren Struktur SIRENS

ne Pipeline-Verarbeitung zu verbessern und die Ergebnisse jeder einzelnen MAC-Einheit zwischenspeichern. Um zusätzlich die Werte der Ableitungen automatisch berechnen zu können, wird ebenfalls der Ausgang auf einen der Eingänge rückgekoppelt, um so eine wiederholte Anwendung des Horner-Schemas während des Betriebs<sup>9</sup> zu ermöglichen.

Bedingt durch das Horner-Schema und die Rückkopplung innerhalb eines Basiselementes (vgl. Abbildung 5.3(a)) kann der Wert der Ableitung an einer Stelle der Aktivierungsfunktion berechnet werden. So kann durch die rekonfigurierbare SIRENS-Struktur nicht nur ein Modell simuliert werden, sondern ebenfalls einfache Lernverfahren innerhalb des Systems umgesetzt werden, welche auf Gradientenverfahren basieren. Als populären Vertreter dieser Art von Trainingsmethoden sei das Backpropagation-Verfahren [Hay99, RHW86b] genannt. So kann ein selbst-optimierendes System entwickelt werden, da entsprechende Optimierungsfähigkeiten im System vorhanden sind.

<sup>9</sup> Für die Berechnung der Ableitung wird der aktuelle Funktionswert benötigt. So muss der aktuell zwischenspeicherte Wert zurückgekoppelt werden.

Die neue Übertragungsfunktion des Basiselementes ergibt sich sodann

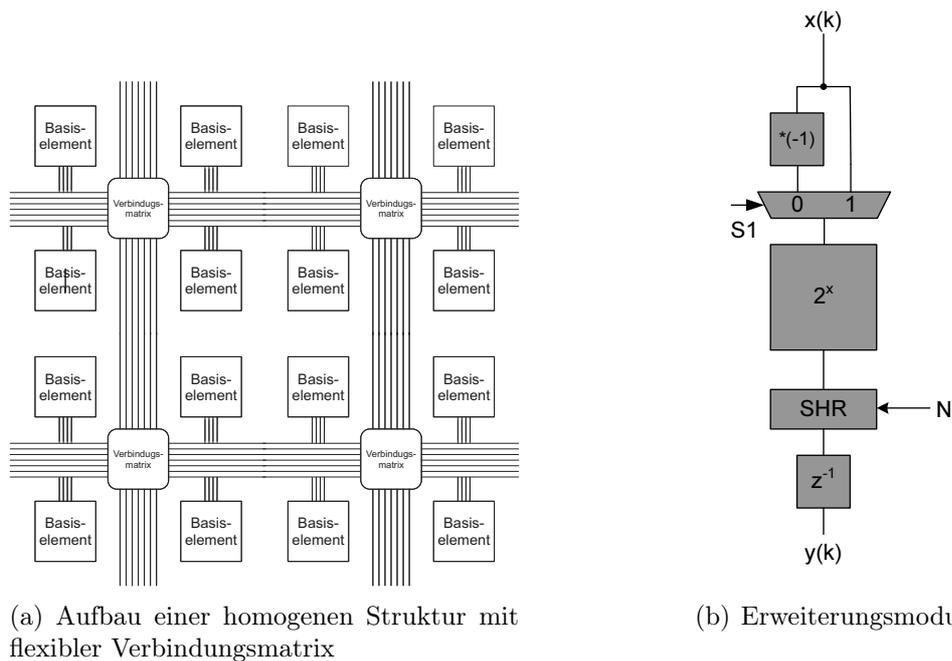
$$y_2(k+1) = \begin{cases} c & \text{falls } S2 = 1 \\ a + bx(k) & \text{falls } S1 = 0 \wedge S2 = 0 \\ y_2(k) + bx(k) & \text{falls } S1 = 1 \wedge S2 = 0 \end{cases}, \quad (5.53)$$

wobei die Steuersignale  $S1$  und  $S2$  der Multiplexer die aktuelle Operation des Grundelementes beeinflussen. Abbildung 5.3(a) zeigt den Aufbau des Basiselementes. Der zusätzliche Komparator ist aus zwei Gründen eingefügt worden. Zum einen lassen sich mit Hilfe der Rückkopplung auch dynamische Modelle von Neuronen durch diese Struktur darstellen, wie später noch genauer gezeigt wird. Durch den zusätzlichen Komparator kann der Aufwand für eine solche Darstellung reduziert werden. Zum anderen dient der Komparator zur verbesserten Darstellung von häufig verwendeten Modellen. Die Polynomdarstellung einer Funktion, welche Unstetigkeitsstellen beinhaltet, ist sehr ungenau und benötigt zur adäquaten Darstellung einen sehr hohen Polynomgrad. Durch den Komparator kann dieser Effekt verbessert werden, und so lässt sich ein MLP-Netz mit einer Heaviside-Funktion als Übertragungsfunktion ebenfalls durch ein Basiselement darstellen (McCulloch-Pitts-Modell).

Durch die serielle Verschaltung der Basiselemente und der Bestimmung der Koeffizienten mit Hilfe von (5.50) kann jede Aktivierungsfunktion approximiert werden, da die Menge von Polynomen die Dichteigenschaft im Vektorraum der reellwertigen Funktionen besitzt [Sau01]. Abbildung 5.3(b) zeigt die Approximation der Übertragungsfunktion eines Neurons mit Hilfe der seriellen Verschaltung dreier Basiselemente. Somit stellen diese Elemente ein Neuron in dem Netz dar. Neben der seriellen Berechnung der Übertragungsfunktion kann die Aktivierungsfunktion auch sequenziell durch ein einzelnes Basiselement berechnet werden, wenn das Ergebnis der aktuellen Berechnung wieder auf den Eingang  $x(k)$  zurückgekoppelt wird und die Koeffizienten dynamisch verändert werden.

Eine schachbrettartige Anordnung dieser Basiselemente ermöglicht die Darstellung verschiedenartiger Netze durch diese Struktur. Es entsteht eine homogene Struktur, welche durch die aktuelle Verschaltung und die Wahl der Koeffizienten an die derzeitigen Anforderungen des gegenwärtigen Modells angepasst werden kann. Um eine solche flexible Verschaltung zu ermöglichen, wird eine Verbindungsstruktur zwischen den einzelnen Basiselementen benötigt. Abbildung 5.4(a) zeigt die entstehende Struktur, die einem *Field Programmable Gate Array* (FPGA) ähnelt, wobei die einzelnen Einheiten die Basiselemente aus (5.52) darstellen. Durch diese konfigurierbaren Verbindungen kann ein einzelnes Basiselement in serieller Arbeitsweise die Übertragungsfunktion eines Neurons annähern.

Mit Hilfe dieser Verbindungsstruktur kann der Typ des Netzes verändert werden. Neben der Approximation verschiedener Aktivierungsfunktionen kann die Art des Modells ebenfalls verändert werden. Durch die Rückkopplung innerhalb der einzel-



**Abbildung 5.4:** Erweiterung der rekonfigurierbaren Struktur zu einer homogenen bzw. heterogenen Struktur

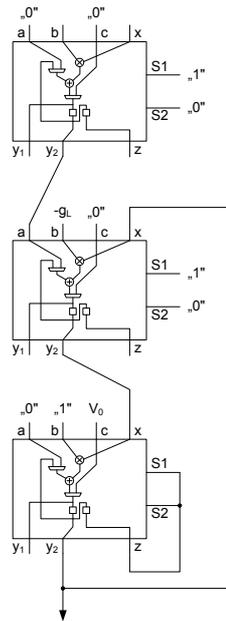
nen Basiselemente können auch dynamische Modelle [Arb02, MB99a] von Neuronen dargestellt werden.

Ebenfalls kann das bekannte *Integrate-and-Fire-Neuron*<sup>10</sup> durch die Struktur mit Hilfe von drei Basiselementen abgebildet werden. Bei diesem Neuronenmodell werden gewichtete Impulse, die von einem Neuron ausgesendet werden, summiert, und sobald ein definierter Schwellenwert überschritten wird, sendet das Neuron einen Impuls aus. Ein solches Neuron ist in Abbildung 5.5 dargestellt. Dabei wird das Aktivierungspotential des Neurons in dem letzten Basiselement gespeichert und wird auf das zweite Element negativ gewichtet  $\hat{g}_L$  zurückgekoppelt. Der für dieses Modell notwendige Integrator wird durch die Rückkopplung des letzten Elements auf sich selbst realisiert, wobei der Eingang  $\tilde{I}$  des Neurons durch den Eingang  $x(k)$  des ersten Elementes abgebildet wird.

Mit diesem dreistufigen System kann eine diskrete Integration durchgeführt werden, so dass die Differentialgleichungen des Neuronenmodells mit dem Signalflussgraphen des umgesetzten Systems verglichen werden können:

$$\begin{aligned} y(k) - y(k-1) &= \hat{b}x(k-1) - \hat{g}_L y(k-1) \\ \frac{dV}{dt} &= \frac{\tilde{I}}{C} - \frac{g_L}{C} V. \end{aligned} \quad (5.54)$$

<sup>10</sup>Für die genaue Beschreibung dieses Modells sei auf die Literatur [Arb02, Hay99] verwiesen.



**Abbildung 5.5:** Abbildung eines dynamischen Neurons durch SIRENS

Wird die Schrittbreite der Integration mitberücksichtigt, so können die Parameter des Neuronenmodells bestimmt werden,  $\hat{b} = \tilde{I}/(C \cdot \text{steps}/s)$ ,  $\hat{g}_L = g_L/(C \cdot \text{steps}/s)$ .

Durch die Rückkopplung und die konfigurierbare Verbindungsstruktur können ebenfalls rückgekoppelte, neuronale Netze auf dem System abgebildet werden. Bei diesen rückgekoppelten Netzen besitzt jedes Neuron der versteckten Ebenen eine Rückkopplung des Ausgangs oder eines anderen Neurons derselben Ebene auf seinen Eingang.

Ebenfalls lassen sich durch die SIRENS-Struktur Netzwerke höherer Ordnung und Netze mit Produkt-Einheiten realisieren und so die Vapnik-Chervonenkis-Dimension (VC-Dimension) [Vap95b] kontrollieren. Durch Nutzung beider Eingänge des Multiplizierers aus Abbildung 5.3(a) durch dasselbe Eingangssignal  $x$  kann ein Monom höherer Ordnung erzeugt und ein Netzwerk mit Produkt-Einheiten aufgebaut werden. Auch andere ähnliche Modelle, wie z.B. das Sigma-Pi-Netzwerk [RHM86], können so umgesetzt und damit die Kapazität der Netze kontrolliert werden. Aufgrund der begrenzten Anzahl von Rechenelementen existiert ein maximaler Grad des Polynoms. So ist die VC-Dimension des Netzes  $\mathcal{N}$  durch  $VCdim(\mathcal{N}) = q + 1$  begrenzt [Vap98], wobei  $q$  den maximalen Grad des Polynoms darstellt. Durch Darstellung eines Monoms besitzt jedes Basiselement die VC-Dimension  $VCdim(\mathcal{N}) = 2$  und ein Netz, bestehend aus mehreren solcher Einheiten, besitzt eine obere Grenze der VC-Dimension von  $VCdim(\mathcal{N}) \leq \mathcal{O}((Wk)^2 + 8Wk \log_2(13Wk))$  [Sch02], wobei  $W$  die Anzahl von programmierbaren Gewichten und  $k$  die Anzahl von Produkt-Einheiten bezeichnet. Somit kann das System an die aktuellen Anforderungen angepasst und die Kapazität des Netzes dynamisch verändert werden.

Durch die Erweiterung um das Modul aus Abbildung 5.4(b) entsteht eine inhomogene Struktur, wobei diese speziellen Elemente in geringerer Anzahl benötigt werden, da diese die Approximation populärer Neuronenmodelle verbessern. Wie in Kapitel 5.2.1.1 und 5.2.1.2 vorgestellt, kann mit diesem Modul die sigmoide Übertragungsfunktion sowie die Funktion  $g(x) = \exp(-x)$  mit geringem Flächenbedarf und hoher Approximationsgüte approximiert werden. Dafür muss lediglich eine Unterscheidung des Eingabebereiches durchgeführt und bei Bedarf der Eingang mit  $(-1)$  multipliziert<sup>11</sup> werden. Die Funktionalität dieser Einheit kann durch

$$y(k+1) = \begin{cases} \frac{2^{-x(k)}}{2^N} & \text{für } S1 = 0 \\ \frac{2^{x(k)}}{2^N} & \text{für } S1 = 1 \end{cases} \quad (5.55)$$

beschrieben werden. So können die Logistik- und die Gaußfunktion aus einer Kombination eines Grundelementes nach (5.53) und der Erweiterung aus Abbildung 5.4(b) mit einer geringeren Anzahl von Elementen angenähert werden.

Nach Abschnitt 5.2.1.2 kann die Funktion  $g(x) = \exp(-x)$  für positive  $x$  durch die Funktion  $2^{-x}$  approximiert werden, wobei ein absoluter Fehler kleiner als 0,135 entsteht. Um eine vergleichbare Auflösung mit Hilfe der Polynomapproximation zu erhalten, werden nach Tabelle 5.2 mindestens zwei Basiselemente benötigt. Für die gleiche Approximationsgüte wird weniger Fläche benötigt, da zum einen weniger Elemente benötigt werden, und zum anderen die Fläche des zusätzlichen Moduls wesentlich geringer als die eines Basiselementes ist (siehe hierfür die Ergebnisse aus Abschnitt 5.3.2). Die Berechnung der Potenzfunktion zur Basis zwei kann mit wenigen, elementaren Rechenelementen durch das Verfahren in [BPVL94] vorgenommen werden (siehe Abschnitt 5.2.1.2). Um die Gaußfunktion zu erhalten, muss das Eingangssignal  $x(k)$  quadriert werden. Dafür wird ein zusätzliches Basiselement benötigt<sup>12</sup>.

Die sigmoide Übertragungsfunktion kann durch ein Erweiterungsmodul und ein Basiselement dargestellt werden, da für die Approximation nach [BPVL94] eine Bereichsunterscheidung vorgenommen werden muss. Das Ergebnis der Potenzfunktion muss um eine Stelle nach rechts verschoben werden. In Abhängigkeit des Einganges wird das Ergebnis subtrahiert, was durch ein Basiselement vorgenommen werden kann. Für dieses Verfahren zur Berechnung der sigmoiden Aktivierungsfunktion kann der absolute Fehler kleiner als 0,019 gehalten werden [BPVL94], wobei die Effekte aufgrund begrenzter Genauigkeit berücksichtigt sind. Um mit Hilfe der Polynomapproximation einen Fehler in der gleichen Größenordnung in einem Eingangsintervall  $[-4; 4]$  zu erreichen, werden mindestens vier Basiselemente benötigt. So kann die benötigte Fläche für eine Approximation reduziert werden, da weni-

<sup>11</sup> Bei der SIRENS-Struktur wird als Zahlendarstellung das 2er-Komplement verwendet, so dass der negative Wert sehr einfach ermittelt werden kann.

<sup>12</sup> Dies wird aber ebenfalls bei der Polynomapproximation von  $\exp(-x)$  benötigt.

ger Elemente verwendet werden und die Fläche des Erweiterungsmoduls wesentlich kleiner als die Fläche eines Basiselementes ist (vgl. Ergebnisse aus Abschnitt 5.3.2). Der höhere Grad des Polynoms ergibt sich aufgrund der Tatsache, dass im Gegensatz zur Gaußfunktion keine gerade Funktion vorliegt und die Achsensymmetrie bei der Approximation nicht ausgenutzt werden kann.

So werden durch die rekonfigurierbare SIRENS-Struktur zwei verschiedene Modelle zu Approximation von Funktionen bereitgestellt. Zum einen kann mit Hilfe der Basiselemente jedes beliebige Polynom umgesetzt werden, wodurch ein parametrisches Modell für die Approximation zur Verfügung gestellt wird. Hiermit wird für die Regression nur eine endliche Anzahl von Parametern verwendet. Zum anderen erhält man mit der Aktivierungsfunktion eines Neurons ein nicht-parametrisches Modell für die Approximation. Dadurch lassen sich die in Kapitel 2.1 beschriebenen Vorteile erreichen, wobei die Aktivierungsfunktion aufgrund des parametrischen Ansatzes beliebig verändert werden kann, wodurch eine Approximation eines unbekanntem funktionalen Zusammenhanges erleichtert wird.

In der Literatur existieren ähnliche Ansätze für rekonfigurierbare, neuronale Systeme, wie z.B. [Ham90]. Dieser Ansatz beruht auf der Verbindung mehrerer elementarer Prozessorelemente, die durch ihre Struktur verschiedene Typen von Netzen abbilden können. Diese Struktur kann als ein Prozessorfeld angesehen werden, wobei die einzelnen Prozessoren für die Verarbeitung neuronaler Prinzipien entworfen wurden. Aus diesem Grund besitzt dieses System eine komplexe Kommunikationsstruktur mit Hilfe von Busprotokollen.

#### 5.2.1.4 SIRENS als Filterstruktur

Aufgrund der Verbindungsmatrix und der Struktur der Grundelemente aus SIRENS lassen sich alle grundlegenden Operationen aus dem Signalflussgraphen für die Abbildung einer Filterstruktur realisieren [Fra03]. Dadurch kann SIRENS auch für die Umsetzung von digitalen Filtern verwendet werden.

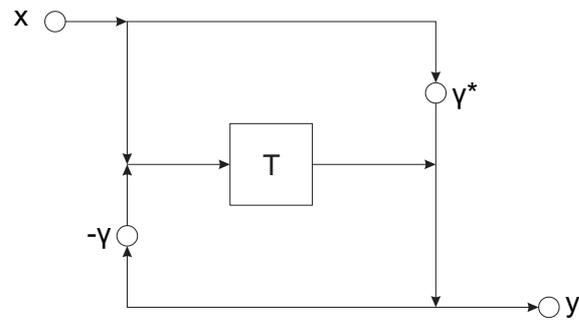
Eine Übertragungsfunktion mit einem Zähler- und Nennerpolynom vom Grad  $n$  kann durch eine Direktstruktur mit insgesamt  $2 \cdot n + 1$  Grundelementen dargestellt werden.

Daneben folgt für den Ausgangswert  $y(k)$  der digitalen Struktur aus Abbildung 5.6(b), bestehend aus zwei Grundelementen

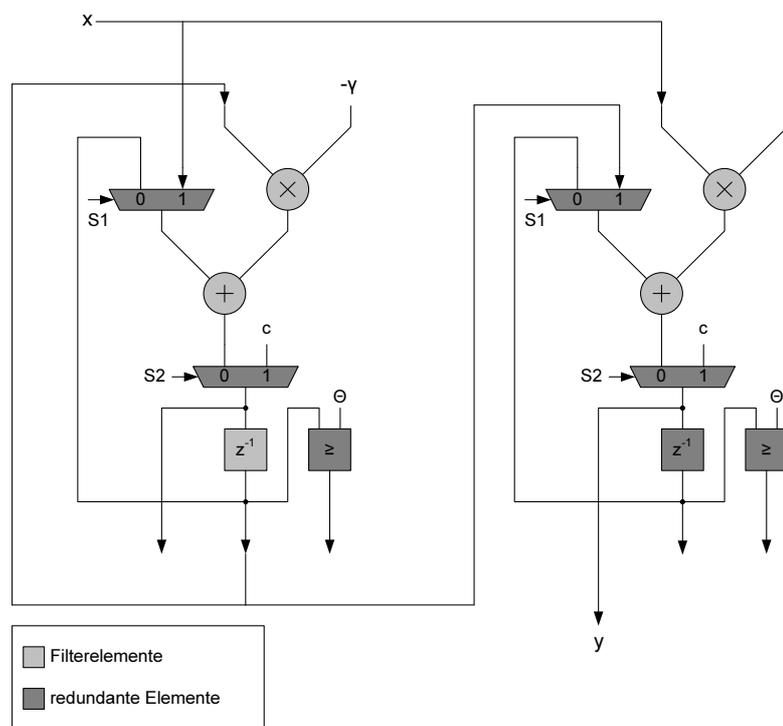
$$y(k) = \gamma x(k) + x(k-1) - \gamma y(k-1) \quad (5.56)$$

sowie für die Übertragungsfunktion

$$H(z) = \frac{\gamma + z^{-1}}{1 + \gamma z^{-1}}, \quad (5.57)$$



(a) Kreuzstruktur eines komplexen Allpasses



(b) Abbildung eines reellen Allpasses durch SIRENS

**Abbildung 5.6:** Realisierung von Allpässen

was eine reelle Allpassfunktion nach Abbildung 5.6(a) mit  $|H(e^{j\omega T})| = 1$  darstellt [Fra03]. Für die Auswirkungen der möglichen Strukturen und deren Vor- und Nachteile sei auf [Sch94] verwiesen.

Für die Darstellung der Übertragungsfunktion der Kreuzstruktur werden zwei Basiselemente der SIRENS-Struktur benötigt, siehe Abbildung 5.6(b). Allerdings ergeben sich zusätzliche Elemente, die zur Darstellung nicht benötigt werden. Dies sind jedoch nur die Komparatoren des Grundelementes, so dass der Überhang an zusätzlichen Recheneinheiten sehr gering ausfällt. Die Multiplexer könnten auch entfallen, was zusätzlich die Kosten reduzieren würde, vgl. Abbildung 5.6(b). Aufgrund der Struktur können nur reelle Koeffizienten verarbeitet werden.

*Bemerkung.* Der Zusammenhang zwischen neuronalen Netzen und Filterstrukturen ist in der Literatur eingängig untersucht worden. Dabei wurde gezeigt, dass neuronale Netze entsprechende Filterstrukturen und deren Verhalten beschreiben [MS00].

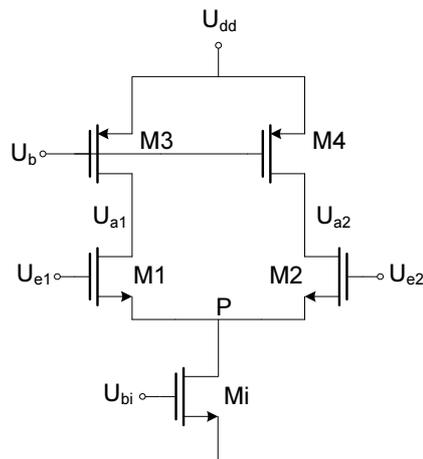
## 5.2.2 Analoger Entwurf

Neben einer digitalen Implementierung, die eine Quantisierung der Signale vornimmt, kann eine kontinuierliche Darstellung genutzt werden, um den Fehler aufgrund der Quantisierung zu vermeiden. Daneben können aufgrund der Eigenschaften von integrierten Bauelementen kleinere Schaltungen entstehen, da mit deren Übertragungsverhalten die Funktionalität sichergestellt wird. So kann durch den Einsatz eines Bipolartransistors oder von MOS-Transistoren im Betrieb unterhalb der Schwellenspannung die exp-Funktion nachgebildet werden. Der Kollektorstrom bzw. Drainstrom besitzen eine exponentielle Abhängigkeit von der Basis-Emitter-Spannung respektive der Gate-Source-Spannung.

Wie in Kapitel 2.2.1 vorgestellt wurde, kann für ein MLP neben der logistischen Funktion auch eine tanh-Funktion als Aktivierungsfunktion der Neuronen eingesetzt werden. So erhält man eine bipolare Repräsentation, wobei diese Übertragungsfunktion sehr leicht mit Hilfe von Transistoren umgesetzt werden kann. Für einen Differenzverstärker mit Bipolartransistoren oder MOS-Transistoren im Betrieb unterhalb der Schwellenspannung folgt für die Ausgangsspannung

$$U_a = R_C I_0 \tanh\left(\frac{\kappa_n}{2U_T} U_e\right), \quad (5.58)$$

wobei  $R_C$  den Lastwiderstand im jeweiligen Zweig,  $U_T$  die Temperaturspannung,  $I_0$  den eingepprägten Strom darstellt, und  $\kappa_n$  für die MOS-Transistoren den Einfluss der Gate-Spannung auf den Drainstrom modelliert. Neben dem Betrieb unterhalb der Schwellenspannung können die MOS-Transistoren im Sättigungsbereich betrieben werden, da sich hier ebenfalls eine sigmoide Übertragungsfunktion ergibt.



**Abbildung 5.7:** Analoge Realisierung der sigmoiden Funktion z.B. für ein MLP

Da aufgrund ihrer hohen Integrationsdichte und der kleinen Leckströme MOS-Transistoren weit verbreitet sind, wird in dieser Arbeit nur der Differenzverstärker auf Basis dieser Bauelemente analysiert und hinsichtlich der Skalierbarkeit für zukünftige Technologien bewertet.

Zur Umsetzung der sigmoiden Übertragungsfunktion wird die Schaltung aus Abbildung 5.7 verwendet, wobei alle MOS-Transistoren im Sättigungsbereich arbeiten. Dabei wird aus mehreren Gründen diese Differenzverstärkerschaltung zur Realisierung des Sigmoids genutzt. Zum einen können die Transistoren unterhalb der Schwellenspannung betrieben werden, so dass eine direkte Entsprechung der Übertragungsfunktion (5.58) erreicht wird. Zum anderen werden mit Hilfe dieser Architektur mehrere Vorteile erreicht. Aufgrund der Skalierungsfaktoren der CMOS-Technologie (siehe Tabelle 2.2) sinkt die Versorgungsspannung der kommenden Prozesstechnologien, während die Schwellenspannungen der Transistoren annähernd konstant bleiben [Sem05, ZC06]. Aufgrund dieses Umstandes verringert sich der Spielraum, die Transistoren in Sättigung zu betreiben, und so lassen sich Kaskodeschaltungen eines Differenzverstärkers nur schwer realisieren.

Weiter soll der Differenzverstärker so entworfen werden, dass die Spannungsverstärkung der Steigung der Aktivierungsfunktion  $\tanh(k \cdot x)$  entspricht, und eine Verstärkung von  $A = k$  für  $x = 0$  erreicht wird. Im Allgemeinen werden bei der Umsetzung der Aktivierungsfunktion kleine Steigungen ( $k \ll 100$ ) verwendet, so dass eine Kaskodeschaltung nicht notwendigerweise erforderlich ist. Als Last wird ein p-Kanal-Transistor mit einer Vorspannung verwendet. So kann über diese Spannung der Betriebszustand des Transistors eingestellt werden. Weiter ist der Spielraum bei einem Betrieb in Sättigung gegenüber einem p-Kanal-Transistor als Diode geschaltete Last größer [Raz00].

Für die grundlegenden Analysetechniken sei auf [Raz00] verwiesen. Da für den Verstärker keine ideale Stromquelle verwendet wird, ergibt sich neben der Spannungsverstärkung der Eingangsdifferenz auch noch eine Verstärkung der Gleichspannung. Das Sourcepotential der beiden Transistoren  $M1$  und  $M2$  bleibt jedoch aufgrund der Symmetrie der Schaltung konstant und kann als virtuelle Masse angesehen werden [Ehr00]. Dabei wird der Transistor  $M_i$  durch eine Stromquelle mit Innenwiderstand modelliert. Für die Analyse des Differenzverstärkers aus Abbildung 5.7 kann das *Half-Circuit*-Prinzip<sup>13</sup> [GMHL01] angewendet werden. Bei der Analyse wird davon ausgegangen, dass zusätzlich noch jeweils eine Last  $Z$  an den beiden Ausgangsspannungen vorhanden ist, und der Substrateffekt wird für die Transistoren  $M1$  und  $M2$  vernachlässigt.

Zur Anwendung des *Half-Circuit*-Konzeptes muss für die Symmetrie der Schaltung gelten

$$g_{m1} = g_{m2} \quad r_{o1} = r_{o2} \quad r_{o3} = r_{o4} \quad (5.59)$$

und so folgt direkt für die (differentielle) Spannungsverstärkung

$$A = \frac{U_{\text{out}}}{U_{\text{in}}} = \frac{U_{a1} - U_{a2}}{U_{e1} - U_{e2}} = -g_{m1} (r_{o1} \parallel r_{o3} \parallel Z), \quad (5.60)$$

wobei  $r_{oj}$ ,  $j \in \{1; 3; i\}$  die differentiellen Ausgangswiderstände der jeweiligen Transistoren bezeichnen, und  $g_{mi}$  die Transkonduktanz des (korrespondierenden) MOS-Transistors darstellt. Der Widerstand  $Z$  stellt weiterhin die Last am Ausgang dar, wobei die beiden Ausgänge mit der gleichen Last beschaltet sein sollen.

Für die Gleichtaktverstärkung ergibt sich dabei

$$A_{cm} = -\frac{g_{m1} r_{o1} (r_{o3} \parallel Z)}{(r_{o3} \parallel Z) + 2r_{oi} + (1 + 2g_{m1} r_{oi}) r_{o1}}, \quad (5.61)$$

wobei auch hier das *Half-Circuit*-Konzept angewendet werden kann. Für die Analyse ergibt sich dann eine Source-Stufe.

Für den Fall eines nicht symmetrischen Differenzverstärkers kann die Verstärkung über das Kleinsignalersatzschaltbild bestimmt werden. Eine Störung der Symmetrie liegt z.B. aufgrund von Fertigungstoleranzen vor, so dass sich unterschiedliche Transkonduktanzen oder Drainwiderstände ergeben. Diese Unterschiede resultieren zum einen in einer anderen differentiellen Spannungsverstärkung, und zum anderen wirkt sich eine Änderung des Gleichanteils ebenfalls auf diese Verstärkung aus. Dies bewirkt, dass Rauschen im Gleichanteil auch zu Rauschen in der differentiellen Ausgangsspannung führt. Für die Skalierung dieser Schaltung und die Simulation wurden die Transistormodelle aus [ZC06] verwendet.

<sup>13</sup> Alternativ kann die (differentielle) Spannungsverstärkung mit Hilfe des Überlagerungssatzes bestimmt werden, wobei die Spannungsquellen kurzgeschlossen werden.

## 5.3 Auswirkungen und Grenzen bedingt durch die Technologie

Ein neuronales Netz kann sowohl in digitaler als auch in analoger Schaltungstechnik implementiert werden. Dabei sollen die Vor- und Nachteile der beiden Varianten herausgestellt werden und beide Varianten hinsichtlich ihrer Skalierbarkeit näher untersucht werden. Im Fokus der Analysen stehen dabei die Auflösung und die Fläche der Netze.

### 5.3.1 Genauigkeit

Die Approximationsgenauigkeit der SIRENS-Struktur in Abhängigkeit der Auflösung der Signale soll näher betrachtet werden. Für die Simulationsergebnisse sind im Folgenden zur Darstellung der ganzen Zahlen einheitlich 5 bit verwendet worden, wobei die höchstwertige Stelle zur Repräsentation des Vorzeichens in der 2er-Komplement-Darstellung genutzt wird. Zur beispielhaften Veranschaulichung der Auswirkungen werden die sigmoide Übertragungsfunktion und die Gaußfunktion bzw. die Funktion  $g(x) = \exp(-x)$  verwendet, da es sich hier um populäre Aktivierungsfunktionen handelt. Dabei wurde die logistische Funktion im Intervall  $[-4; 4]$  durch ein Polynom vierten Grades approximiert. Die Gaußfunktion ist im Intervall  $[0; 3]$  und die exp-Funktion innerhalb von  $[0; 9]$  approximiert worden. Die Ergebnisse wurden auf der Basis einer gemeinsamen Simulation der VHDL-Beschreibung mit MATLAB und Modelsim gewonnen.

Bei der Berechnung führt das Grundelement aus Abbildung 5.3 eine Multiplikation zweier Eingänge durch. Da die Auflösung des Ausganges aufgrund der Homogenität der Struktur mit der eines Einganges identisch sein soll, muss eine Reduzierung der Wortbreite stattfinden. Aufgrund der Assoziativität der Addition ist es unerheblich, ob die Reduzierung vor oder nach der Addition stattfindet. Für die Verringerung der Wortbreite können verschiedene Verfahren eingesetzt werden. Es können die Stellen abgeschnitten werden, die Zahl zur nächsten Maschinenzahl gerundet werden oder die restlichen Stellen abgeschnitten werden, wobei die niederwertige Stelle der neuen Darstellung sich aus einer ODER-Verknüpfung der abgeschnitten Stellen berechnet. Eine mathematische Modellierung dieser Verfahren hinsichtlich ihres Erwartungswertes und der Varianz des Fehlers findet sich in [HH93]. Aufgrund mehrerer Architekturen innerhalb der VHDL-Beschreibung ist es möglich, die gewünschte Variante auszuwählen. In den hier vorgestellten Analysen wird die Variante des Abschneidens gewählt.

Tabelle 5.4 zeigt die Approximationseigenschaften der SIRENS-Struktur, falls ausschließlich die Grundelemente nach (5.53) verwendet werden, wobei  $q$  den Grad des verwendeten Polynoms angibt. Hier bezeichnet  $\Delta y$  den maximalen, theoretischen

**Tabelle 5.4:** Absoluter Fehler für die Approximation verschiedener Aktivierungsfunktionen mit Hilfe von SIRENS in Abhängigkeit der Auflösung für den Bruchanteil

Funktion	Breite Bruchanteil						$\Delta y$
	2	4	8	12	16	32	
Sigmoid [-4;4]	0,5419	0,2751	0,1327	0,0300	0,0231	0,0228	0,0227
$e^{-x}, q = 6$	15,4987	15,8776	15,9958	15,9878	2,7864	0,0187	0,0025
$e^{-x}, q = 3$	3,5002	5,4374	0,8166	0,1162	0,0768	0,0746	0,0709
$e^{-x^2}, q = 5$	9,7535	5,8749	0,3829	0,0753	0,0363	0,0359	0,0048
$e^{-x^2}, q = 3$	1,2510	0,6874	0,1234	0,0643	0,0644	0,0643	0,0618

Fehler aufgrund der Approximationsart, und die Tabelle zeigt den absoluten Fehler in Abhängigkeit der verwendeten Auflösung zur Darstellung des Bruchs. Als Breite für den ganzzahligen Anteil der Zahlendarstellung werden 4 bit und eine zusätzliche Stelle für das Vorzeichen verwendet.

An den Simulationsergebnissen zeigen sich die Erkenntnisse aus Abschnitt 5.1. Mit einem hohen Grad des Approximationspolynoms entsteht aufgrund der Fortpflanzung der Abweichungen ein hoher absoluter Fehler. Um eine möglichst gute Approximation der Aktivierungsfunktion zu erreichen, muss bei der Verwendung von Polynomen hohen Grades ebenfalls eine hohe Auflösung eingesetzt werden, um so die Quantisierungsfehler gering zu halten. Eine Reduzierung des Polynomgrads wirkt sich positiv auf die benötigte Fläche aus. Zum einen werden weniger Grundelemente benötigt, und zum anderen kann eine geringere Auflösung eingesetzt werden, wobei beide Effekte zu einem kleineren System beitragen (siehe ebenfalls Abschnitt 5.3.2). Für die Approximation einer Funktion kann es somit besser sein, ein kürzeres Polynom zu verwenden, ohne Einbußen in der Approximationsgüte hinnehmen zu müssen.

Die Ergebnisse zeigen weiter, dass eine Verkleinerung des Intervalls sich positiv auf die Eigenschaften der Approximation auswirkt. Durch die Verwendung der Gaußfunktion im Gegensatz zur Funktion  $g(x) = \exp(-x)$  kann ein kleineres Intervall genutzt werden, obwohl der gleiche Ausgangsbereich abgedeckt wird. Hier ergeben sich für die untersuchten Auflösungen bessere Ergebnisse für die Gaußfunktion. Dies ermöglicht eine Nutzung höherer Polynome bei geringeren Auflösungen, ohne schlechte Ergebnisse der Approximation akzeptieren zu müssen. Für kleinere Polynome nähert sich der Approximationsfehler bei hohen Auflösungen dem theoretischen Minimum an, was ebenfalls für die besseren Eigenschaften der Approximation spricht.

Die Genauigkeit der Approximation hängt im Wesentlichen von der aktuellen Anforderung an das neuronale Netz ab, wobei diese über zwei Parameter, der Länge des Polynoms und die verwendete Auflösung, gesteuert werden kann. Da die verwendete Auflösung der Struktur jedoch festgelegt ist, ergibt sich für das Problem ein maximaler Grad des Polynoms in Abhängigkeit der Approximationsgüte.

**Tabelle 5.5:** Absoluter Fehler für die Approximation mit Hilfe von SIRENS und der Erweiterung in Abhängigkeit der Auflösung

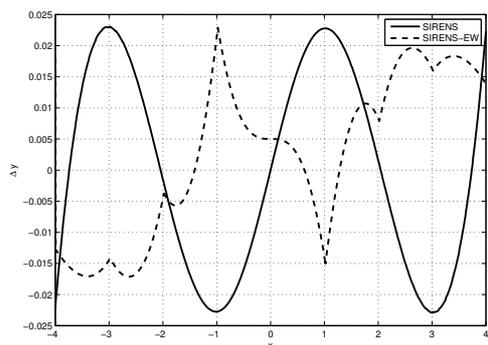
Funktion	Breite Bruchanteil						$\Delta y$
	2	4	8	12	16	32	
Sigmoid [-4;4]	0,2479	0,0775	0,0249	0,0229	0,0229	0,0229	0,0245 <sup>1</sup>
$\exp(-x)$	0,2227	0,1758	0,1641	0,1635	0,1634	0,1634	0,1341

<sup>1</sup> Der hier angegebene Fehler bezieht sich auf die abschnittsweise definierte Funktion aus [BPVL94] ohne eine stückweise lineare Approximation.

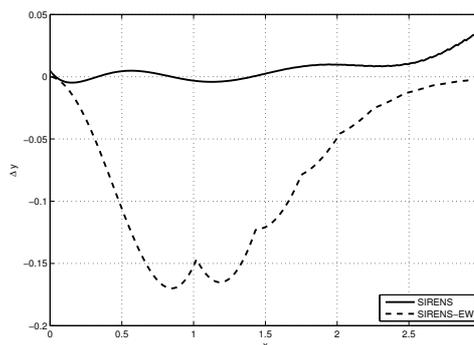
Tabelle 5.5 zeigt die Eigenschaften der Approximation unter zusätzlicher Verwendung des Erweiterungsmoduls aus (5.55) bzw. Abbildung 5.4(b), wobei  $\Delta y$  den maximalen, theoretischen Fehler angibt. Es zeigt sich eine deutliche Verbesserung der Approximationsgüte des Systems, die zudem nahezu unabhängig von der verwendeten Auflösung ist. Allerdings kann durch die Erweiterung kein infinitesimal kleiner Fehler erreicht werden, da der absolute Fehler durch sein theoretisches Minimum bedingt durch die Approximationsart nach unten beschränkt ist. So reicht bei der Darstellung eines Sigmoids bei der Erweiterung eine Auflösung von 4 bit für die Bruchdarstellung aus, während die normale Struktur für einen ähnlichen absoluten Fehler eine Auflösung von 12 bit benötigt.

Ebenso deutlich fällt der Vergleich bei der exp-Funktion als Aktivierungsfunktion aus. So erzielt die Erweiterung bei einer Auflösung von 4 bit ähnliche Ergebnisse wie die Approximation durch ein Polynom mit 8 bit bzw. 16 bit. Der Fehler der Erweiterung basiert hier fast ausschließlich auf der Art der Approximation der exp-Funktion durch eine Potenzfunktion zur Basis zwei für  $x \in \mathbb{R}^-$ . Die Abweichung von dem theoretischen Minimum ergibt sich aus der stückweise, linearen Approximation der Potenzfunktion, welche in der Tabelle mitberücksichtigt wurde.

Jedoch zeigt die Approximation der Aktivierungsfunktion mit Hilfe dieses Schemas eine erhebliche Reduzierung des absoluten Fehlers bei geringen Auflösungen. Die Größe des Fehlers wird dabei hauptsächlich durch die Art der Approximation bestimmt und ist weitgehend unabhängig von der Breite des Intervalls und der Auflösung. Die einzige Abhängigkeit des Fehlers von der Intervallbreite resultiert aus dem Verlauf des theoretischen absoluten Fehlers (siehe Abbildung 5.8). So kann je nach Art der Anforderungen die Art der Approximation gewählt werden.



(a) Absoluter Fehler beim Sigmoid



(b) Absoluter Fehler bei der Gaußfunktion

**Abbildung 5.8:** Approximationseigenschaften der SIRENS-Struktur mit Erweiterungsmoduls

Abbildung 5.8 zeigt den Verlauf des absoluten Fehlers für die SIRENS-Struktur mit und ohne Erweiterungen<sup>14</sup>. Für die Approximation der Aktivierungsfunktion in Abbildung 5.8 wird eine Auflösung des Grundelementes von 32 bit für die Bruchdarstellung verwendet, während für das Erweiterungsmodul eine Auflösung von 16 bit für den Bruch bereitgestellt wird.

Das Erweiterungsmodul kann die numerischen Eigenschaften der Architektur verbessern und gleichzeitig die Anforderungen an die Fläche reduzieren. Für den Sigmoid in Abbildung 5.8(a) reicht die halbe Auflösung aus, um einen absoluten Fehler in der gleichen Größenordnung zu erreichen. Die geringere Auflösung wirkt sich besonders bei einer Multiplikation günstig auf die Fläche aus. Abbildung 5.8(b) zeigt die Stärken und Schwächen des Erweiterungsmoduls. Aufgrund der Approximationsart der Gaußfunktion durch eine Potenzfunktion ergibt sich der maximale absolute Fehler für einen Eingangswert  $x \approx 0,75$ . Dieser ist wesentlich höher als die Approximation durch ein Polynom an der gleichen Stelle. Allerdings ist der Aufbau des Erweiterungsmoduls wesentlich numerisch stabiler, was besonders in den Randbereichen deutlich wird. Hier ergeben sich bessere Eigenschaften als bei der Verwendung des Polynomansatzes.

### 5.3.2 Syntheseresultate

Zur Analyse der rekonfigurierbaren Struktur wurde die VHDL-Beschreibung auf den Standardzellenprozess *HCMOS* von STMicroelectronics [STM05] umgesetzt. Dieser Fertigungsprozess besitzt eine (effektive) minimale Gatelänge von  $0,12 \mu\text{m}$  und eine Versorgungsspannung von 1,2 V. Als Betriebsbedingungen für die Umsetzung wurden die schlechtesten Randbedingungen der Kategorie „Laborbedingungen“ ge-

<sup>14</sup>Die Bezeichnung *SIRENS-EW* steht für ein System mit dem Erweiterungsmodul.

wählt. Hier beträgt die Versorgungsspannung 1,08 V und die Temperatur 358 K. Als Entwurfswerkzeug wurde der *Design Compiler* von Synopsys verwendet.

**Tabelle 5.6:** Ergebnisse der Abbildung eines Basiselementes und des Erweiterungsmoduls auf die Technologie von ST

Auflösung <sup>1</sup>	SIRENS			SIRENS-EW		
	A <sup>2</sup> [ $\mu\text{m}^2$ ]	P <sup>3</sup> [mW]	T <sup>4</sup> [ns]	A <sup>2</sup> [ $\mu\text{m}^2$ ]	P <sup>3</sup> [mW]	T <sup>4</sup> [ns]
1	1430,19	0,75	1,00	883,53	0,22	0,71
2	2065,61	1,09	1,06	1385,82	0,29	0,85
3	3001,59	1,64	1,15	2178,58	0,40	0,97
4	4901,80	2,81	1,23	2884,60	0,44	1,05
5	6634,57	3,56	1,32	3915,39	0,63	1,19
6	8359,28	4,82	1,42	6110,10	0,95	1,22
7	9031,00	5,39	1,48	6164,56	2,82	1,29
8	10265,53	6,08	1,57	6898,82	2,62	1,33
9	12986,73	8,57	1,56	8460,14	1,25	1,43
10	14763,89	9,95	1,64	11096,62	1,63	1,43
11	16952,55	11,70	1,70	10868,67	5,34	1,50
12	19407,48	13,90	1,71	12793,08	6,33	1,52
13	22667,28	16,50	1,73	16054,89	8,87	1,57
14	24962,85	18,10	3,18	17212,77	9,06	1,58
15	28307,37	21,40	1,83	20696,47	12,80	1,67
16	30842,99	24,10	1,87	21491,25	12,50	1,68

<sup>1</sup> Die Auflösung gibt die Anzahl der Stellen nach dem Festpunkt an.

<sup>2</sup> Fläche: Hierbei ist die Fläche für die Verdrahtung nicht berücksichtigt worden.

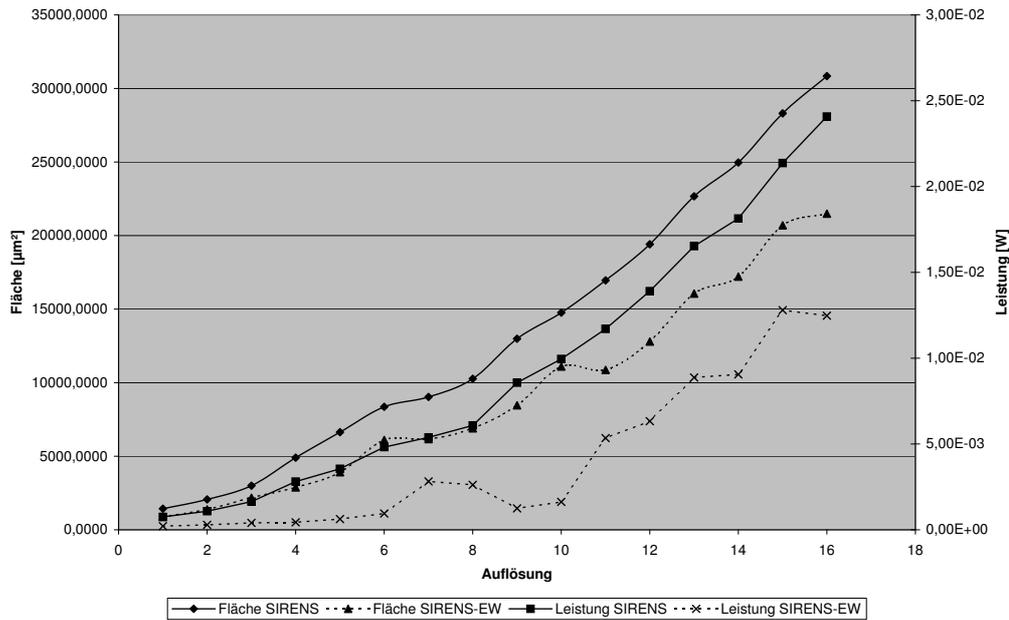
<sup>3</sup> Leistung: Hier wird die gesamte dynamische Leistung bei einer Frequenz von 500 MHz angegeben.

<sup>4</sup> längster Pfad

Tabelle 5.6 und Abbildung 5.9 zeigen die Ergebnisse der Abbildung der Struktur auf die Technologie von ST. Zur Bestimmung der Leistung wurde dabei eine Frequenz von 500 MHz gewählt, um die Entwürfe miteinander vergleichen zu können.

Hier zeigt sich der geringere Flächenverbrauch des Erweiterungsmoduls. Das Basismodul benötigt sowohl einen Multiplizierer als auch einen Addierer (vgl. Abbildung 5.3), während das Erweiterungsmodul nur einen Multiplizierer benötigt<sup>15</sup>. Diese wird in der VHDL-Beschreibung mit doppelter Auflösung berechnet. Für eine Optimierung des Entwurfes ließe sich die Wortbreite weiter verringern, was sich positiv auf die Fläche auswirken würde. Auf eine Addition kann bei dem Erweiterungsmodul verzichtet werden, da diese durch geschickte Kombination der Signale erreicht

<sup>15</sup> Für die Umsetzung der Potenzfunktion sei auf [BPVL94] verweisen.



**Abbildung 5.9:** Fläche und Leistung des Basiselementes und des Erweiterungsmoduls

werden kann. Dadurch ergibt sich für das Erweiterungsmodul ein geringerer Flächenbedarf.

Bedingt durch weniger kombinatorische Logik fällt die Leistung des Erweiterungsmoduls im Vergleich zu einem Basiselement ebenfalls geringer aus. Das Phänomen einer geringeren Leistung des Erweiterungsmoduls bei einer Auflösung von 9 bzw. 10 bit lässt sich nicht erklären. Obwohl es sich bei der Synthese um einen stochastischen Optimierungsprozess handelt, wird in diesen beiden Fällen mehr kombinatorische Logik benötigt, so dass eigentlich eine höhere Leistung als bei kleineren Entwürfen entstehen sollte.

Ebenfalls kann ein geringerer kritischer Pfad gegenüber dem Basismodul mit dem Erweiterungsmodul erreicht werden. Dieser ergibt sich aufgrund der fehlenden Addition, wobei die zusätzlich benötigte Schiebeoperation diesen Vorteil teilweise kompensiert. Um ein allgemeingültiges System zu erreichen, wurde bei dem Erweiterungsmodul eine Schiebeoperation um eine beliebige Anzahl von Stellen vorgesehen. Soll durch das Modul jedoch nur die Approximation der sigmoiden Funktion oder der Gaußfunktion verbessert werden, so kann diese entfallen, und der Geschwindigkeitsvorteil fällt deutlicher aus.

Zusammenfassend bietet das Erweiterungsmodul gegenüber dem Grundmodul bessere Eigenschaften bei der Umsetzung auf einen Standardzellenprozess. Einschränkungen sind hier allerdings hinsichtlich der Approximationsarten zu machen, da mit dem Erweiterungsmodul nur bestimmte Aktivierungsfunktionen approximiert werden können. Auch ist eine Darstellung der Ableitung mit diesem Modul nicht

möglich. Darüber hinaus ist der Entwurf des Grundelementes nicht auf digitale Systeme beschränkt, wobei in dem Erweiterungsmodul spezielle digitale Techniken eingesetzt wurden. Für eine analoge Umsetzung können die Elemente aus Abbildung 5.3 in die entsprechenden Schaltungsteile, wie z.B. eine Gilbert-Zelle oder eine Kapazität zur Speicherung der Werte, umgesetzt werden, was bei einem Erweiterungsmodul nicht möglich ist.

Neben den besseren Eigenschaften bei der Umsetzung auf eine Technologie bietet das Erweiterungsmodul zudem bessere Approximationseigenschaften bei gleicher Auflösung. Hier sollen im Folgenden für die beiden Strukturen die Punkte im Entwurfsraum betrachtet werden. Dafür wird auf die Aspekte des Approximationsfehlers  $\Delta y$  und die Fläche  $A$  des Elementes näher eingegangen. Da es sich bei digitalen Systemen i.A. um synchrone Systeme handelt, bietet der geringere kritische Pfad des Erweiterungsmoduls bei einer inhomogenen SIRENS-Struktur keinen Geschwindigkeitsvorteil, da der Takt von dem Pfad des Grundmoduls bestimmt wird.

Für die Auswertung hinsichtlich Fläche  $A$  und Genauigkeit  $\Delta y$  der Module kann zur Bewertung eine Kostenfunktion aufgestellt werden. Hier kann z.B. die multiplikative Verknüpfung<sup>16</sup> der beiden Eigenschaften zur Analyse herangezogen werden:

$$F(A, \Delta y) = A \cdot \Delta y. \quad (5.62)$$

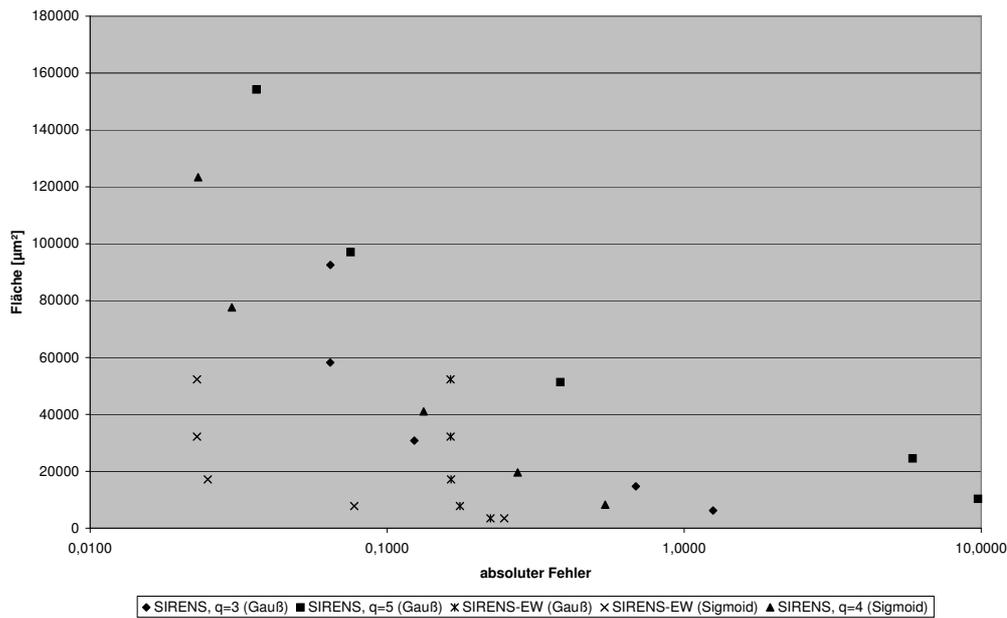
Dabei hängt die benötigte Fläche nicht nur von dem Basismodul ab, sondern ebenfalls von der darzustellenden Aktivierungsfunktion. Dafür werden die Polynome und Funktionen aus Abschnitt 5.3.1 verwendet. Für die Fläche wird von einem linearen Wachstum in Abhängigkeit des Polynomgrades ausgegangen, wobei aufgrund des zusätzlichen Flächenbedarfs der Verbindungsleitungen zwischen den Elementen die Fläche bei einem realen System stärker ansteigen wird.

Abbildung 5.10 zeigt die Punkte im Entwurfsraum des Systems bei der Approximation verschiedener Aktivierungsfunktionen. Eine Bewertung des jeweiligen Systems kann in Anlehnung an die Kostenfunktion aus (5.62) vorgenommen werden, indem die Fläche betrachtet wird, die durch den Punkt im Entwurfsraum und den beiden Koordinatenachsen aufgespannt wird. Eine Minimierung von (5.62) entspricht somit einem Punkt möglichst nahe des Ursprungs. Aufgrund dieser Analogie wird auf eine explizite Darstellung der Kostenfunktion für die einzelnen Systeme in Abhängigkeit der Wortbreite verzichtet.

Betrachtet man die Punkte im Entwurfsraum, so erkennt man den positiven Einfluss des Erweiterungsmoduls besonders für geringe Auflösungen<sup>17</sup>. Dadurch, dass die

<sup>16</sup> Bei der multiplikativen Verknüpfung können die einzelnen Eigenschaften noch über höhere Potenzen gewichtet werden.

<sup>17</sup> Geringe Auflösungen entsprechen aufgrund eines annähernd linearen Zusammenhangs einem kleinen Flächenbedarf.



**Abbildung 5.10:** Entwurfspunkte der Basismodule für verschiedene Aktivierungsfunktionen

Approximationsgüte der Aktivierungsfunktion bei der Erweiterung nicht durch die numerische Stabilität bestimmt wird, sondern durch die Art der Approximation, erhält man im Sinne der Kostenfunktion ein besseres System, da insgesamt weniger Fläche bei gleicher Auflösung benötigt wird. Diese Eigenschaft kann an dem Anstieg der Fläche für das Erweiterungsmodul ab einem gewissen absoluten Fehler abgelesen werden. Durch eine größere Wortbreite erhöht sich die Fläche, ohne den absoluten Fehler zu reduzieren. Dies ist in der Approximationsart begründet, so dass sich der Fehler für hohe Auflösungen seinem theoretischen Minimum annähert.

Weiter zeigt sich in Abbildung 5.10 der Vorteil eines geringen Polynomgrades bei der Approximation. Durch die kleineren Potenzen ergibt sich im Ausgang ein kleinerer absoluter Fehler, so dass eine geringere Auflösung verwendet werden kann. Bei den hier untersuchten Auflösungen entsteht bei der Abbildung der Gaußfunktion durch ein kleineres Polynom ein geringerer absoluter Fehler. Daneben werden zur Darstellung des Polynoms selbst weniger Basiselemente benötigt, was sich ebenfalls positiv auf die Fläche auswirkt. So stellt dies ein effizienteres System im Vergleich zu einem System mit mehreren Basiselementen im Sinne der Kostenfunktion dar. Ist man ausschließlich an einer hohen Genauigkeit interessiert, so können hohe Potenzen bei der Approximation zusammen mit einer hohen Auflösung verwendet werden. Allerdings entstehen so sehr große Systeme, die aufgrund der hohen Anzahl an kombinatorischer Logik ebenfalls viel Leistung benötigen.

Weiter zeigt sich, dass für eine Approximation des Sigmoids das Erweiterungsmodul sehr gut geeignet ist. Es ergibt sich gegenüber dem Basismodul ein kleineres Sys-

tem, das gleichzeitig einen geringeren Approximationsfehler besitzt. Während sich bei der Abbildung der Gaußfunktion bzw. der exp-Funktion die Achsensymmetrie ausnutzen lässt, kann diese Technik bei der sigmoiden Funktion nicht angewendet werden, so dass Polynome höheren Grades für eine ähnliche Approximationsgüte verwendet werden müssen. Dadurch ergibt sich die Anforderung an eine hohe Auflösung; das gleiche Phänomen, welches bei der Gaußfunktion beobachtet werden kann. Bei zu geringer Wortbreite entstehen in den nahezu konstanten Bereichen des Sigmoids (für  $x \rightarrow \infty$  bzw.  $x \rightarrow -\infty$ ) aufgrund der Fehlerfortpflanzung Oszillationen bei der Funktionswertberechnung. Diese können vermieden werden, wenn eine Überlaufkontrolle in das Basismodul eingefügt wird. Dadurch wird die Komplexität des Systems weiter erhöht, so dass aufgrund der besseren Eigenschaften das Erweiterungsmodul eingesetzt werden sollte.

### 5.3.3 Rauschen

Bei der analogen Verarbeitung entsteht durch die Verarbeitung der Signale durch die Schaltungen zusätzlich Rauschen, das sich additiv den Nutzsignalen überlagert. Für die weitere Analyse wird angenommen, dass die Transistoren im Sättigungsbereich arbeiten. Dabei treten verschiedene Rauschquellen auf, die alle Einfluss auf das Signal-Rausch-Verhältnis (SNR) besitzen [LKI<sup>+</sup>02].

Einen wichtigen Anteil an der Rauschleistung stellt das thermische Rauschen dar. Die Rauschleistungsdichte<sup>18</sup> aufgrund des Drainwiderstands kann dabei durch

$$\overline{I_n^2} = 4k_B T \gamma g_m \quad (5.63)$$

beschrieben werden, wobei  $k_B$  die Boltzmannkonstante und  $T$  die Temperatur bezeichnet. Der Parameter  $\gamma$  besitzt viele Abhängigkeiten, z.B. von der Drain-Source-Spannung [Man99, STL<sup>+</sup>03], und ist immer noch das Ziel theoretischer Untersuchungen. Dieser Rauschstrom wird durch eine Stromquelle zwischen Drain und Source modelliert. Damit folgt für die Rauschspannung am Gate

$$\overline{v_n^2} = \frac{\overline{i_n^2}}{g_m^2}. \quad (5.64)$$

Ein weiterer Rauschanteil entsteht durch das *1/f-Rauschen* oder *Flicker-Rauschen* bedingt durch Defekte in der Kristallstruktur und für diesen gilt

$$\overline{v_n^2} = \frac{K}{C_{ox} A} \cdot \frac{1}{f}, \quad (5.65)$$

<sup>18</sup>In dieser Arbeit werden die Rauschleistungsdichten verwendet.



Verstärkung der Schaltung festgelegt, da bei einer sigmoiden Übertragungsfunktion explizit Nichtlinearitäten gewünscht sind.

Zuerst soll das Rauschen des Transistors  $M3$  betrachtet werden. Dabei teilt sich der Rauschstrom des Transistors  $M3$  auf den Widerstand  $r_{o3}$  und den Widerstand  $R_X$  auf, wobei  $R_X$  den äquivalenten Widerstand bezeichnet, den man durch den Drainanschluss des Transistors  $M1$  sieht. Dabei folgt für  $R_X$  unter der Annahme, dass  $M1$  und  $M2$  symmetrisch sind

$$R_X = r_{o1} \left( 1 + \frac{g_{m1} r_{oi}}{1 + \frac{r_{oi}(1+g_{m1}r_{o1})}{r_{o4}+r_{o1}}} \right) + \frac{r_{oi}}{1 + \frac{r_{oi}(1+g_{m1}r_{o1})}{r_{o4}+r_{o1}}}. \quad (5.67)$$

Somit kann mit Hilfe des Stromteilers der Rauschstrom durch den Widerstand  $r_{o3}$  berechnet werden, und es folgt für die Rauschspannung am Ausgang  $X$  infolge des rauschenden Transistors  $M3$

$$U_{nX} = g_{m3} U_{n3} \frac{r_{o3} R_X}{r_{o3} + R_X}, \quad (5.68)$$

wobei  $U_{n3}$  die Rauschspannung des Transistors  $M3$  bezeichnet.

Die Rauschspannung an dem Punkt  $Y$  kann mit Hilfe des Rauschstromes durch  $R_X$  bestimmt werden. Insgesamt fließt ein Rauschstrom von

$$I_{n4} = \frac{g_{m3} U_{n3} r_{o3} r_{oi} (1 + g_{m1} r_{o1})}{(r_{o3} + r_{o1})(r_{o1} + 2r_{oi} g_{m1} r_{o1} + r_{o3} + 2r_{oi})} \quad (5.69)$$

durch den Drainwiderstand  $r_{o4}$  des Transistors  $M4$ , wobei die gesamte Rauschspannung an diesem Widerstand abfällt. So ergibt sich für die Differenzspannung

$$U_{na} = U_{nX} - U_{nY} = g_{m3} U_{n3} (r_{o1} \parallel r_{o3} \parallel Z). \quad (5.70)$$

Aufgrund der Symmetrie des Differenzverstärkers in Abbildung 5.11 ergibt sich für die Rauschspannung am Ausgang aufgrund des Rauschstroms von Transistor  $M4$  der gleiche Zusammenhang wie zuvor für den Transistor  $M3$ . Da die Quellen voneinander unabhängig sind, können die Leistungen direkt überlagert werden. Der Einfluss des Transistors  $Mi$  ist aufgrund des symmetrischen Aufbaus der Verstärkerstufe am Ausgang nicht vorhanden. Damit ergibt sich für das eingangsbezogene Rauschen mit der Verstärkung aus (5.60)

$$\overline{U}_{ne}^2 = 2\overline{U}_{n1}^2 + \frac{2g_{m3}^2 (r_{o1} \parallel r_{o3} \parallel Z)^2}{A^2} \overline{U}_{n3}^2 = 2\overline{U}_{n1}^2 + \frac{2g_{m3}^2}{g_{m1}^2} \overline{U}_{n3}^2 \quad (5.71)$$

und so für die Rauschleistungsdichte am Ausgang

$$\overline{U}_{na}^2 = 2\overline{U}_{n1}^2 \cdot g_{m1}^2 (r_{o1} \parallel r_{o3} \parallel Z)^2 + 2g_{m3}^2 (r_{o1} \parallel r_{o3} \parallel Z)^2 \overline{U}_{n3}^2. \quad (5.72)$$

Bei Berücksichtigung von thermischem Rauschen und Flicker-Rauschen kann die Ausgangsleistung bestimmt werden zu

$$P_{na} = \int_{f_1}^{f_2} \overline{U^2}_{na} df = P_{t1} + P_{f1} + P_{t3} + P_{f3}, \quad (5.73)$$

wobei die jeweiligen Leistungen  $P_{ti}$  aus dem thermischen Rauschen und  $P_{fi}$  aus dem 1/f-Rauschen der Transistoren  $M1$  und  $M3$  resultieren und  $f_1$  bzw.  $f_2$  die untere und obere Grenzfrequenz bezeichnen. Für diese Zusammenhänge gilt

$$P_{t1} = \frac{4k_B\gamma T g_{m1} r_{o3} r_{o1} \left[ \arctan\left(\frac{2f_2 r_{o1} r_{o3} \pi C_L}{r_{o1} + r_{o3}}\right) - \arctan\left(\frac{2f_1 r_{o1} r_{o3} \pi C_L}{r_{o1} + r_{o3}}\right) \right]}{\pi(r_{o1} + r_{o3})C_L} \quad (5.74)$$

$$P_{f1} = K g_{m1}^2 (r_{o1} \parallel r_{o3})^2 \frac{\ln\left(\frac{1+4(r_{o1} \parallel r_{o3})^2 C_L^2 \pi^2 f_1^2}{1+4(r_{o1} \parallel r_{o3})^2 C_L^2 \pi^2 f_2^2}\right) + 2 \ln\left(\frac{f_2}{f_1}\right)}{C_{ox} W_1 L_1} \quad (5.75)$$

$$P_{t3} = \frac{4k_B\gamma T g_{m3} r_{o3} r_{o1} \left[ \arctan\left(\frac{2f_2 r_{o1} r_{o3} \pi C_L}{r_{o1} + r_{o3}}\right) - \arctan\left(\frac{2f_1 r_{o1} r_{o3} \pi C_L}{r_{o1} + r_{o3}}\right) \right]}{\pi(r_{o1} + r_{o3})C_L} \quad (5.76)$$

$$P_{f3} = K g_{m3}^2 (r_{o1} \parallel r_{o3})^2 \frac{\ln\left(\frac{1+4(r_{o1} \parallel r_{o3})^2 C_L^2 \pi^2 f_1^2}{1+4(r_{o1} \parallel r_{o3})^2 C_L^2 \pi^2 f_2^2}\right) + 2 \ln\left(\frac{f_2}{f_1}\right)}{C_{ox} W_3 L_3}, \quad (5.77)$$

wobei für die Last  $Z$  eine kapazitive Last  $C_L$  angenommen wurde.

So folgt für den Zusammenhang des Signal-Rausch-Verhältnisses am Ausgang des Verstärkers bei ausschließlich thermischem Rauschen

$$\text{SNR} \propto g_{m1} C_L \quad (5.78)$$

bzw.

$$\text{SNR} \propto 1/g_{m3}. \quad (5.79)$$

Wie man an (5.78) und (5.79) erkennen kann, muss für eine Maximierung des SNR die Transkonduktanz des Transistors  $M1$  maximiert werden, während  $g_{m3}$  zu minimieren ist. So wird der Effekt aufgrund des thermischen Rauschens minimiert. Ebenfalls kann die Lastkapazität maximiert werden, was allerdings zu einer Reduzierung der 3dB-Grenzfrequenz der Verstärkung führt. Um auch den Einfluss des Flicker-Rauschens zu minimieren und so die Rauschleistungen nach (5.75) bzw. (5.77) zu reduzieren, muss die Fläche der Transistoren vergrößert werden.

Dies bedeutet, dass für ein sehr hohes SNR der Strom maximiert werden muss, da  $g_{m1}$  möglichst groß sein soll. Dadurch steigt die Leistung des Verstärkers an und ebenfalls erhöht sich die Fläche, da größere W/L-Verhältnisse benötigt werden. So besteht hier ein Bedarf zur exakten Erfassung der Zusammenhänge und einer unterstützenden Optimierung. Da hier mehrere Ziele verfolgt werden, wie z.B. minimale Fläche, maximales SNR und minimale Leistung bei einer gewünschten Verstärkung

als Randbedingung, ist der Einsatz von Mehrzieloptimierung, wie z.B. [DSH05], sinnvoll.

Für die Bestimmung der maximalen Auflösung aufgrund des SNR wird die Skalierung mit der CMOS-Technologie betrachtet. Die Parameter für diese Skalierung sind [ZC06] und der Theorie aus Abschnitt 2.3.1 entnommen. Dazu wird in einem ersten Schritt nur das thermische Rauschen der Transistoren betrachtet und gleiche Transkonduktanzen verwendet. Zur Bestimmung der äquivalenten Wortbreite für eine digitale Implementierung kann die Informationstheorie [Sha48] verwendet werden. So ergibt sich für das maximale SNR, welches durch die Rauschleistung bestimmt wird

$$\text{SNR} = \frac{u_m^2 g_{m1}(r_{o1} \parallel r_{o3}) C_L}{8\gamma k_B T}, \tag{5.80}$$

wobei  $u_m$  die Amplitude des Eingangssignals bezeichnet.

Die maximale Auflösung, begrenzt durch das thermische Rauschen der Transistoren, skaliert nach (5.80) nur bedingt mit der Technologie, da die Verstärkung von der Steilheit des Sigmoids abhängig ist. Eine leichte Herabsetzung erfolgt durch die geringeren Amplituden  $u_m$  des Eingangssignals, da die Versorgungsspannung verringert wird. Alle Effekte des Drainrauschstromes werden in dem Parameter  $\gamma$  zusammengefasst, wobei der theoretische Wert für Transistoren mit großen Kanallängen  $2/3$  beträgt. Für kürzere Kanallängen erhöht sich der Einfluss von parasitären Widerständen, so dass  $\gamma$  ansteigt [STL<sup>+</sup>03]. Zum Teil müssen neue Modelle für das thermische Rauschen aufgrund des Einflusses heißer Ladungsträger erstellt werden, wie z.B. [Kle99, KKT01], da das Modell aus (5.63) für Kanallängen unterhalb von  $2\ \mu\text{m}$  gegenüber experimentellen Daten abweicht [Kle99].

Die Lastkapazität des Neurons setzt sich aus der Ausgangskapazität des Differenzverstärkers und der Eingangskapazität der nächsten Stufe zusammen. Da bei einem MLP der Sigmoid mit einem Gewicht multipliziert wird und zur Bildung der Ausgabe mit anderen Funktionen linear überlagert wird, besteht die nachfolgende Stufe aus einer Gilbert-Zelle<sup>19</sup>. So kann die Lastkapazität an dem Punkt  $X$  bzw.  $Y$  aus Abbildung 5.11 abgeschätzt werden

$$C_L = C_{out} + C_{in} \tag{5.81}$$

mit

$$\begin{aligned} C_{out} &\approx C_{DBn} + C_{DBp} + \left(1 + \frac{1}{g_{m1}(r_{o1} \parallel r_{o3})}\right) \cdot (C_{GDn} + C_{GDp}) \\ &= 2C_{DB} + 4C_{GD} \end{aligned} \tag{5.82}$$

sowie

$$C_{in} \approx C_{GSn} + (1 - A)C_{GDn} = C_{GSn} + 2C_{GDn} \approx C_{GSn}. \tag{5.83}$$

<sup>19</sup>Die Gilbert-Zelle wird für die Multiplikation mit dem Gewicht eingesetzt.

Hierbei bezeichnet  $C_{XYZ}$  die Kapazität zwischen dem Anschluss  $X$  und  $Y$  mit der Anschlussbezeichnung: Gate (G), Drain (D), Source (S). Die letzte Stelle des Index bezeichnet den Transistortyp. Dabei ist Kapazität zwischen dem Gate und dem Oxid größer als die Kapazität aufgrund der Überlappung des Gates mit dem Source- bzw. Drainanschluss. Zusätzlich wird angenommen, dass für die Multiplikation mittels Gilbert-Zelle die Verstärkung  $A = 1$  gewählt wird und sich aufgrund des Miller-Effektes so die doppelte Gate-Drain-Kapazität<sup>20</sup> ergibt. In (5.82) wurden gleiche Transkonduktanzen wie zuvor angenommen und dadurch ergeben sich bei gleichem Strom und gleichen Übersteuerungsspannungen der Transistoren gleiche Abmessungen. In (5.82) dominiert die Drain-Bulk-Kapazität, die zudem von der Drainspannung abhängt. So folgt für die Lastkapazität

$$C_L \approx 2C_{DB} + C_{GSn}. \quad (5.84)$$

Bei der idealen Skalierung [DGR<sup>+</sup>74] von MOS-Transistoren werden alle Längen, die Schwellen- und die Versorgungsspannung um einen Faktor  $\alpha > 1$  reduziert sowie die Dotierungen um den Faktor  $\alpha$  erhöht, um ein konstantes elektrisches Feld im Transistor zu erreichen. Allerdings wird diese Skalierung für neuartige Prozesse nicht mehr anhalten, wie man an Tabelle 2.2 verifizieren kann. Hier ändern sich die Spannungen um einen anderen Faktor als die Längenabmessungen. So werden im Folgenden zwei Skalierungsfaktoren

$$\alpha_U \approx 1,10 \quad \text{Skalierungsfaktor der Spannungen} \quad (5.85)$$

$$\alpha_L \approx 1,40 \quad \text{Skalierungsfaktor der Längen} \quad (5.86)$$

verwendet, die mit den Werten aus Tabelle 2.2 bestimmt wurden. Bedingt durch die zwei Skalierungsfaktoren bleibt das elektrische Feld im Bauelement nicht konstant, so dass die Vorhersagen an diesen Umstand angepasst wurden [FDN<sup>+</sup>01].

So folgt für die Lastkapazität

$$\hat{C}_L \approx \frac{C_L}{\alpha_L} \quad (5.87)$$

nach der idealen Skalierungstheorie. Jedoch lassen sich die Abmessungen des Gateoxids [Hil04] bei aktuellen Prozessen nicht mehr mit dem Skalierungsfaktor  $\alpha_L$  der Längen verringern, sondern es erfolgt eine Reduzierung mit dem Faktor  $\alpha_U$ , so dass gilt

$$\hat{C}_L \approx \frac{\alpha_U}{\alpha_L^2} C_L. \quad (5.88)$$

<sup>20</sup>Bei hohen Verstärkungen des Neurons, d.h. bei großen Steilheiten in der Übertragungsfunktion, ist die Vernachlässigung von  $C_{GD}$  nicht mehr zulässig.

**Tabelle 5.7:** Maximale Auflösung eines Neurons in analoger Schaltungstechnik in Abhängigkeit der minimalen Gatelänge

Gatelänge (nm)	130	90	65	45	32
$U_{dd}$ (V)	1,3	1,2	1,1	1,0	0,9
$C_{ox}$ (fF/ $\mu m^2$ )	15,3	16,8	18,7	19,7	20,9
SNR	$6,61 \cdot 10^5$	$2,96 \cdot 10^5$	$1,44 \cdot 10^5$	$6,03 \cdot 10^4$	$2,62 \cdot 10^4$
k (bit)	9,67	9,09	8,57	7,94	7,34
$\Delta k$ (bit)	-	0,58	0,52	0,63	0,60

Damit folgt insgesamt für das neue SNR

$$\widehat{\text{SNR}} = \frac{U_{dd}^2 g_{m1}(r_{o1} \parallel r_{o3}) C_L \alpha_U}{8 \alpha_U^2 \alpha_L^2 \gamma k_B T} \quad (5.89)$$

$$= \frac{1}{\alpha_U \alpha_L^2} \cdot \text{SNR} \quad (5.90)$$

$$\approx k - 0,5542, \quad (5.91)$$

wobei  $k$  der äquivalenten Auflösung in bit entspricht und  $\widehat{\text{SNR}}$  das Signal-Rausch-Verhältnis der skalierten Schaltung bezeichnet.

Mit der äquivalenten Auflösung [Sha48] und den Skalierungsregeln ergibt sich für diese einfache Betrachtung eine Herabsetzung der maximalen Auflösung des Neurons von 0,5542 bit pro Technologiegeneration. Hier sei darauf hingewiesen, dass es sich aufgrund der Vereinfachungen um die theoretische, obere Grenze der Auflösung handelt, die durch das thermische Rauschen bestimmt wird. Tatsächlich wird sich eine geringere Auflösung ergeben, da zum einen die Effekte aufgrund von Fehlanpassungen und den daraus resultierenden Störungen der Symmetrie oder der Gleichtaktverstärkung nicht betrachtet wurden, und zum anderen die Effekte des Rauschens bei Technologien im unteren nm-Bereich mit Hilfe von erweiterten Modellen beschrieben werden müssen [Kle99, KKT01, KWSZVD00].

In Tabelle 5.7 ist die Abhängigkeit der maximalen Auflösung eines sigmoiden Neurons aufgrund des thermischen Rauschens nach (5.80) von der Technologie dargestellt. Hier wurden für die technologischen Parameter die Werte aus [ZC06] verwendet, und als Steilheit für den Sigmoid  $A = 1$  gewählt. Der Wert des Rauschfaktors  $\gamma$  wurde konservativ zu  $2/3$  gewählt, und es wurde für die Analyse von einer Temperatur von 300 K ausgegangen. Als Lastkapazität wurde nur die Gate-Source-Kapazität der nächsten Stufe betrachtet, wobei die Eingangstransistoren ein (typisches) W/L-Verhältnis von 100 bei minimaler Gatelänge besitzen.

**Tabelle 5.8:** Maximale Auflösung bei Berücksichtigung von 1/f-Rauschen

Gatelänge (nm)	130	90	65	45	32
SNR	$3,31 \cdot 10^5$	$1,48 \cdot 10^5$	$7,20 \cdot 10^4$	$3,01 \cdot 10^4$	$1,31 \cdot 10^4$
k (bit)	9,17	8,59	8,07	7,44	6,84
$\Delta k$ (bit)	-	0,58	0,52	0,63	0,60

An Tabelle 5.7 erkennt man, dass die maximale Auflösung einer analogen Implementierung der sigmoiden Funktion durch maximal 9 bit begrenzt ist. Bei kleineren CMOS-Technologien sinkt die Genauigkeit um ca. 0,6 bit pro Prozessgeneration, so dass nur noch 7 bit bei einer 32 nm-Technologie zu erreichen sind. Damit ist eine digitale Implementierung einer analogen Variante hinsichtlich der Genauigkeit überlegen. Die Abweichungen in Tabelle 5.7 von der theoretischen Verringerung aus (5.91) resultieren aus den nicht konstanten Skalierungsfaktoren über die Technologiegenerationen.

Im Folgenden soll zusätzlich der Einfluss des Flicker-Rauschens betrachtet werden. Die Rauschleistungen am Ausgang aufgrund des thermischen Rauschens aus (5.74) und (5.76) skalieren folgendermaßen

$$P_{ti} \propto \frac{\alpha_L}{\alpha_U^2} \quad i = 1; 3, \quad (5.92)$$

wenn davon ausgegangen wird, dass die Frequenz  $f_2$  als die 3dB-Grenzfrequenz der Verstärkung und  $f_1$  sehr klein gewählt werden. Die Anteile des Flicker-Rauschens skalieren in erster Näherung mit dem gleichen Faktor. Das Flicker-Rauschen ist besonders bei tiefen Frequenzen von Bedeutung, so dass bei höheren Frequenzen der Anteil des thermischen Rauschens überwiegt [SC99]. Zur Bestimmung der Rauschleistung ist somit die obere Frequenz  $f_2$  nicht weiter von Bedeutung und kann zur Eckfrequenz<sup>21</sup>  $f_C$  gewählt werden. Die untere Frequenz  $f_1$  muss ebenfalls fest gewählt werden, da sich für  $f_1 \rightarrow 0$  eine unendlich große Rauschleistungsdichte ergeben würde. Damit bleiben die Zähler aus (5.75) und (5.77) annähernd konstant, so dass sich die gleiche Skalierung wie beim thermischen Rauschen ergibt. Als Folge daraus reduziert sich die Auflösung eines Neurons ebenfalls um 0,5542 bit pro Technologiegeneration.

Tabelle 5.8 zeigt die maximale Auflösung unter Berücksichtigung von 1/f-Rauschen. Dabei wurde für den Widerstand der Parallelschaltung der Drainwiderstände ein Wert von 500  $\Omega$  angenommen, so dass ungefähr ein Drainstrom von 0,5 mA durch die Transistoren  $M1$  und  $M3$  fließt. Als untere Frequenz wurden 10 Hz festgelegt. Man

<sup>21</sup>Bei der Eckfrequenz  $f_C$  sind die Spektralanteile aufgrund des thermischen Rauschens und des Flicker-Rauschens gleich groß.

erkennt, dass unter diesen Annahmen die maximale Auflösung um 0,5 bit – 0,6 bit herabgesetzt wird und die theoretisch ermittelte Skalierung der Auflösung mit der Technologiegeneration eingehalten wird.

Die theoretischen Ergebnisse wurden durch Simulationen mit Hilfe des *BSIM4*-Modells [LCJH00] unter Spice verifiziert. Als Simulationsumgebung wurde dabei HSpice von Synopsys verwendet, und die entsprechenden BSIM-Modelle für die unterschiedlichen Technologiegenerationen wurden [ZC06] entnommen. Bei dem Entwurf des Differenzverstärkers wurde zuerst der notwendige Strom bestimmt, um die abgeschätzte Lastkapazität innerhalb von 0,1 ns aufzuladen. Mit Hilfe des Stromes und der Anforderung nach einem Spannungshub von  $\Delta U = U_{dd}$  der Ausgangsspannung wurden die Dimensionen der Transistoren aus Abbildung 5.7 festgelegt.

Von den Differenzverstärkern wurde jeweils die Spannungsverstärkung im Arbeitspunkt sowie die Rauschleistung am Ausgang bestimmt. Die Ergebnisse dieser Simulationen bzw. die Dimensionierung der Transistoren finden sich in Tabelle 5.9.

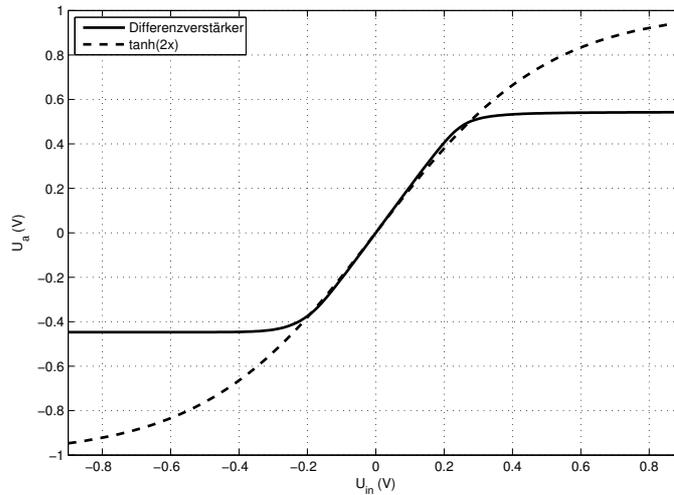
**Tabelle 5.9:** Simulationsergebnisse des analogen Neurons

Gatelänge (nm)	130	90	65	45	32
U <sub>dd</sub> (V)	1,3	1,2	1,1	1,0	0,9
W <sub>i</sub> (μm)	7,22	2,66	1,03	0,53	0,18
W <sub>n</sub> (μm)	10,03	3,70	3,21	1,06	0,36
W <sub>p</sub> (μm)	9,03	5,20	2,01	0,66	0,40
I <sub>d</sub> (μA)	920	368	167	48	17
Fläche (μm <sup>2</sup> )	5,89	1,84	0,74	0,18	0,05
Leistung (mW)	1,2038	0,4416	0,1837	0,0480	0,0153
Verstärkung A	6,76	4,82	4,35	3,17	2,08
SNR (Ausgang)	4,99 · 10 <sup>6</sup>	1,70 · 10 <sup>6</sup>	7,95 · 10 <sup>5</sup>	1,82 · 10 <sup>5</sup>	4,46 · 10 <sup>4</sup>
SNR (Eingang) <sup>1</sup>	2,50 · 10 <sup>6</sup>	8,48 · 10 <sup>5</sup>	3,98 · 10 <sup>5</sup>	9,10 · 10 <sup>4</sup>	2,23 · 10 <sup>4</sup>
k Theorie <sup>2</sup> (bit)	10,54	9,72	9,13	8,27	7,37
k (bit)	11,13	10,35	9,80	8,74	7,72
Δk (bit)	-	0,78	0,55	1,06	1,01

<sup>1</sup> Mit dem SNR am Eingang können die unterschiedlichen Entwürfe miteinander verglichen werden, da die (unterschiedliche) Verstärkung für das Verhältnis am Eingang nicht relevant ist.

<sup>2</sup> In Tabelle 5.8 wurde von einer Verstärkung  $A = 1$  ausgegangen, so dass die Berechnung für die Auflösung mit der gemessenen Verstärkung durchgeführt wurde.

Wie man im Vergleich der Tabellen 5.8 und 5.9 feststellen kann, ergibt sich bei der Simulation mittels HSpice ein besseres SNR und damit eine höhere Auflösung als bei den theoretischen Ergebnissen. Dieser Vorteil zugunsten der Simulation ist darin begründet, dass bei den theoretischen Berechnungen  $g_{m1} = g_{m3}$  angenommen wurde. Dies ist jedoch bei dem Entwurf des Differenzverstärkers nicht eingehalten



**Abbildung 5.12:** Übertragungsfunktion des Differenzverstärkers in 32 nm CMOS

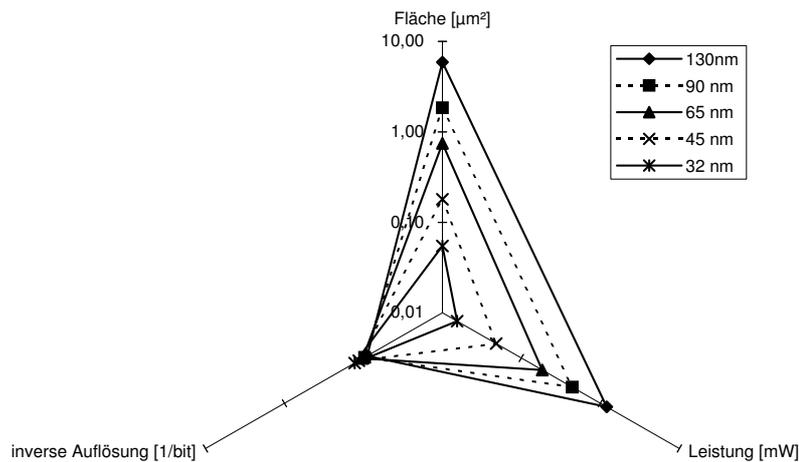
worden, sondern es wurde eine größere Transkonduktanz für die Transistoren  $M1$  und  $M2$  als für die p-Kanal-Transistoren verwendet. Dadurch ergibt sich ein besseres Signal-Rausch-Verhältnis bei der HSpice-Simulation gegenüber den Berechnungen.

Ebenfalls zeigen die Simulationen, dass bei der Verwendung von einem analogen Neuron dies in seiner Auflösung durch 11 bit begrenzt wird. Mit Skalierung der Technologie nimmt diese maximale Auflösung mit mehr als 0,5 bit pro Technologie ab, so dass sich in einer 32nm-Technologie nur noch 7 bit als maximale Auflösung erreichen lassen. Allerdings kann als konservativer Richtwert mit einem Abfall von einem bit pro Technologiegeneration gerechnet werden. Dass dieser Abfall von den theoretischen Betrachtungen abweicht, liegt auch hier an den geänderten Voraussetzungen. Die Randbedingung  $g_{m1} = g_{m3}$  ist bei dem Entwurf nicht erfüllt, und auch die Transistoren lassen sich bei diesen Strukturgrößen nicht mehr mit den einfachen Gleichungen, z.B. aus [Raz00], beschreiben, sondern es ergeben sich komplexere Zusammenhänge [Raz00].

Abbildung 5.12 zeigt beispielhaft die Übertragungsfunktion des analogen Neurons in einer CMOS-Technologie mit 32 nm Strukturgröße zusammen mit der Funktion  $y = \tanh(2x)$ . Aufgrund des begrenzten Signalhubs ergibt sich ein kleiner Aussteuerungsbereich, so dass eine Skalierung der Daten vorgenommen werden muss. Es zeigt sich eine gute Übereinstimmung im linearen Bereich.

In Abbildung 5.13 sind die Entwürfe in den verschiedenen Technologien miteinander hinsichtlich ihrer Eigenschaften Leistung, Fläche und maximale Auflösung<sup>22</sup> verglichen worden. Für einen optimalen Punkt im Entwurfsraum liegen die jeweiligen Koordinatenschnittpunkte näher an dem Ursprung. So wird deutlich, dass die

<sup>22</sup> Zur Darstellung der Kostenfunktion wurde der Kehrwert der Auflösung verwendet, so dass auch hier ein Punkt näher dem Ursprung bessere Eigenschaften besitzt.



**Abbildung 5.13:** Vergleich des analogen Entwurfes in verschiedenen Technologien

Auswirkungen der Technologie auf die Leistung und die Fläche wesentlich positiver als auf das SNR ausfallen. Die möglichen Auflösungen eines Neurons verringern sich deshalb mit den Technologiegenerationen. Für ein analoges Neuron bedeutet dies, dass adaptive Verfahren, wie z.B. inhärente Lernverfahren, zum Einsatz in diesen Technologien kommen müssen. Bedingt durch das Lernverfahren lassen sich die Abnahmen in der Genauigkeit kompensieren. Auf der anderen Seite wirken sich die neuen Technologieprozesse positiv auf die Leistung und die Fläche aus, so dass bei annähernd gleicher Genauigkeit wesentlich kleinere Netze mit geringerer Leistung umgesetzt werden können. So stellen die analogen Netze eine effiziente Architektur im Hinblick auf Fläche, Leistung und Auflösungsvermögen für zukünftige Technologien dar, wobei die Notwendigkeit inhärenter Lernverfahren besteht (vgl. Abbildung 5.13), da die maximale Auflösung abnimmt.

Daneben sollen die Auswirkungen von Parameterschwankungen bei der Fertigung auf die Übertragungsfunktion des analogen Neurons betrachtet werden. Für diesen Zweck werden die geometrischen Dimensionen der Transistoren, wie die Weite und die Länge, aus Abbildung 5.7 innerhalb der Fertigungsungenauigkeiten verändert und so der Einfluss auf die Leistung, die Verstärkung und die Rauschspannung am Ausgang betrachtet. Für die Schwankungen der geometrischen Abmessungen werden die Daten aus [Sem05] verwendet, wobei die Fertigungsungenauigkeiten z.B. aufgrund der Lithografie- und Ätztechniken zustande kommen [Hil04]. Diese Schwankungen bewegen sich ungefähr im Bereich von 10 % der minimalen Gatelänge der jeweiligen Technologie.

Neben diesen Schwankungen tritt noch eine Vielzahl von weiteren Parametervariationen bei der Fertigung auf, die hier jedoch nicht betrachtet werden, so dass die Ergebnisse eine konservative Untersuchung erlauben. Neben den Abmessungen des Kanals unterliegen auch die Oxiddicke, das Dotierungsprofil, etc. statistischen Schwankungen und beeinflussen so die elektrischen Parameter des Transistors. So

**Tabelle 5.10:** Ergebnisse der Parametervariation mit Hilfe der Monte-Carlo-Simulation

Gatelänge	130 nm	90 nm	65 nm	45 nm	32 nm
	Erwartungswert				
$U_{na}$ (mV)	0,9894	1,0870	1,1938	1,6244	1,9499
Verstärkung	6,7489	4,5472	4,3798	3,2042	2,1123
Leistung (mW)	1,2145	0,4486	0,1859	0,0493	0,0159
	Varianz				
$U_{na}$ (mV)	$1,43 \cdot 10^{-2}$	$2,47 \cdot 10^{-2}$	$1,51 \cdot 10^{-2}$	$2,03 \cdot 10^{-2}$	$2,93 \cdot 10^{-2}$
Verstärkung	$7,78 \cdot 10^{-1}$	$5,22 \cdot 10^{-1}$	$2,47 \cdot 10^{-1}$	$1,01 \cdot 10^{-1}$	$3,73 \cdot 10^{-2}$
Leistung (mW)	$5,81 \cdot 10^{-3}$	$1,00 \cdot 10^{-3}$	$2,38 \cdot 10^{-4}$	$3,98 \cdot 10^{-5}$	$7,54 \cdot 10^{-6}$

wirken sich diese Phänomene ebenfalls auf die Arbeitsweise des Neurons aus. Weiter sind die eingepprägten Spannungen aus Abbildungen 5.7 als ideale Spannungsquellen angenommen worden. Allerdings müssen diese mittels spezieller Schaltungen erzeugt werden, so dass sich hier zusätzliche Auswirkungen auf die Architektur ergeben.

Tabelle 5.10 zeigt die Ergebnisse einer Monte-Carlo-Simulation für die Technologiegenerationen, wobei die Dimensionen des Kanals (Weite und Länge) durch eine Gaußverteilung modelliert wurden. Für den Mittelwert wurden die Abmessungen aus Tabelle 5.9 verwendet und für die Standardabweichungen<sup>23</sup> die Daten aus [Sem05].

Wie zuvor kann man an Tabelle 5.10 die schlechteren SNR-Eigenschaften mit den Technologiegenerationen beobachten, da die Rauschspannung am Ausgang ansteigt und gleichzeitig die maximale Leistung am Eingang abnimmt. Zusätzlich muss dabei beachtet werden, dass die Verstärkung über die Prozesstechnologien abnimmt. So besitzen die Technologien mit kleinerer Strukturgröße schlechtere Eigenschaften gegenüber Rauschen, wenn das SNR am Eingang des Differenzverstärkers betrachtet wird. Daneben stimmt die erwartete Steigung des Neurons in erster Näherung gut mit der zuvor bestimmten Verstärkung (vgl. Tabelle 5.7) überein. Der Zusammenhang der Kleinsignalverstärkung und den Dimensionen stellt einen nichtlinearen Zusammenhang dar, so dass in erster Näherung für den Erwartungswert sich der Funktionswert der Erwartungswerte der Abmessungen ergibt [BSM05]. Gleiches gilt für den Zusammenhang der Verstärkung mit den Varianzen. Da die Varianzen mit den Abmessungen betragsmäßig kleiner werden, verringern sich auch die Auswirkungen auf die Eigenschaften des Neurons.

Abbildung 5.14 zeigt die Histogramme der Leistung eines Neurons in den verschiedenen Technologien, welche zur Bestimmung der Erwartungswerte und Varianzen

<sup>23</sup>Die Daten aus dem Fahrplan beschreiben insgesamt den dreifachen Wert der Varianz.

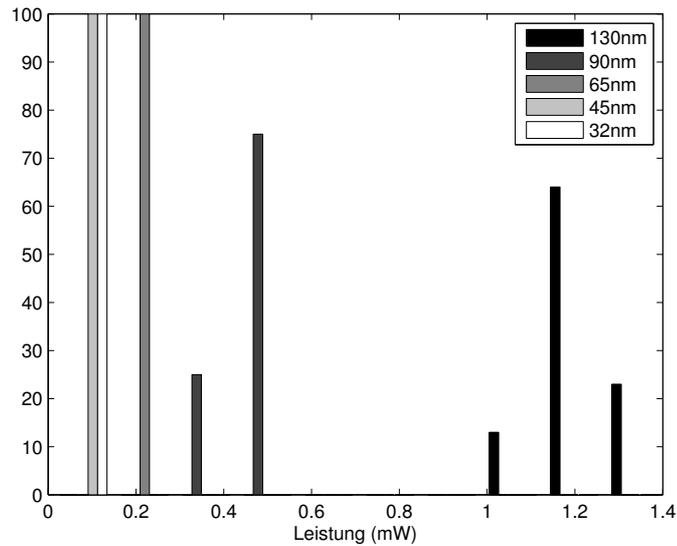


Abbildung 5.14: Häufigkeit der Leistung

aus Tabelle 5.10 genutzt wurden. Hier erkennt man die geringere Varianz für die kleineren Strukturgrößen, was mit der geringeren Varianz in den Abmessungen und dem gleichen mathematischen Zusammenhang zu erklären ist. Deutlich wird hier nochmals der Vorteil kleinerer Strukturgrößen hinsichtlich der Leistung des Neurons. Diese sinkt mit den Generationen stark ab, so dass deutlich leistungsärmere Schaltungen entwickelt werden können, was dem Vorbild des menschlichen Gehirns entgegenkommt.

In Abbildung 5.15 sind die Ergebnisse der Monte-Carlo-Simulation für die Steilheit des Neurons dargestellt. An den Verteilungen wird deutlich, dass für eine hohe Genauigkeit (geringe Abweichung der Verstärkung) ein intrinsisches Lernverfahren benötigt wird, da nicht von einer gleichen Verstärkung für alle Neuronen ausgegangen werden kann. So besitzen diese unterschiedliche Steilheiten, auf die ein integriertes Training reagieren und anpassen kann, um eine hohe Approximationsgüte dennoch zu erreichen. Aufgrund der Schwankungen in der Verstärkung und des begrenzten Ausgangsspannungshubs von  $U_{dd}/2$  ist die Funktionalität eines analogen Neurons bei der Approximation in zukünftigen Technologien beschränkt, da eine exakte Approximation von Funktionen in diesen nur noch schwer möglich ist. Vielmehr ist der Einsatz des analogen Neurons auf eine nichtlineare Transformation des Einganges in den Bildraum beschränkt, wo dann eine lineare Trennung der Daten vorgenommen werden kann [Duc06]. So können diese Verarbeitungskonzepte zur Darstellung boolescher Funktionen verwendet werden [SGJF06]. Die Approximation einer kontinuierlichen Funktion ist aufgrund der Schwankungen in den Parametern in den Approximationseigenschaften begrenzt und benötigt entweder ein adapti-

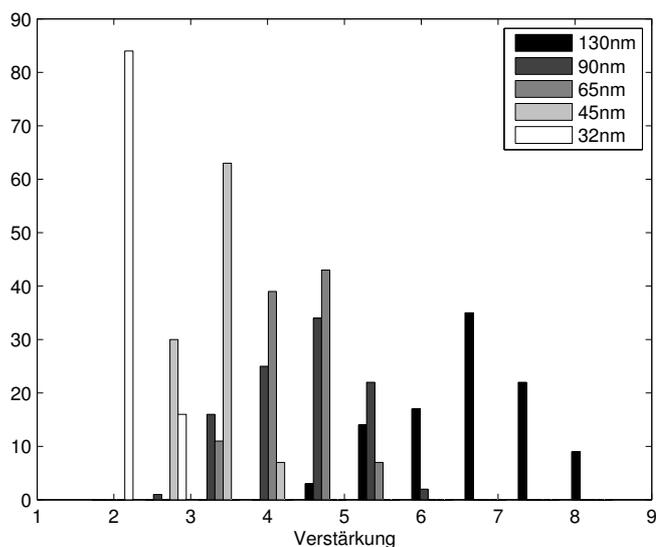


Abbildung 5.15: Häufigkeit der Verstärkung

ves Lernverfahren oder basiert auf der Approximation durch Heaviside-Funktionen [HZM<sup>+</sup>06].

Wie schon angedeutet, enthalten die Untersuchungen zu den Parametervariationen der Technologien nur die statistischen Schwankungen der geometrischen Dimensionen eines einzelnen Transistors. Daneben tritt eine Vielzahl von weiteren Schwankungen bei der Fertigung eines Transistors auf, die z.B. direkten Einfluss auf die Steilheit oder die Schwellenspannung haben. Als Resultat ergeben sich zusätzliche Beeinflussungen der Auflösung und der Zuverlässigkeit eines analogen Neurons. Weiter handelt es sich bei den Daten aus dem Fahrplan [Sem05] um Fertigungsgenauigkeiten, die für einen zuverlässigen Betrieb erreicht werden müssen. Falls diese verfehlt werden, besitzen die gefertigten Systeme eine geringere Zuverlässigkeit oder es ergeben sich größere Variationen, so dass die Architekturen unter diesen Randbedingungen arbeiten müssen. Hiervon ist auszugehen, da Lösungsmöglichkeiten für das Erreichen solcher Fertigungsgenauigkeiten teilweise oder gar nicht bekannt sind [Sem05].

### 5.3.4 Vergleich der digitalen und analogen Realisierungen

Weiter soll die Auflösung des analogen Neurons mit einer digitalen Implementierung verglichen werden, um Aussagen über die Eignung der Entwurfsart hinsichtlich zukünftiger Technologien zu erhalten. Tabelle 5.11 zeigt den Vergleich zwischen einem digitalen und einem analogen Neuron hinsichtlich der maximalen Auflösung, seiner Leistung und Fläche. Die Übertragungsfunktion des analogen Neurons wird durch die tanh-Funktion beschrieben, während für die digitale Variante die logistische

**Tabelle 5.11:** Vergleich des analogen und digitalen Entwurfs eines Neurons

Gatelänge (nm)	Fläche ( $\mu\text{m}^2$ )		Leistung (mW)		Auflösung (bit)	
	analog	digital	analog	digital	analog	digital
130	5,89	7140,89	1,2038	3,2946	11	5
90	1,84	3422,56	0,4416	1,4576	10	5
65	0,74	1785,22	0,1837	0,6969	9	5
45	0,18	855,64	0,0480	0,3037	8	5
32	0,05	432,68	0,0153	0,1382	7	5

Funktion verwendet wird. Diese Funktion wird durch das Verfahren aus [BPVL94] umgesetzt und kann mit Hilfe der SIRENS-Struktur und einem Erweiterungsmodul umgesetzt werden. Nach Tabelle 5.5 ergibt sich ein maximaler absoluter Fehler von 0,0249 bei einer Verwendung von 8 bit für die Bruchdarstellung. Dies entspricht einem maximalen Quantisierungsfehler (bei Nutzung eines Speichers) bei einer Auflösung von  $k = 5$  bit. Alle größeren Wortbreiten für die Bruchdarstellung erreichen aufgrund der Art der Approximation die gleiche maximale Auflösung, so dass zu einem Vergleich mit der analogen Variante die kleinste digitale Implementierung mit 8 bit für die Bruchdarstellung verwendet wird. Für die Abbildung wurde ebenfalls die Technologie von STMicroelectronics (vgl. Abschnitt 5.3.2) verwendet, und die Leistung wurde bei einer Frequenz von  $f = 500$  MHz ermittelt.

**Bemerkung.** Für den Vergleich kann die SIRENS-Struktur mit Erweiterungsmodul genutzt werden. Für die logistische Funktion als Übertragungsfunktion werden ein Erweiterungsmodul und ein Basismodul benötigt, was zu einer hohen Fläche führt. Deswegen wurde für den Vergleich ein angepasster Entwurf nach [BPVL94] erstellt. Dieser enthält neben dem Datenpfad Register zur Speicherung des Ergebnisses.

Um beide Entwürfe in zukünftigen Technologien vergleichen zu können, wurden die Ergebnisse in Tabelle 5.11 des digitalen Neurons auf diese Prozessgenerationen mit Hilfe der Skalierungsregeln projiziert. Für die Skalierung der Leistung wurde von einer konstanten Frequenz ausgegangen und von der gleichen Anzahl von Gattern zur Darstellung der Logikfunktion.

Wie man an Tabelle 5.11 erkennen kann, ist die analoge Implementierung der genutzten digitalen Variante in allen Punkten überlegen. Der kleinere Flächenanteil ergibt sich aufgrund der benötigten Multiplikation bei der digitalen Übertragungsfunktion. So nimmt diese wesentlich mehr Fläche ein als die fünf Transistoren des Differenzverstärkers, und damit ist das analoge Neuron in jeder Technologie hinsichtlich der Fläche seiner digitalen Variante überlegen. Dadurch kommt ebenfalls die höhere Leistung der digitalen Implementierung zustande, da zum einen insgesamt mehr Transistoren zu den Leckströmen beitragen und zum anderen mehr

Knoten (Kapazitäten) umgeladen werden müssen. Allerdings entsteht bei dem digitalen Entwurf in CMOS-Technologie kein statischer Querstrom, wie er bei dem Differenzverstärker auftritt. So fällt beim digitalen System nur eine geringe statische Leistung gegenüber dem Analogen an, und dadurch fällt der Unterschied bei der Leistung nicht so deutlich aus wie bei der Fläche.

Der deutliche Vorteil der digitalen Implementierung liegt in der konstanten Auflösung, die nicht oder nur kaum durch die Technologie gestört wird. So sinkt bei dem analogen System die Auflösung mit einem bit pro Technologiegeneration aufgrund des schlechteren Signal-Rausch-Verhältnisses. Bei einem digitalen System bleibt dagegen die Auflösung unter der Randbedingung eines zuverlässigen Entwurfes konstant. Eine Abnahme der effektiven Auflösung ergibt sich durch eine Wahrscheinlichkeit  $p$  von Fehlern in den binären Stellen bei zukünftigen Technologien.

Die geringe Auflösung des digitalen Neurons ist durch die Art der Approximation begrenzt, da diese den maximalen Fehler durch eine höhere Auflösung nicht weiter reduzieren kann. Durch eine Wertetabelle, realisiert als Speicher oder eine boolesche Funktion, kann die Genauigkeit des digitalen Systems jedoch beliebig vergrößert werden. Die begrenzenden Faktoren sind dabei lediglich die Fläche und die Leistung.

**Tabelle 5.12:** Prognosen für ein Neuron auf Basis von Speicher und boolescher Funktionen

Techn. (nm)	Auflösung (bit)			Fläche ( $\mu\text{m}^2$ )		
	Sigmoid	Speicher	boolesche Funktion	Sigmoid	Speicher <sup>1</sup>	boolesche Funktion
130	4	11	11	7140,89	3632,20	227217,41
90	4	10	10	3422,56	1740,88	108903,02
65	4	9	9	1785,22	908,05	56804,35
45	4	8	8	855,64	435,22	27225,75
32	4	7	7	432,68	220,08	13767,49

<sup>1</sup> Die Größe des Speichers wurde anhand der Daten aus [Gla99] sowie den Skalierungsregeln bestimmt.

Tabelle 5.12 ermöglicht einen Vergleich der digitalen Implementierungsformen<sup>24</sup>. Hier wurden ebenfalls die Skalierungsregeln angewendet, um Aussagen über künftige Prozesse treffen zu können. Für die Realisierung mittels boolescher Funktionen kann für die Komplexität gezeigt werden, dass

$$\mathcal{O}(2^k) \quad (5.93)$$

Gatter für eine Funktion benötigt werden [Weg91]. Dabei kann diese obere Grenze noch weiter verbessert werden, wenn neben UND bzw. ODER-Gattern auch ande-

<sup>24</sup>Dabei bezeichnet der Sigmoid das System nach [BPVL94].

re Gatter, z.B. XOR-Gatter, zugelassen werden. Bei insgesamt  $k$  Funktionen zur Bestimmung des Ausgangs des Neurons kann dann mit Hilfe der Fläche für UND bzw. ODER-Gatter die Gesamtfläche bestimmt werden. Dabei stellt die so bestimmte Fläche eine konservative obere Schranke dar, so dass die tatsächliche Fläche darunter liegen wird. Zum einen gilt die Schranke aus (5.93) für nichtlinear separierbare Funktionen, wobei deren Anteil mit steigender Auflösung sinkt [Duc06]. Zum anderen besitzt die STMicroelectronics-Technologie auch Gatter mit mehr als zwei Eingängen, so dass diese Anzahl reduziert werden kann.

**Beispiel.** Der Funktionswert der Aktivierung eines Neurons und der Eingang  $x$  sollen mit jeweils 5 bit aufgelöst werden. Dafür wird ein Speicher mit einer Größe von 20 byte benötigt bzw. fünf boolesche Funktionen, die jeweils eine Abbildung  $[0; 1]^5 \rightarrow [0; 1]$  vornehmen. Mit Hilfe dieser Repräsentation kann ein einzelnes Neuron und dadurch ein komplettes Netz abgebildet werden. Jedoch kann bei der Kenntnis der zu approximierenden Funktion, z.B. der si-Funktion, diese direkt in dem Speicher abgelegt bzw. durch die booleschen Funktionen dargestellt werden. So stellen die Abbildungen durch Speicher oder boolesche Funktionen eine Entartung der neuronalen Prinzipien dar, welche auf der Umsetzung in Hardware basiert.

Ein Vergleich der drei digitalen Varianten untereinander zeigte, dass die Verwendung von Speicher die geringste Fläche bei gleicher Auflösung benötigt. Der Einsatz von booleschen Funktionen hat die Fläche um eine Größenordnung erhöht, während der Einsatz einer Recheneinheit als Neuron die Fläche um zwei Größenordnungen anwachsen ließ<sup>25</sup>. Deshalb werden in Tabelle 5.12 die Auflösungen des analogen Systems für diese beiden digitalen Formen verwendet.

Auch gegenüber diesen beiden Systemen besitzt die analoge Implementierung deutliche Vorteile hinsichtlich der Fläche, wie man an dem Vergleich der Tabellen 5.11 und 5.12 feststellen kann. So liegt diese mehrere Größenordnungen unterhalb des Flächenbedarfs der digitalen Systeme. Wird davon ausgegangen, dass die Leistung annähernd proportional zu der Fläche eines Systems ist, ergibt sich hier ein ähnliches Bild, wenn man die Leistungen des digitalen Systems aus Tabelle 5.11 als Referenz nimmt.

Zusammenfassend bleibt festzuhalten, dass eine analoge Implementierung einer sigmoiden Übertragungsfunktion einem digitalen Entwurf hinsichtlich Flächenbedarf und Leistung klar überlegen ist. Eine Betrachtung der Geschwindigkeit wurde allerdings nicht durchgeführt. Ebenso ist die digitale Implementierung nach [BPVL94] ungeeignet, da sich aufgrund der Approximationsart eine untere Grenze des absoluten Fehlers ergibt. Deswegen kann dieses System bei dem Vergleich mit anderen Möglichkeiten, der Realisierung durch Speicher oder boolesche Funktionen, hinsichtlich der Fläche und Approximationsgüte nicht mithalten. Allerdings kommt jenes

<sup>25</sup> Geht man bei der Leistung der Systeme von einem proportionalen Zusammenhang zwischen Leistung und Fläche aus, so würde sich hier ein ähnliches Bild ergeben.

System den neuronalen Prinzipien nahe, wo ein Neuron als ein elementares Rechenelement mit integriertem Speicher angesehen werden kann. Im Falle des Speichers würde dieses System in ein Extremum getrieben, da ein Neuron nur aus einem Gedächtnis bestünde. Eine Realisierung mittels boolescher Funktionen dagegen führt zu einem flächenmäßig kleineren und hochauflösenden System mit neuronalen Eigenschaften als eigenständiges Rechenelement.

## 5.4 Zusammenfassung und Diskussion

In diesem Kapitel wurden die Auswirkungen einer Umsetzung neuronaler Netze in digitale und analoge Schaltungstechnik betrachtet. Bei der Umsetzung der neuronalen Verfahren in Hardware entstehen aufgrund begrenzter Auflösung der Signale, durch überlagertes Rauschen, begrenzten Speicherplatz etc. Fehler, die mit Hilfe der numerischen Mathematik analysiert wurden. So kann eine Aussage über die Umsetzung der radialen Basisfunktionen getroffen werden, mit welchem mathematischen Verfahren diese berechnet werden sollen. Direkt auf die Umsetzung des LCNNs wurde in dieser Arbeit nicht eingegangen, da eine analoge Umsetzung dieses Netzes existiert [SKR98] und mit Hilfe des in Abschnitt 5.2.1.3 vorgestellten Sigmoids eine digitale Variante erstellt werden kann.

Neben der idealen Umsetzung der Berechnung der Gaußfunktion können mit Hilfe der Numerik Aussagen über die Parameter des Netzes getroffen werden, die zu den Ergebnissen aus Kapitel 3 und Kapitel 4 aufgrund ähnlicher Beschreibungen konsistent sind. Für das Erreichen numerisch stabiler Berechnungen müssen die Parameter des Netzes beschränkt werden, um zuverlässig arbeiten zu können. Daneben konnte das Verfahren mit dem kleineren Fehler zur Berechnung einer multivariaten Gaußfunktion gefunden werden, indem die Fortpflanzungen der Fehler für die beiden Algorithmen betrachtet wurden.

Weiter wurde für die Umsetzung neuronaler Verfahren eine parallele, rekonfigurierbare, digitale Struktur entwickelt, die verschiedene Modelle von Neuronen abbilden kann. Diese basiert auf der Approximation der Aktivierungsfunktion eines Neurons mit Hilfe der Polynomapproximation. Mit den numerischen Betrachtungen aus diesem Kapitel kann für die Approximation der Aktivierungsfunktion ein geeignetes Polynom gefunden werden, um ein flächeneffizientes System und eine genaue Approximation zu erhalten.

Die Simulationsergebnisse des Systems zeigten, dass diese rekonfigurierbare Struktur verschiedene Neuronenmodelle darstellen kann. Dabei konnten nicht nur statische Modelle, sondern auch dynamische Modelle abgebildet werden. Die Fläche des Systems konnte reduziert werden, indem die homogene Struktur durch ein spezielles Modul erweitert wird. Dieses benötigt bei gleicher Genauigkeit einen kleineren Bedarf an Fläche, ist jedoch nur für bestimmte Modelle von Neuronen einsetzbar.

Das so entstandene System wurde hinsichtlich der Faktoren Fläche, Leistung und Geschwindigkeit untersucht und bewertet.

Eine interessante Fragestellung stellt die Limitierung der rekonfigurierbaren Struktur auf binäre Werte dar. In diesem Fall werden durch das System boolesche Funktionen approximiert, und es ergeben sich neue Konzepte für die Umsetzung boolescher Funktionen, wie z.B. in [SGJF06]. Weitere Untersuchungen sind hier hinsichtlich der Separierbarkeit [Duc06] und der automatisierten Umsetzung bekannter Methoden zur Sicherstellung der Fehlertoleranz [HJ03a, RB05] möglich.

Weiter wurden in diesem Kapitel die Auswirkungen des Fortschrittes in der CMOS-Technologie auf analoge und digitale, neuronale Netze untersucht und bewertet. Anhand von Simulationsmodellen und der Skalierungstheorie konnten Aussagen über die Entwicklung der Systemeigenschaften, wie Auflösung, Fläche und Leistung getroffen werden. Bei diesem Vergleich zeigte sich die deutliche Dominanz der analogen Schaltungstechnik über einen digitalen Entwurf. Es können deutlich kleinere Systeme entworfen werden, deren elektrische Leistung wesentlich geringer als des digitalen Systems ist. Die negativen Effekte aufgrund schlechterer Skalierung des analogen Entwurfes hinsichtlich der maximal erreichbaren Auflösung fallen aufgrund eines ausreichenden Spielraumes nicht so gravierend aus. Vielmehr lassen sich durch den Einsatz neuronaler Verfahren und intrinsischer Trainingsverfahren adaptive, analoge Schaltungen entwerfen, welche widerstandsfähig gegenüber Störungen sind. Für dieses Ziel ist ein integriertes Trainingsverfahren unablässig. Daneben können die analogen Netze zur Umsetzung boolescher Funktionen eingesetzt werden [SGJF06] und hierfür neue Konzepte bereitstellen.



# Zusammenfassung

Neuronale Prinzipien stellen aufgrund der beobachteten Eigenschaften aus der Biologie eine interessante Möglichkeit dar, neuartige Architekturen für technische Systeme zu entwickeln. Durch die Übernahme dieser Verfahren erhofft man sich, die positiven Eigenschaften der biologischen Vorbilder, wie z.B. Fehlertoleranz, Robustheit oder niedriger Energiebedarf, zu übernehmen. Mit der zunehmenden Miniaturisierung der Halbleiterherstellung und dem Einzug neuartiger Materialien und Bauelemente steigt die Notwendigkeit nach adaptiven Architekturen, um so die Kosten und den Aufwand für das Testen zu begrenzen oder den Entwurf komplexer Systeme zu unterstützen.

Mit dieser Arbeit wurde die Fragestellung untersucht, ob neuronale Verfahren als mögliche Architektur zur Handhabbarkeit dieser Systeme dienen können. Dabei wurde im Speziellen die Robustheit und Fehlertoleranz von Basisfunktionsnetzwerken und des Local Cluster Neural Networks im Hinblick auf die Approximation einer Funktion analysiert. Besonders für den speziellen Fall boolescher Funktionen ist dieses Anwendungsgebiet der neuronalen Netze für die Elektrotechnik interessant, da sich so neuartige Konzepte den Logikentwurf ergeben.

Die Untersuchung dieser Fragestellung gliederte sich dabei in drei Teilgebiete. Zuerst wurden für die Analyse der Begriffe Robustheit und Fehlertoleranz mathematische Entsprechungen eingeführt. Mit Hilfe dieser Problembeschreibungen ließen sich mit mathematischen Werkzeugen Aussagen über diese beiden Eigenschaften der Netze treffen. Die Betrachtungen zur Robustheit konnten über die Eigenschaft der gleichgradigen Stetigkeit des neuronalen Netzes geführt werden. Dabei zeigte sich, dass die Netze nur unter der Randbedingung beschränkter Parameter und Eingänge eine gleichgradig stetige Funktionsmenge beschreiben. Die Konsequenz daraus bedeutete, dass künstliche neuronale Netze integriert in Hardware diese Eigenschaft besitzen und somit jede gewünschte Robustheit erreicht werden kann. Die Beschränkung der Parameter ergibt sich bei einem technischen System durch die begrenzte Kapazität des Speichers sowie die begrenzte Auflösung der Signale. Die hier erzielten Abschätzungen können zu einem Vergleich verschiedener Netztopologien und ihrer garantierten Auflösungen verwendet werden. Mit Hilfe der Vapnik-Chervonenkis-Dimension und der garantierten Auflösung lassen sich große Netze mit wenigen Eingängen und kleine Netze mit hoher Eingangsdimension in den Eigenschaften Kapazität des Netzes und Robustheit gegenüber Rauschen vergleichen. Für die Fehlertoleranz wurden

die Auswirkungen von Defekten und transienten Fehlern betrachtet. Dabei ließ sich die Ausbeute für ein neuronales Netz über die Poissonverteilung bestimmen und der Effekt transienter Fehler durch einen Markov-Prozess beschreiben. Dadurch konnten einfache Verfahren zur Verbesserung der Fehlertoleranz bewertet und die Effektivität der Verfahren in Abhängigkeit der Randbedingungen, vorgegeben durch die Technologie, bestimmt werden. Auch hier zeigte sich, dass neuronale Netze nur bei beschränkten Parametern fehlertolerant sind.

Danach wurden Verfahren vorgestellt, mit denen sich beide Eigenschaften für die neuronalen Netze verbessern lassen. Hierfür wurden Maße entwickelt, die eine Identifikation von rauschanfälligen Neuronen und Parametern sowie kritischen Neuronen bei einem Ausfall erlauben. So ist es möglich, Verfahren zur Erhöhung der Redundanz an den notwendigen Stellen im Netz einzusetzen. Zusätzlich ließen sich die umgekehrten Effekte der Gütekriterien nutzen, womit die Komplexität der Netztopologie erfasst und kontrolliert werden konnte. Weiter konnten diese Mechanismen genutzt werden, neben einer Topologieveränderung des Netzes durch geschickte Parameterwahl die Eigenschaften des Netzes zu verbessern. Dazu wurden diese innerhalb eines zusätzlichen Trainingsverfahrens, einer Mehrzieloptimierung, als zusätzliches Ziel verwendet. Wie in Simulationen gezeigt wurde, lassen sich deutlich robustere und fehlertolerantere Netze erreichen. Dieses Verfahren arbeitet unabhängig von der Netztopologie und dem verwendeten Trainingsverfahren des neuronalen Netzes, so dass es für eine Vielzahl von Anwendungen einsetzbar ist.

Als Abschluss wurden die schaltungstechnische Realisierung neuronaler Prinzipien und deren Skalierung mit den Technologiegenerationen untersucht. Dafür wurde eine parallele, rekonfigurierbare, digitale Struktur entwickelt, mit der sich verschiedene Neuronenmodelle, statische und dynamische Modelle, und unterschiedliche Topologien, vorwärtsgerichtete und rückgekoppelte Netze, abbilden lassen. So kann die Wiederverwendbarkeit des Systems aufgrund der Möglichkeit der Rekonfiguration und des Einsatzes neuronaler Verfahren maximiert werden. Dieses System wurde hinsichtlich seiner Approximationseigenschaften sowie der benötigten Fläche und Leistung analysiert. Interessant ist dabei besonders der Fall einer binären Auflösung, da sich so zum einen ein minimales System mit geringer Fläche und hoher Geschwindigkeit ergibt. Dynamische und statische Modelle von Neuronen können durch eine bit-serielle Berechnung dargestellt werden. Zum anderen werden dann boolesche Funktionen approximiert, wobei dieses System analog zu gängigen Verfahren der Erhöhung der Fehlertoleranz [HJ03a, RB05] arbeitet. Weiter wurden digitale und analoge Neuronen auf ihre Skalierbarkeit mit den Technologiegenerationen untersucht. Die analoge Implementierung eines Neurons ist seiner digitalen Form in den Punkten Leistung und Fläche überlegen und stellt in diesen Punkten eine bessere Architektur zur Umsetzung neuronaler Prinzipien dar. Die maximale Auflösung eines analogen Neurons nimmt zwar gegenüber der digitalen Form ab, jedoch sind akzeptable Auflösungen in den Technologien erreichbar.

Auf Grundlage der hier durchgeführten Untersuchungen lässt sich zusammenfassend festhalten, dass neuronale Verfahren für neuartige Architekturen als Lösungsansatz der Probleme neuer Technologien bedingt geeignet sind. Als technisches System und den daraus folgenden Randbedingungen ergeben sich Robustheit und Fehlertoleranz, wobei die hier getroffenen Abschätzungen, z.B. für die gleichgradige Stetigkeit, in der Praxis teilweise schlecht handhabbar sein können. Auch stellt die Topologie des Netzes an sich noch nicht die gewünschten Eigenschaften sicher. Zudem ist es technisch schwierig, eine ähnlich hohe Anzahl von Verbindungen bei künstlichen neuronalen Netzen wie in der Biologie zu erreichen. Dennoch steht bei der Integration eines Lernprozesses sowie für neuartige Implementierungen boolescher Funktionen mit den neuronalen Verfahren eine viel versprechende Architektur zur Verfügung. Der Lernprozess ermöglicht adaptive Systeme, die auf geänderte Randbedingungen reagieren können. Dieser Vorgang darf jedoch nicht als Minimierung einer mathematischen Funktion verstanden werden, da fehlerhafte Berechnungen, eine der Randbedingungen dieser Untersuchungen, zu ungünstigen Lösungen führen. Hier können biologische Verständnisse des Lernvorganges zum Finden einer Lösung beitragen. Bei dem Entwurf boolescher Funktionen und den neuronalen Verfahren gibt es Zusammenhänge zwischen der Berechnung der Funktion und der Toleranz gegenüber Fehlern aufgrund identischer Beschreibungsformen. Hier kann dieses Wissen die Eigenschaften beim Logikentwurf positiv beeinflussen..

Weiterhin Forschungsbedarf besteht in der automatisierten Nutzung neuronaler Prinzipien durch die Entwurfswerkzeuge. Solange keine intrinsischen Lernverfahren angewendet werden, kann die Fehlertoleranz boolescher Funktionen durch Anwendung von Schwellenwertlogik erhöht werden. Die in dieser Arbeit vorgestellten Methodiken zur Bestimmung wichtiger Neuronen und Verbesserung der Eigenschaften können ebenfalls dazu beitragen, Systeme zuverlässiger zu entwerfen, wenn diese Verfahren automatisiert von Entwurfswerkzeugen durchgeführt werden. Eine essentielle Verbesserung der Fehlertoleranz kann durch die selbstständige Erkennung von Fehlern durch das System erreicht werden, wenn z.B. die Rekonfiguration eines Systems für die Kompensation von Fehlern genutzt wird. Dies ließe sich durch asynchrone Logiken, wie die *Null-Convention-Logik* [FB96], erreichen, die ebenfalls aufgrund des Einsatzes von Schwellenwertgatter Parallelen zu neuronalen Verfahren aufweist.

Projekte auf dem Bereich neuartiger Technologien (vgl. Abschnitt 2.3.2) sind im Bereich der Robustheit und Fehlertoleranz notwendig, da sich so z.B. die Herstellungskosten aufgrund geringerer Anforderungen adaptiver Systeme senken lassen. Wichtige Punkte sind dabei die Entwicklung neuer Fehlermodelle für die entsprechenden Technologien (Einzelelektronentransistor, Resonant-Tunnelemente, etc.) und die Ermittlung der zu erwartenden Ausbeute mit Hilfe statistischer Methoden. So können die Auswirkungen auf die Ausbeute und die Fehlertoleranz ermittelt und Verfahren zur Erhöhung der Ausbeute eingesetzt werden (z.B. durch Veränderungen

des Floorplans des Systems), wobei die Effekte einer weiteren Miniaturisierung näher zu untersuchen sind. Neben den Auswirkungen auf digitale Schaltungen müssen die Möglichkeiten der Analogtechnik auf Grundlage dieser neuartigen Bauelemente näher analysiert werden. Neben geeigneten Modellen für die Funktionalität oder das Rauschen sind die Möglichkeiten und Auswirkungen einer Mehrpegellogik zu untersuchen, welche z.B. mit den Resonant-Tunnelementen möglich ist. Ebenso sind hybride Technologien mit konventioneller CMOS-Technologie zu betrachten, um globale Verbindungen zu ermöglichen und daneben neue Übertragungsverfahren einzusetzen, welche das Problem hoher Verbindungsgrade betrachten. Hier können dann asynchrone Schaltungen, wie die Null-Convention-Logik, zum Einsatz kommen. Wünschenswert sind hier auch ein automatisierter und unterstützender Entwurf und die Optimierung der gesamten Strukturen durch die Entwurfswerkzeuge.

# Glossar

- $\gamma$  Konstante beim thermischen Rauschen von MOS-Transistoren. Für Langkanaltransistoren kann  $\gamma = 2/3$  gesetzt werden.
- $A$  Verstärkung.
- $B$  obere Schranke der Ausgangsgewichte:  $|\alpha_k| \leq B$ .
- $C$  obere Schranke des Abstandes zwischen Eingangsvektor und Zentrum:  
 $\|\mathbf{x} - \mathbf{c}_k\| \leq C$ .
- $\mathcal{C}$  Menge der stetigen Funktionen.
- $C_{DB}$  Drain-Bulk-Kapazität.
- $C_{GD}$  Gate-Drain-Kapazität.
- $C_{GS}$  Gate-Source-Kapazität.
- $C_L$  Lastkapazität am Ausgang.
- $C_{ox}$  flächenbezogene Oxidkapazität beim MOS-Transistor.
- $d$  Menge von Testvektoren, welche die zu approximierende Funktion  $v$  des neuronalen Netzes bestimmen.  $d = \{(\mathbf{x}_k, y_k) \in \mathbb{R}^n \times \mathbb{R}\}_{k=1}^N$ .
- $d(\mathbf{x}, \mathbf{y})$  Metrik zwischen  $\mathbf{x}$  und  $\mathbf{y}$ .
- $E$  Erwartungswert.
- $f$  Funktion des neuronalen Netzes:  $y = f(\mathbf{x})$ . Bei  $f_m$  bezeichnet der Index  $m$  die Anzahl von Neuronen im Netz.
- $g$  Gaußfunktion  $g(x) = e^{-x^2}$ .
- $g_m$  Transkonduktanz.
- $\mathbf{I}$  Einheitsmatrix.
- $\overline{i_n^2}$  (einseitige) Rauschstromdichte.
- $k$  Auflösung eines Signals in bit.

- $k_1$  Steilheit der Logistikkfunktion in der lokalen Funktion beim LCNN.
- $k_2$  Steilheit der Logistikkfunktion im Ausgang beim LCNN.
- $k_B$  Boltzmannkonstante.
- $l$  Wortbreite des Bruchanteils bei Festpunktdarstellung.
- $M$  Anzahl der Testvektoren.
- $m$  Anzahl von Neuronen. Bei einem RBF-Netzwerk bezeichnet  $m$  die Anzahl der überlagerten Basisfunktionen.
- $N$  Anzahl von eingefügten Neuronen zur Erhöhung der Redundanz. Dabei wird ein einzelnes Neuron durch  $N$  Neuronen ersetzt.
- $n$  Dimension des Eingangsvektors;  $\dim(\mathbf{x}) = n$ .
- $\mathbb{N}$  Menge der natürlichen Zahlen.
- $P$  elektrische Leistung.
- $p_{\underline{X}}(\mathbf{x})$  Verteilungsdichtefunktion eines Zufallsvektors.
- $q$  Grad des Polynoms zur Approximation.
- $\mathbb{R}$  Menge der reellen Zahlen.
- $T$  untere Schranke der Varianzen:  $|1/\sigma_j^k| \leq T$ . In Kapitel 5 bezeichnet  $T$  die Temperatur.
- $U$  Lebensdauer als Zufallsvariable.
- $U_{dd}$  Versorgungsspannung.
- $v$  Funktion des normierten Raumes  $X$  mit der Norm  $\|\cdot\|$ , welche die zu approximierende Funktion darstellt.
- $\overline{v_n^2}$  (einseitige) Rauschspannungsdichte.
- $\mathbf{W}$  Matrix; Gewichtsmatrix.
- $X$  Zufallsvariable.
- $x$  skalare Größe; Eingang.
- $\mathbf{x}$  Eingangsvektor eines neuronalen Netzes; vektorielle Größe.
- $x^i$  Element  $i$  aus einer Menge.

$y$  Ausgang des neuronalen Netzes, wobei  $y_m$  die Anzahl  $m$  der Neuronen im Netz bezeichnet.

**Ausgangsgewicht** Das Ausgangsgewicht bezeichnet die multiplikative Gewichtung der Signale einer Synapse von einem Neuron der letzten versteckten Ebene zum Ausgang.

**Bedeutung** Als Bedeutung wird der statische Beitrag eines Neurons zu der Ausgabe eines Netzes definiert:

**Definition 10** (Bedeutung eines Neurons). *Es seien  $y$  die ursprüngliche Netzfunktion und  $\hat{y}$  die Funktion des Netzes, aus dem das Neuron  $k$  entfernt wurde. Weiter bezeichne  $d^q(y, \hat{y}) = |y - \hat{y}|^q$  eine Metrik. Dann heißt*

$$s_k = E \{d^q(y, \hat{y})\} = \int_{\underline{\mathbf{x}}} d^q(y, \hat{y}) p_{\underline{\mathbf{x}}}(\mathbf{x}) d\mathbf{x} \quad (1)$$

Bedeutung des Neurons  $k$ .

**Ebene** Ein neuronales Netz ist aus mehreren Ebenen aufgebaut, wobei die Elemente eines vorwärtsgerichteten Netzes in derselben Ebene keine Verbindungen untereinander besitzen. Auch die Eingänge und die Ausgänge eines neuronalen Netzes stellen eine Ebene dar.

**Empfindlichkeit** Als Empfindlichkeit eines Neurons wird der maximale Gradient bzgl. der Eingänge verwendet. Die Empfindlichkeit beschreibt im Gegensatz zur Bedeutung eines Neurons eine Änderung einer Ausgangsgröße gegenüber einer Eingangsgröße

**Definition 11** (Empfindlichkeit eines Neurons). *Es sei  $g_k(\mathbf{x})$  die Aktivierungsfunktion eines Neurons  $k$  aus einem RBF-Netz, und  $\alpha_k$ ,  $\sigma_k$  und  $\mathbf{c}_k$  seien die freien Parameter dieser Gaußfunktion. Dann bezeichnet*

$$\max \{ \|\nabla g_k(\mathbf{x})\|^2 \} = \frac{\alpha_k^2}{\sigma_k^2} e^{-1} \quad (2)$$

die Empfindlichkeit des Neurons  $k$ .

**Fehler** In der mathematischen Definition bezeichnet der Fehler die Differenz zwischen dem korrekten/theoretischen und dem fehlerhaften/praktischen Wert. Ein Fehler bezeichnet in dieser Arbeit auch einen physikalischen Zusammenbruch oder ein Abweichen von der Spezifikation in der Hard- oder Software.

**fehlerbehaftet** Ein Neuron, dessen Ausgabe einen permanenten Fehler erzeugt, wird als fehlerbehaftet bezeichnet.

**fehlertolerant** Ein fehlertolerantes Netz ist gegenüber dem Ausfall bzw. dem Entfernen von Neuronen immun und liefert einen begrenzten Fehler im Ausgang. Bei dem Auftreten eines Fehlers kann das System innerhalb seiner Spezifikationen weiterarbeiten.

**Gewicht** Das Gewicht bezeichnet die multiplikative Gewichtung einer Synapse zwischen zwei Neuronen zweier unterschiedlicher Ebenen; siehe auch Ausgangsgewicht.

**gleichgradige Stetigkeit** Eine Menge von Funktionen besitzt die Eigenschaft der gleichgradigen Stetigkeit, falls  $\epsilon$  und  $\delta$  der stetigen Funktionen unabhängig von allen Funktionen in der Menge gewählt werden können.

**Definition 12** (gleichgradig stetig). *Es sei  $E \subseteq \mathbb{R}^n, E \neq \emptyset$ .  $\mathcal{S}$  sei eine Menge (Familie) von reellwertigen Funktionen  $f : E \rightarrow \mathbb{R}^m$ .  $\mathcal{S}$  heißt gleichgradig stetig, wenn es zu jedem  $\epsilon > 0$  ein  $\delta > 0$  gibt, so dass für alle  $f \in \mathcal{S}$  und alle  $x, y \in E$  mit  $|x - y| < \delta$  stets  $|f(x) - f(y)| < \epsilon$  bleibt.*

**gleichmäßige Stetigkeit** Eine Funktion ist gleichmäßig stetig, wenn in der Definition der Stetigkeit  $\delta$  nur von  $\epsilon$  abhängt und nicht von der Stelle  $x_0$ . Die Definition nach [Heu03] ergibt sich zu:

**Definition 13** (gleichmäßig stetig). *Die Funktion  $f$  heißt gleichmäßig stetig auf  $E$ , wenn es zu jedem  $\epsilon > 0$  ein  $\delta > 0$  gibt, so dass für alle  $x, y \in E$  mit  $|x - y| < \delta$  immer  $|f(x) - f(y)| < \epsilon$  ist.*

Jede gleichmäßig stetige Funktion ist somit erst recht stetig.

**inkonsistentes LGS** Für das LGS ergibt sich keine Lösung.

**kompakter Raum** Ein kompakter Raum bezeichnet eine topologische Eigenschaft, welche nützliche Eigenschaften für stetige Funktionen zur Folge hat. Für die Definition folgt [Heu03]:

**Definition 14** (kompakter Raum). *Ein topologischer Raum  $X$  heißt kompakt, wenn jede offene Überdeckung*

$$X = \bigcup_{i \in I} U_i, \quad U_i \text{ offen in } X \quad (3)$$

*eine endliche Teilüberdeckung*

$$X = U_{i_1} \cup U_{i_2} \cup \dots \cup U_{i_n} \quad (4)$$

*besitzt.*

**konvex** Bei einer konvexen Funktion liegen die Funktionswerte zwischen zwei Werten  $x$  und  $y$  unterhalb der Verbindungsgeraden durch die Funktionswerte  $f(x)$  und  $f(y)$ . Somit ist ein lokales Minimum einer konvexen Funktion auch gleichzeitig ein globales Minimum, wobei bei einer strikt konvexen Funktion nur ein globales Maximum existiert. Die mathematische Definition lautet [Heu03]:

**Definition 15** (konvexe Funktion). *Es sei  $\mathcal{I}$  ein nichtleeres Intervall auf  $\mathbb{R}$ . Eine Funktion  $f : \mathcal{I} \rightarrow \mathbb{R}$  heißt konvex auf  $\mathcal{I}$ , wenn gilt:*

$$f(ax + (1 - a)y) \leq af(x) + (1 - a)f(y) \quad (5)$$

für alle Punktepaare  $(x, y)$  in  $\mathcal{I}$  und  $a \in (0; 1)$ .

**LCNN** Local Cluster Neural Network.

**LGS** lineares Gleichungssystem.

**Logistikfunktion** Die Logistikfunktion ist eine monoton steigende Funktion, die nach unten und oben beschränkt ist. Für diese gilt:

$$\sigma(\iota, x) = \frac{1}{1 + e^{-\iota x}}.$$

**MLP** Multilayer Perceptron. Als Aktivierungsfunktion der Neuronen wird die Logistikfunktion verwendet.

**Monom** Ein Monom bezeichnet ein Polynom, das nur aus Produkten und Potenzen der Variablen und Koeffizienten besteht.

**mse** mittlerer quadratischer Fehler:  $\text{mse} = E[(y - \hat{y})^2]$ .

**multivariat** (lat. -engl.) mehrere Variablen betreffend.

**Neuron** Ein Neuron ist eine Funktion, die den Eingangsvektor  $\mathbf{x}$  auf den Ausgang  $y$  abbildet:  $y = h(\mathbf{x})$ . Die Funktion  $h$  ist die Aktivierungsfunktion des Neurons.

**nmse** normierter mittlerer quadratischer Fehler:  $\text{nmse} = \frac{E[(y - \hat{y})^2]}{\sigma_y^2}$  mit  $\sigma_y^2$  der Varianz des Ausgangssignals.

**NSR** Das Rausch-Signal-Verhältnis bezeichnet das Verhältnis von Signalleistung zur Leistung des Rauschens

$$\text{NSR} = \frac{N}{S}.$$

**RBF** radiales Basisfunktionsnetz. Ohne nähere Bezeichnung handelt es sich bei den Basisfunktionen um Gaußfunktionen.

**robust** Der Begriff bezieht sich auf die Eigenschaften eines neuronalen Netzes bei Rauschen in den Eingängen und den Parametern des Netzes. Ein robustes Netz liefert ein begrenztes Ausgangssignal bei Rauschen in den Eingängen oder Gewichten und kann somit Rauschen tolerieren.

**SA0** *Stuck-At-0* Fehler. Festsitzen auf dem logischen Wert '0'.

**SA1** *Stuck-At-1* Fehler. Festsitzen auf dem logischen Wert '1'.

**SNR** Das Signal-Rausch-Verhältnis bezeichnet das Verhältnis von Signalleistung zur Leistung des Rauschens. Für die Messung des Verhältnisses in dB folgt

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \frac{S}{N}.$$

**Synapse** Verbindungen zwischen Neuronen, die mit einem Faktor gewichtet werden.

**Zuverlässigkeit** Wahrscheinlichkeit, dass ein System nach seinen Spezifikationen korrekt funktioniert.

# Abbildungsverzeichnis

2.1	Verdeutlichung der Best-Approximation . . . . .	7
2.2	Approximation der si-Funktion durch Polynome . . . . .	9
2.3	Vergleich lokale und globale Approximation . . . . .	11
2.4	Funktionsweise eines MLPs . . . . .	15
2.5	Struktur des LCNNs durch den Aufbau mittels SLPs . . . . .	17
2.6	Datenfluss einer lokalen Funktion . . . . .	19
2.7	Approximation der si-Funktion durch ein MLP und ein LCNN . . . . .	20
2.8	Biologische Motivation eines RBFs . . . . .	21
2.9	Vergleich von RBF und LCNN . . . . .	24
2.10	Approximation der si-Funktion durch ein RBF-Netz . . . . .	25
2.11	Querschnitt eines MOSFETs . . . . .	28
2.12	Funktionsweise der Resonant-Tunnelemente . . . . .	31
3.1	Auswirkungen des Rauschens auf die Approximation eines RBF-Netzes . . . . .	38
3.2	Absoluter Fehler aufgrund Störungen der Parameter . . . . .	57
3.3	Auswirkungen der Quantisierung . . . . .	58
3.4	Datenflussgraph eines RBF-Netzes, Variante I . . . . .	59
3.5	Datenflussgraph eines RBF-Netzes, Variante II . . . . .	61
3.6	Datenflussgraph einer lokalen Funktion eines LCNNs . . . . .	62
3.7	Auswirkungen des Ausfalls von Neuronen im RBF-Netz . . . . .	63
3.8	Ausbeute bei mehreren Kopien . . . . .	77
3.9	Markov-Prozess des Ausfalls von Neuronen . . . . .	79
3.10	Vergleich der stationären Kosten . . . . .	84
4.1	Approximation mittels zweier Basisfunktionen . . . . .	88
4.2	Vereinfachte Bestimmung der Bedeutung einer Basisfunktion . . . . .	93

4.3	Ursachen für die ungenaue Bestimmung der Bedeutung . . . . .	98
4.4	Relative Größe des Netzes für die Testfunktionen . . . . .	107
4.5	Parameterwahl aufgrund der Ähnlichkeit . . . . .	111
4.6	Netzgröße und Approximationsgenauigkeit in Abhängigkeit des Schwellenwertes $\rho_{th}$ . . . . .	113
4.7	Definition der Sensibilität von Neuronen eines RBF-Netzes . . . . .	122
4.8	Verfahren zur Lösung des Mehrzieloptimierungsproblems . . . . .	127
4.9	Verlauf des mittleren quadratischen Fehlers in Abhängigkeit des Rauschens . . . . .	131
4.10	Verteilung der Zentren bei Robustheit . . . . .	133
4.11	Höhe und Breite der Gaußfunktionen bei Robustheit . . . . .	134
4.12	Approximation von Funktionen mit Berücksichtigung der Fehlertoleranz als Randbedingung . . . . .	137
4.13	Bedeutung der Neuronen nach der Mehrzieloptimierung . . . . .	138
4.14	Verteilung der Zentren für Fehlertoleranz . . . . .	139
4.15	Höhe und Breite der Gaußfunktionen bei Fehlertoleranz . . . . .	140
5.1	Addition mittels Repräsentation durch Ströme . . . . .	155
5.2	Fehler bei Approximation der Gaußfunktion . . . . .	158
5.3	Aufbau der rekonfigurierbaren Struktur SIRENS . . . . .	163
5.4	Erweiterung der SIRENS-Struktur . . . . .	165
5.5	Abbildung eines dynamischen Neurons durch SIRENS . . . . .	166
5.6	Realisierung von Allpässen . . . . .	169
5.7	Analoge Realisierung der sigmoiden Funktion . . . . .	171
5.8	Approximationseigenschaften der SIRENS-Struktur . . . . .	176
5.9	Fläche und Leistung des Basiselementes und des Erweiterungsmoduls	178
5.10	Entwurfspunkte der Basismodule für verschiedene Aktivierungsfunktionen . . . . .	180
5.11	Berechnung des eingangsbezogenen Rauschens . . . . .	182
5.12	Übertragungsfunktion des Differenzverstärkers in 32 nm CMOS . . .	190
5.13	Vergleich des analogen Entwurfes in verschiedenen Technologien . .	191
5.14	Häufigkeit der Leistung . . . . .	193
5.15	Häufigkeit der Verstärkung . . . . .	194

# Tabellenverzeichnis

2.1	Skalierungsregeln der CMOS-Technologie . . . . .	28
2.2	Technologieparameter für zukünftige CMOS-Technologien . . . . .	30
3.1	Anzahl der benötigten Rechenelemente zur Umsetzung des neuronalen Netzes . . . . .	63
4.1	Einstellende normierte, mittlere quadratische Fehler für verschiedene RBF-Netzwerke und Testfunktionen . . . . .	95
4.2	Fehler zwischen der Bedeutung eines Neurons und dem mittleren quadratischen Fehler bei einem Ausfall . . . . .	97
4.3	Normierter mittlerer quadratischer Fehler für verschiedene Testfunktionen beim LCNN . . . . .	103
4.4	Ergebnisse des Hypothesentests für zwei RBF-Netze bei Reduktion der Komplexität . . . . .	106
4.5	Reduzierung der Modellkomplexität mit Hilfe der Bedeutung bei einem LCNN . . . . .	108
4.6	Fehler der Approximation bei der Reduzierung der Modellkomplexität mittels der Ähnlichkeit zweier Gaußfunktionen . . . . .	114
4.7	Lösung des Mehrzieloptimierungsproblems für die Empfindlichkeit .	130
4.8	Lösung des Mehrzieloptimierungsproblems für den mse . . . . .	132
4.9	Lösung des Mehrzieloptimierungsproblems für die Fehlertoleranz . .	136
4.10	Eigenschaften unterschiedlicher Trainingsverfahren . . . . .	143
5.1	Anzahl von Grundrechenarten beim RBF . . . . .	155
5.2	Genauigkeit der Polynomapproximation . . . . .	161
5.3	Anzahl von Operationen der Polynomdarstellung . . . . .	162
5.4	Approximationseigenschaften der SIRENS-Struktur . . . . .	174
5.5	Approximationseigenschaften der inhomogenen SIRENS-Struktur .	175

---

5.6	Ergebnisse der Abbildung eines Basiselementes und des Erweiterungsmoduls auf die Technologie von ST . . . . .	177
5.7	Maximale Auflösung eines analogen Neurons . . . . .	187
5.8	Maximale Auflösung bei Berücksichtigung von 1/f-Rauschen . . . . .	188
5.9	Simulationsergebnisse des analogen Neurons . . . . .	189
5.10	Ergebnisse der Parametervariation . . . . .	192
5.11	Vergleich des analogen und digitalen Entwurfs eines Neurons . . . . .	195
5.12	Prognosen für ein Neuron auf Basis von Speicher und boolescher Funktionen . . . . .	196

# Literaturverzeichnis

- [Arb02] ARBIB, Michael A.: *The Handbook of Brain Theory and Neural Networks: Second Edition*. Cambridge, MA, USA : The MIT Press, 2002
- [Aro50] ARONSZAJN, N.: Theory of reproducing kernels. In: *Transactions of the American Mathematical Society* 68 (1950), S. 337–404. – ISSN 0002–9947
- [asa96] *1996 International Conference on Application-Specific Systems, Architectures, and Processors (ASAP '96), August 19-23, 1996, Chicago, IL , USA*. IEEE Computer Society, 1996
- [BAG00] BAUER, K.W. ; ALSING, S.G. ; GREENE, K.A.: Feature screening using signal-to-noise ratios. In: *Neurocomputing* 31 (March 2000), S. 29–44(16)
- [Ber12] BERNSTEIN, S.: Démonstration du théorème de Weierstrass, fondée sur le calcul des probabilités. In: *Harkov Soobs. Matem ob-va* 13 (1912), S. 1–2
- [Bis61] BISHOP, Errett: A generalization of the Stone-Weierstrass theorem. In: *Pacific J. Math.* 11 (1961), S. 777–783. – ISSN 0030–8730
- [Bis93] BISHOP, C.M.: Curvature-driven smoothing: a learning algorithm for feedforward networks. In: *IEEE Transactions on Neural Networks* 4 (1993), Nr. 5, S. 882–884. – ISSN 1045–9227
- [Bis95b] BISHOP, Christopher M.: *Neural Networks for Pattern Recognition*. New York, NY, USA : Oxford University Press, Inc., 1995. – ISBN 0198538642
- [BK80] BRENT, R. P. ; KUNG, H. T.: The chip complexity of binary arithmetic. In: *STOC '80: Proceedings of the twelfth annual ACM symposium on Theory of computing*. New York, NY, USA : ACM Press, 1980. – ISBN 0–89791–017–6, S. 190–200
- [BPVL94] BEIU, Valeriu ; PEPPERSTRAETE, Jan A. ; VANDEWALLE, Joos ; LAUWEREINS, Rudy: VLSI Complexity Reduction by Piece-Wise

- Approximation of the Sigmoid Function. In: VERLEYSEN, Michel (Hrsg.): *Proceedings of the European Symposium on Artificial Neural Networks*. Bruges, Belgium, April 1994, S. 181–186
- [BRRN05] BEIU, Valeriu ; RÜCKERT, Ulrich ; ROY, Sandip ; NYATHI, Jabulani: On Nanoelectronic Architectural Challenges and Solutions. In: *Proceedings of IEEE Conference on Nanotechnology IEEE NANO 2004*. München, Germany, 17 - 19 August 2005
- [BSM05] BRONSTEIN, Ilja N. ; SEMENDJAJEW, Konstantin A. ; MUSIOL, Gerhard: *Taschenbuch der Mathematik*. Deutsch (Harri), 2005. – ISBN 3817120060
- [Cas93] CASSANDRAS, Christos G.: *Discrete Event Systems: Modeling and Performance Analysis*. USA : Aksen Associates, Inc, 1993
- [CCG91] CHEN, S. ; COWAN, C.F.N. ; GRANT, P.M.: Orthogonal least squares learning algorithm for radial basis function networks. In: *IEEE Transactions on Neural Networks* 2 (1991), Nr. 2, S. 302–309. – ISSN 1045–9227
- [CFPF94] CHIPPERFIELD, A. J. ; FLEMING, P. J. ; POHLHEIM, H. ; FONSECA, C. M.: A genetic algorithm toolbox for Matlab. In: *Systems engineering: 10th International conference* Bd. 1. Coventry, UK : Coventry University, 1994. – ISBN 0–905949–23–4, S. 200–207
- [Com00] COMPAÑÓ, R.: Technology Roadmap for Nanoelectronics / European Commission. 2000 (Second Edition). – Forschungsbericht
- [Con03] CONSTANTINESCU, Cristian: Trends and Challenges in VLSI Circuit Reliability. In: *IEEE Micro* 23 (2003), Nr. 4, S. 14–19. – ISSN 0272–1732
- [CP03] CHAKRABORTY, D. ; PAL, N.R.: A novel training scheme for multilayered perceptrons to realize proper generalization and incremental learning. In: *IEEE Transactions on Neural Networks* 14 (2003), Nr. 1, S. 1–14. – ISSN 1045–9227
- [CPH05] CABESTANY, Joan (Hrsg.) ; PRIETO, Alberto (Hrsg.) ; HERNÁNDEZ, Francisco S. (Hrsg.): *Computational Intelligence and Bioinspired Systems, 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Vilanova i la Geltrú, Barcelona, Spain, June 8-10, 2005, Proceedings*. Bd. 3512. Springer, 2005 (Lecture Notes in Computer Science). – ISBN 3–540–26208–3

- [CS04b] COLLETTE, Yann ; SIARRY, Patrick: *Multiobjective Optimization : Principles and Case Studies (Decision Engineering)*. Springer, 2004. – ISBN 3540401822
- [Cun90] CUNNINGHAM, J.A.: The use and evaluation of yield models in integrated circuit manufacturing. In: *Semiconductor Manufacturing, IEEE Transactions on* 3 (1990), Nr. 2, S. 60–71. – ISSN 0894–6507
- [DGR<sup>+</sup>74] DENNARD, R.H. ; GAENSSLEN, F.H. ; RIDEOUT, V.L. ; BASSOUS, E. ; LEBLANC, A.R.: Design of ion-implanted MOSFET's with very small physical dimensions. In: *IEEE Journal of Solid-State Circuits* 9 (1974), Oct, Nr. 5, S. 256–268
- [DKOZ05a] DUCH, Włodzisław (Hrsg.) ; KACPRZYK, Janusz (Hrsg.) ; OJA, Erkki (Hrsg.) ; ZADROZNY, Sławomir (Hrsg.): *Artificial Neural Networks: Biological Inspirations - ICANN 2005, 15th International Conference, Warsaw, Poland, September 11-15, 2005, Proceedings, Part I*. Bd. 3696. Springer, 2005 (Lecture Notes in Computer Science). – ISBN 3–540–28752–3
- [DSH05] DELLNITZ, M. ; SCHÜTZE, O. ; HESTERMEYER, T.: Covering Pareto sets by multilevel subdivision techniques. In: *Journal of Optimization Theory and Applications* 124 (2005), Nr. 1, S. 113–136. – ISSN 0022–3239
- [Duc06] DUCH, Włodzisław: K-Separability. In: KOLLIAS, Stefanos (Hrsg.) ; STAFYLOPATIS, Andreas (Hrsg.) ; DUCH, Włodzisław (Hrsg.) ; OJA, Erkki (Hrsg.): *Artificial Neural Networks – ICANN 2006* Bd. 4131, Springer, 2006 (LNCS). – ISBN 3–540–38625–4, S. 188–197
- [Ehr00] EHRHARDT, Dietmar: *Integrierte analoge Schaltungstechnik*. Vieweg Verlagsgesellschaft, 2000. – ISBN 3528038608
- [FB96] FANT, Karl M. ; BRANDT, Scott A.: NULL Convention Logic/sup TM/: A Complete And Consistent Logic For Asynchronous Digital Circuit Synthesis. In: ASAP[asa96], S. 261–273
- [FDN<sup>+</sup>01] FRANK, D.J. ; DENNARD, R.H. ; NOWAK, E. ; SOLOMON, P.M. ; TAUR, Y. ; WONG, Hon-Sum P.: Device scaling limits of Si MOSFETs and their application dependencies. In: *Proceedings of the IEEE* 89 (2001), March, Nr. 3, S. 259–288
- [Fra03] FRAENKEN, Dietrich: *Entwurf und Synthese von Digitalfiltern / Universität Paderborn*. Paderborn, 2003. – Skript zur Vorlesung

- [Fun89] FUNAHASHI, K.: On the approximate realization of continuous mappings by neural networks. In: *Neural Netw.* 2 (1989), Nr. 3, S. 183–192. – ISSN 0893–6080
- [FWV06] FRANÇOIS, D. ; WERTZ, V. ; VERLEYSSEN, M.: The permutation test for feature selection by mutual information. In: VERLEYSSEN, Michel (Hrsg.): *Proceedings of the 14th European Symposium on Artificial Neural Networks (ESANN)*. Evere, Belgium : D-Side Publications, 26-28 April 2006. – ISBN 2–930307–06–4, S. 239–244
- [Gen92] GENERAL ACCOUNTING OFFICE: Patriot Missile Defense - Software problem led to system failure at Dhahran / GAO/IMTEC-92-26. Washington, United States, February 1992 (B-247094). – Bericht
- [GGML+97] GOLDHABER-GORDON, D. ; MONTEMERLO, M.S. ; LOVE, J.C. ; OPITECK, G.J. ; ELLENBOGEN, J.C.: Overview of nanoelectronic devices. In: *Proceedings of the IEEE* 85 (1997), Nr. 4, S. 521–540. – ISSN 0018–9219
- [GJP95] GIROSI, Federico ; JONES, Michael ; POGGIO, Tomaso: Regularization Theory and Neural Networks Architectures. In: *Neural Computation* 7 (1995), Nr. 2, S. 219–269
- [Gla99] GLASKOWSKY, P.N.: MoSys Explains 1T-SRAM Technology. In: *Microprocessor Report* 13 (1999), September, Nr. 12
- [Gle96] GLEICK, James: Little Bug, Big Bang. In: *The New York Times Magazine* (1996), 1 December
- [GLL96] GOLUB, Gene H. ; LOAN, Charles F. ; LOAN, Charles F. V.: *Matrix Computations (Johns Hopkins Series in the Mathematical Sciences)*. Johns Hopkins University Press, 1996. – ISBN 0801854148
- [GMHL01] GRAY, Paul R. ; MEYER, Robert G. ; HURST, Paul J. ; LEWIS, Stephen H.: *Analysis and Design of Analog Integrated Circuits*. New York, NY, USA : John Wiley & Sons, Inc., 2001. – ISBN 0471321680
- [GP90] GIROSI, Federico ; POGGIO, Tomaso: Networks and the best approximation property. In: *Biological Cybernetics* 63 (1990), S. 169–176

- [GPC90] GIROSI, Federico ; POGGIO, Tomaso ; CAPRILE, Bruno: Extensions of a Theory of Networks for Approximation and Learning. In: *NIPS*, 1990, S. 750–756
- [GRO<sup>+</sup>03] GONZALEZ, J. ; ROJAS, I. ; ORTEGA, J. ; POMARES, H. ; FERNANDEZ, F.J. ; DIAZ, A.F.: Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. In: *IEEE Transactions on Neural Networks* 14 (2003), Nr. 6, S. 1478–1495. – ISSN 1045–9227
- [GS92] GEVA, Shlomo ; SITTE, Joaquin: A Constructive Method for Multivariate Function Approximation by Multilayer Perceptrons. In: *IEEE Transactions on Neural Networks* 3 (1992), Juli, Nr. 4, S. 621–624
- [GS93] GEVA, S. ; SITTE, J.: Local response neural networks and fuzzy logic for control. In: *Emerging Technologies and Factory Automation, 1993. Design and Operations of Intelligent Factories. Workshop Proceedings. IEEE 2nd International Workshop on*, 1993, S. 51–57
- [Ham90] HAMMERSTROM, D.: A VLSI architecture for high-performance, low-cost, on-chip learning. In: *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, 1990, S. 537–544 vol.2
- [Hay99] HAYKIN, Simon: *Neural Networks. A Comprehensive Foundation*. Second Edition. New Jersey, USA : Prentice Hall, 1999
- [HCS06] HUANG, Guang-Bin ; CHEN, Lei ; SIEW, Chee-Kheong: Universal approximation using incremental constructive feedforward networks with random hidden nodes. In: *IEEE Transactions on Neural Networks* 17 (2006), Nr. 4, S. 879–892. – ISSN 1045–9227
- [Heu03] HEUSER, Harro: *Lehrbuch der Analysis*. 15. Stuttgart, Leipzig, Wiesbaden : B. G. Teubner, 2003
- [HH93] HOLT, J. L. ; HWANG, J. N.: Finite Precision Error Analysis of Neural Network Hardware Implementations. In: *IEEE Transactions on Computers* 42 (1993), Nr. 3, S. 281–290. – ISSN 0018–9340
- [HHO98] HYVÄRINEN, Aapo ; HOYER, Patrik O. ; OJA, Erkki: Sparse Code Shrinkage: Denoising by Nonlinear Maximum Likelihood Estimation. In: [KSC99], S. 473–479
- [Hil00] HILBERT, David: Mathematische Probleme. In: *Göttinger Nachrichten* (1900), S. 253–197

- [Hil04] HILLERINGMANN, Ulrich: *Silizium-Halbleitertechnologie*. Teubner, 2004. – ISBN 3519301490
- [HJ03a] HAN, J. ; JONKER, P.: A defect- and fault-tolerant architecture for nanocomputers. In: *Nanotechnology* 14 (2003), Februar, S. 224–230
- [HK04] HANSEN, Nikolaus ; KERN, Stefan: *Evaluating the CMA Evolution Strategy on Multimodal Test Functions*. Bd. 3242. Springer, 2004. – 282–291 S.
- [HKO01] HYVÄRINEN, A. ; KARHUNEN, J. ; OJA, E.: *Independent Component Analysis*. New York : John Wiley, 2001. – 481+xxii pages
- [HN87] HECHT-NIELSEN, Robert: Kolmogorov’s mapping neural network existence theorem. In: *Proceedings of IEEE First Annual International Conference on Neural Networks* Bd. 3. San Diego, CA, 1987, S. 11–13
- [HND<sup>+</sup>06] HAENSCH, W. ; NOWAK, E. J. ; DENNARD, R. H. ; SOLOMON, P. M. ; BRYANT, A. ; DOKUMACI, O. H. ; KUMAR, A. ; WANG, X. ; JOHNSON, J. B. ; FISCHETTI, M. V.: Silicon CMOS devices beyond scaling. In: *IBM Journal of Research and Development* 50 (2006), July/September, Nr. 4/5, S. 339–361
- [HP03] HUFFMAN, W. C. ; PLESS, Vera: *Fundamentals of Error-Correcting Codes*. Cambridge University Press, 2003. – ISBN 0521782805
- [HSS05] HUANG, Guang-Bin ; SARATCHANDRAN, P. ; SUNDARARAJAN, N.: A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation. In: *IEEE Transactions on Neural Networks* 16 (2005), Nr. 1, S. 57–67. – ISSN 1045–9227
- [HSW89] HORNIK, K. ; STINCHCOMBE, M. ; WHITE, H.: Multilayer feedforward networks are universal approximators. In: *Neural Networks* 2 (1989), Nr. 5, S. 359–366. – ISSN 0893–6080
- [HUL93] HIRIART-URRUTY, Jean-Baptiste ; LEMARÉCHAL, Claude: *Convex Analysis and Minimization Algorithms I: Fundamentals (Grundlehren der mathematischen Wissenschaften)*. Springer, 1993
- [HZM<sup>+</sup>06] HUANG, G.-B. ; ZHU, Q.-Y. ; MAO, K.Z. ; SIEW, C.-K. ; SARATCHANDRAN, P. ; SUNDARARAJAN, N.: Can Threshold Networks

- be Trained Directly? In: *IEEE Transactions on Circuits and Systems II: Express Briefs*[see also *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*] 53 (2006), Nr. 3, S. 187–191. – ISSN 1057–7130
- [Isk94] ISKE, Armin: *Charakterisierung bedingt positiv definierter Funktionen für multivariante Interpolationsmethoden mit radialen Basisfunktionen*. Göttingen, Georg-August-Universität, Doktorarbeit, 1994
- [Kur92] KURKOVÁ, V.: Kolmogorov's theorem and multilayer neural networks. In: *Neural Networks* 5 (1992), S. 501–506
- [Kam04] KAMMEYER, Karl D.: *Nachrichtenübertragung*. 3., neubearb. und erg. Aufl. Teubner, 2004. – ISBN 3–519–26142–1
- [KK00] KOECHER, Max ; KRIEG, Aloys: *Ebene Geometrie*. 2., neu bearb. und erw. Aufl. Berlin/Heidelberg : Springer, 2000. – ISBN 3–540–67643–0
- [KKT01] KNOBLINGER, G. ; KLEIN, P. ; TIEBOUT, H.: A new model for thermal channel noise of deep-submicron MOSFETs and its application in RF-CMOS design. In: *IEEE Journal of Solid-State Circuits* 36 (2001), May, Nr. 5, S. 831–837
- [KKV00] KAINEN, Paul C. ; KURKOVÁ, Vera ; VOGT, Andrew: Best approximation by Heaviside perceptron networks. In: *Neural Networks* 13 (2000), Sep, Nr. 7, S. 695–697
- [Kle99] KLEIN, P.: An analytical thermal noise model of deep submicron MOSFET's. In: *IEEE Electron Device Letters* 20 (1999), Aug., Nr. 8, S. 399–401
- [Koi95] KOIVISTO, Hannu: *A Practical Approach to Model Based Neural Network Control*. Tampere, Finland, Tampere University of Technology, PhD-Thesis, 1995
- [Kol57] KOLMOGOROV, A. N.: On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition. In: *Doklady Akademii Nauk SSSR* 114 (1957), S. 953–956
- [KSC99] KEARNS, Michael J. (Hrsg.) ; SOLLA, Sara A. (Hrsg.) ; COHN, David A. (Hrsg.): *Advances in Neural Information Processing Systems 11, [NIPS Conference, Denver, Colorado, USA, November*

- 30 - December 5, 1998]. The MIT Press, 1999 . – ISBN 0–262–11245–0
- [KWSZVD00] KNITEL, M.J. ; WOERLEE, P.H. ; SCHOLTEN, A.J. ; ZEGERS-VAN DUIJNHOFEN, A.: Impact of process scaling on 1/f noise in advanced CMOS technologies. In: *Electron Devices Meeting, 2000. IEDM Technical Digest. International*, 2000, S. 463–466
- [LCJH00] LIU, W. ; CAO, K.M. ; JIN, X. ; HU, Chenming: BSIM 4.0.0 Technical Notes / EECS Department, University of California, Berkeley. 2000 (UCB/ERL M00/39). – Forschungsbericht
- [Lil30] LILIENFELD, J. E.: *Device for controlling electric current*. US Patent 1,900,018, March 1930
- [LKI<sup>+</sup>02] LIU, Shih-Chii ; KRAMER, Jörg ; INDIVERI, Giacomo ; DELBRÜCK, Tobias ; DOUGLAS, Rodney: *Analog VLSI: Circuits and Principles*. Cambridge, Massachusetts, London, England : A Bradford Book, The MIT Press, 2002
- [LXM03] LAI, C.D. ; XIE, Min ; MURTHY, D.N.P.: A modified Weibull distribution. In: *Reliability, IEEE Transactions on* 52 (2003), March, Nr. 1, S. 33–37
- [Man99] MANKU, T.: Microwave CMOS-device physics and design. In: *Solid-State Circuits, IEEE Journal of* 34 (1999), March, Nr. 3, S. 277–285
- [MB99a] MAASS, Wolfgang (Hrsg.) ; BISHOP, Christopher M. (Hrsg.): *Pulsed neural networks*. Cambridge, MA, USA : MIT Press, 1999. – ISBN 0–626–13350–4
- [MB99b] MENDIL, B. ; BENMAHAMMED, K.: Simple activation functions for neural and fuzzy neural networks. In: *Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on* Bd. 5, 1999, S. 347–350 vol.5
- [McC43] MCCULLOCH, W. Pitts W. S.: A Logical Calculus of Ideas Immanent in Nervous Activity. In: *Bulletin of Mathematical Biophysics* 5 (1943), S. 115–133
- [MD89] MOODY, J. ; DARKEN, C. J.: Fast learning in networks of locally-tuned processing units. In: *Neural Computation* 1 (1989), Nr. 2, S. 281–294

- [MH98] MILLER, Gad ; HORN, David: Probability Density Estimation Using Entropy Maximization. In: *Neural Computation* 10 (1998), Nr. 7, S. 1925–1938
- [MH05] MAO, K.Z. ; HUANG, G.-B.: Neuron Selection for RBF Neural Network Classifier Based on Data Structure Preserving Criterion. In: *Neural Networks, IEEE Transactions on* 16 (2005), Nr. 6, S. 1531–1540. – ISSN 1045–9227
- [MHL92] MOODY, John E. (Hrsg.) ; HANSON, Stephen J. (Hrsg.) ; LIPPMANN, Richard (Hrsg.): *Advances in Neural Information Processing Systems 4, [NIPS Conference, Denver, Colorado, USA, December 2-5, 1991]*. Morgan Kaufmann, 1992. – ISBN 1–55860–222–4
- [MMMR96] MENON, Anil ; MEHROTRA, Kishan ; MOHAN, Chilukuri K. ; RANKA, Sanjay: Characterization of a class of sigmoid functions with applications to neural networks. In: *Neural Netw.* 9 (1996), Nr. 5, S. 819–835. – ISSN 0893–6080
- [Moo65] MOORE, Gordon E.: Cramming more components onto integrated circuits. In: *Electronics* 38 (1965), 19 April, Nr. 8, S. 114–117
- [Moo75] MOORE, G.E.: Progress in digital integrated electronics. In: *Electron Devices Meeting, 1975 International Bd.* 21, 1975, S. 11–13
- [Moo91] MOODY, John E.: The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems. In: [MHL92], S. 847–854
- [Mos04] MOSTAGHIM, S.: *Multi-objective Evolutionary Algorithms. Data Structures, Convergence, and Diversity*. Paderborn, Germany, University of Paderborn, PhD-Thesis, 2004
- [MP99] MAIOROV, Vitaly ; PINKUS, Allan: Lower bounds for approximation by MLP neural networks. In: *Neurocomputing* 25 (1999), Nr. 1-3, S. 81–91
- [MS00] MAASS, Wolfgang ; SONTAG, Eduardo D.: Neural Systems as Nonlinear Filters. In: *Neural Computation* 12 (2000), Nr. 8, S. 1743–1772
- [MS03] MALMSTROM, Kurt ; SITTE, Joaquin: ELSY Release 1.0 / Queensland University of Technology. Brisbane, Australia, 2003. – User Manual

- [MU91] MOODY, John E. ; UTANS, Joachim: Principled Architecture Selection for Neural Networks: Application to Corporate Bond Rating Prediction. In: [MHL92], S. 683–690
- [Nab02] NABNEY, Ian: *NETLAB: algorithms for pattern recognitions*. New York, NY, USA : pub-SV, 2002 (Advances in pattern recognition). – xviii + 420 S. – ISBN 1–85233–440–1
- [NSF02] NIKOLIC, K ; SADEK, A ; FORSHAW, M: Fault-tolerant techniques for nanocomputers. In: *Nanotechnology* 13 (2002), Nr. 3, S. 357–362
- [Pap01] PAPULA, Lothar: *Mathematik für Ingenieure und Naturwissenschaftler Band 3. Vektoranalysis, Wahrscheinlichkeitsrechnung, Mathematische Statistik, Fehler- und Ausgleichsrechnung*. Vieweg Verlag, 2001. – ISBN 3528349379
- [PC01] PARRA, Xavier ; CATALÀ, Andreu: Learning Fault Tolerance in Radial Basis Function Networks. In: VERLEYSSEN, Michel (Hrsg.): *Proceedings of the 9. European Symposium on Artificial Neural Networks (ESANN)*. Evere, Belgium : D-Side Publications, April 2001, S. 341–346
- [Pea01b] PEARSON, K.: On lines and planes of closest fit to systems of points in space. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal* 6 (1901), Nr. 2, S. 559–572
- [PG89] POGGIO, Tomaso ; GIROSI, Federico: A Theory of Networks for Approximation and Learning / Massachusetts Institute of Technology (MIT). Cambridge, MA, USA, 1989. – Forschungsbericht
- [PG90a] POGGIO, T. ; GIROSI, F.: Networks for approximation and learning. In: *Proceedings of the IEEE* 78 (1990), Nr. 9, S. 1481–1497. – ISSN 0018–9219
- [PG90b] POGGIO, T. ; GIROSI, F.: Regularization algorithms for learning that are equivalent to multilayer networks. In: *Science* 247 (1990), S. 978–982
- [Pla04] PLATO, Robert: *Numerische Mathematik kompakt*. 2. überarb. Aufl. Vieweg, 2004. – ISBN 3–528–13153–5
- [Pog90] POGGIO, Tomaso: A Theory of How the Brain Might Work / Massachusetts Institute of Technology. Cambridge, MA, USA : Massachusetts Institute of Technology, 1990. – Forschungsbericht

- [Por02] PORRMANN, Mario: Entwurf mikroelektronischer Systeme / Universität Paderborn. Paderborn, Juni 2002. – Skript zur Vorlesung
- [Rab96] RABAEY, Jan M.: *Digital integrated circuits: a design perspective*. Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 1996. – ISBN 0–13–178609–1
- [Raz00] RAZAVI, Behzad: *Design of Analog CMOS Integrated Circuits*. McGraw-Hill, 2000
- [RB93] RIEDMILLER, Martin ; BRAUN, Heinrich: A Direct Adaptive Method for Faster Backpropagation Learning: the RPROP Algorithm. In: *IEEE International Conference on Neural Networks*, 1993, S. 586–591
- [RB05] ROY, S. ; BEIU, V.: Majority multiplexing-economical redundant fault-tolerant designs for nanoarchitectures. In: *IEEE Transactions on Nanotechnology* 4 (2005), July, Nr. 4, S. 441–451
- [RHM86] RUMELHART, D. E. ; HINTON, G. E. ; MCCLELLAND, J. L.: A general framework for parallel distributed processing. (1986), S. 45–76. ISBN 0–262–68053–X
- [RHW86a] RUMELHART, D. E. ; HINTON, G. E. ; WILLIAMS, R. J.: Learning Internal Representations by Error Propagation. In: RUMELHART, D. (Hrsg.) ; MCCLELLAND, J. (Hrsg.): *Parallel Distributed Processing* Bd. 1. Cambridge, MA : MIT Press, 1986, S. 318–362
- [RHW86b] RUMELHART, D. E. ; HINTON, G. E. ; WILLIAMS, R. J.: Learning internal representations by error propagation. (1986), S. 318–362. ISBN 0–262–68053–X
- [Ros58] ROSENBLATT, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. In: *Psychological Review* 65 (1958), Nr. 6, S. 386–408
- [Ros60] ROSENBROCK, H. H.: An automatic method for finding the greatest or least value of a function. In: *The Computer Journal* 3 (1960), Nr. 3, S. 175–184
- [RT97] ROLLS, E. T. ; TREVES, A.: *Neural Networks and Brain Function*. Oxford : Oxford University Press, 1997
- [Rud76] RUDIN, Walter: *Principles of Mathematical Analysis*. McGraw-Hill, 1976 (International Series in Pure and Applied Mathematics)

- [Sar98] SARPESHKAR, Rahul: Analog versus digital: extrapolating from electronics to neurobiology. In: *Neural Comput.* 10 (1998), Nr. 7, S. 1601–1638. – ISSN 0899–7667
- [Sau01] SAUER, Tomas: Approximationstheorie / Justus-Liebig-Universität Gießen. Gießen, Deutschland, 2001. – Skript zur Vorlesung
- [SC99] SIMOEN, E. ; CLAEYS, C.: On the flicker noise in submicron silicon MOSFETs. 43 (1999), Mai, S. 865–882(18)
- [Sch94] SCHÜSSLER, Hans W.: *Digitale Signalverarbeitung 1*. Bd. 1. Berlin/Heidelberg : Springer-Verlag, 1994
- [Sch97a] SCHMAJUK, Nestor A.: *Animal learning and cognition*. 1. publ. Cambridge Univ. Press, 1997. – ISBN 0–521–45086–1, 0–521–45696–7
- [Sch98a] SCHICHL, Hermann: Numerik 2 / Universität Wien. Wien, 1998. – Skript zur Vorlesung
- [Sch02] SCHMITT, Michael: On the Complexity of Computing and Learning with Multiplicative Neural Networks. In: *Neural Computation* 14 (2002), Nr. 2, S. 241–301
- [Sch04] SCHMITT, Bernhard: Numerik I / Philipps-Universität Marburg, Fachbereich Mathematik und Informatik. Marburg, 2004. – Skript zur Vorlesung
- [Seg98] SEGEY, Idan ; KOCH, Christof (Hrsg.) ; SEGEV, Idan (Hrsg.): *Methods in Neuronal Modeling: From Ions to Networks*. Cambridge, MA, USA : MIT Press, 1998. – ISBN 0262112310
- [Sem05] SEMICONDUCTOR INDUSTRY ASSOCIATION: *International Technology Roadmap for Semiconductors - 2005 Edition*. 2005. <http://public.itrs.net/>, 2005
- [SGJF06] SUBIRATS, José L. ; GÓMEZ, Iván ; JEREZ, José M. ; FRANCO, Leonardo: Optimal Synthesis of Boolean Functions by Threshold Functions. In: KOLLIAS, Stefanos (Hrsg.) ; STAFYLOPATIS, Andreas (Hrsg.) ; DUCH, Włodzisław (Hrsg.) ; OJA, Erkki (Hrsg.): *Artificial Neural Networks – ICANN 2006* Bd. 4131, Springer, 2006 (LNCS). – ISBN 3–540–38625–4, S. 983–992
- [Sha48] SHANNON, C. E.: A mathematical theory of communication. In: *Bell System Technical Journal* 27 (1948), Jul and Oct, S. 379–423 and 623–656

- [Sim64] SIMMONS, G. F.: *Topology and Modern Analysis*. Tokyo, Japan : McGraw-Hill, 1964
- [SKR98] SITTE, Joaquin ; KÖRNER, Tim ; RÜCKERT, Ulrich: Local Cluster Neural Net: Analog VLSI Design. In: *Neurocomputing* 19 (1998), S. 185 – 197
- [Smi97] SMITH, Michael John S.: *Application-Specific Integrated Circuits*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1997. – ISBN 0–201–50022–1
- [SNT85] SAWARAGI, Y. ; NAKAYAMA, H. ; TANINO, T.: *Theory of Multi-objective Optimization*. Academic Press, Orlando, 1985
- [SS02] SCHÖLKOPF, Bernhard ; SMOLA, Alex J.: *Learning with Kernels*. Cambridge, MA : MIT Press, 2002
- [SSM98f] SMOLA, Alex J. ; SCHÖLKOPF, Bernhard ; MÜLLER, Klaus-Robert: The connection between regularization operators and support vector kernels. In: *Neural Networks* 11 (1998), Nr. 4, S. 637–649. – ISSN 0893–6080
- [Ste04b] STEINMETZ, Norbert: Analysis I/II/III / Universität Dortmund. Dortmund, Juni 2004. – Skript zur Vorlesung
- [STL+03] SCHOLTEN, A.J. ; TIEMEIJER, L.F. ; LANGEVELDE, R. van ; HAVENS, R.J. ; DUIJNHOFEN, A.T.A. Zegers-van ; VENEZIA, V.C.: Noise modeling for RF CMOS circuit simulation. In: *Electron Devices, IEEE Transactions on* 50 (2003), March, Nr. 3, S. 618–632
- [STM05] STMICROELECTRONICS: 0.12 CMOS HCMOS9. 2005. – Design Kit
- [Tau02] TAUR, Yuan: CMOS design near the limit of scaling. In: *IBM Journal of Research and Development* 46 (2002), March/May, Nr. 2/3, S. 213–222
- [The92] THE MATHWORKS, INC.: *The Student Edition of MATLAB: Student User Guide*. Upper Saddle River, NJ 07458, USA : Prentice-Hall, 1992 (The MATLAB curriculum series). – xiv + 494 S. – ISBN 0–13–856006–4
- [Tka03] TKACHUK, V. V.: Properties of function spaces reflected by uniformly dense subspaces. In: *Topology and its Applications* 132 (2003), August, Nr. 2, S. 183–193

- [Tri05] TRIESCH, Jochen: A Gradient Rule for the Plasticity of a Neuron's Intrinsic Excitability. In: [DKOZ05a], S. 65–70
- [Vap95b] VAPNIK, V.: *The Nature of Statistical Learning Theory*. N.Y. : Springer, 1995. – ISBN 0–387–94559–8
- [Vap98] VAPNIK, Vladimir N.: *Statistical Learning Theory*. New York : Wiley-Interscience, 1998. – ISBN 0471030031
- [VB02] VERIKAS, A. ; BACAUSKIENE, M.: Feature selection with neural networks. In: *Pattern Recognition Letters* 23 (2002), September, Nr. 11, S. 1323–1335
- [VBTG05] VINCENT, B. T. ; BADDELEY, R. J. ; TROSCIANKO, T ; GILCHRIST, I. D.: Is the early visual system optimised to be energy efficient? In: *Network: Computation in Neural Systems* 16 (2005), S. 175–190
- [VF05] VERLEYSSEN, Michel ; FRANÇOIS, Damien: The Curse of Dimensionality in Data Mining and Time Series Prediction. In: [CPH05], S. 758–770
- [Weg91] WEGENER, Ingo: *The Complexity of Boolean Functions*. John Wiley & Sons, 1991. – ISBN 0471915556
- [WTK05] WALLACE, Manolis ; TSAPATSOULIS, Nicolas ; KOLLIAS, Stefanos: Intelligent initialization of resource allocating RBF networks. In: *Neural Networks* 18 (2005), März, Nr. 2, S. 117–122
- [ZC06] ZHAO, Wei ; CAO, Yu: New Generation of Predictive Technology Model for Sub-45nm Design Exploration. In: *ISQED '06: Proceedings of the 7th International Symposium on Quality Electronic Design*. Washington, DC, USA : IEEE Computer Society, 2006. – ISBN 0–7695–2523–7, S. 585–590

## Eigene Veröffentlichungen

- [EKR06] EICKHOFF, Ralf ; KAULMANN, Tim ; RÜCKERT, Ulrich: SIRENS: A Simple Reconfigurable Neural Hardware Structure for artificial neural network implementations. In: *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN)*. Vancouver, Canada, 16-21 jul 2006, S. on CD

- [EKR07b] EICKHOFF, Ralf ; KAULMANN, Tim ; RÜCKERT, Ulrich: Neural inspired architectures for nanoelectronics. In: *Proceedings of the 9th International Work-Conference on Artificial Neural Networks (IWANN)* Bd. 4507. San Sebastian, Spain : Springer Berlin/Heidelberg, 20-22 jun 2007 (Lecture Notes in Computer Science), 414–421
- [ENPR05] EICKHOFF, Ralf ; NIEMANN, Jörg-Christian ; PORRMANN, Mario ; RÜCKERT, Ulrich: Adaptable Switch boxes as on-chip routing nodes for networks-on-chip. In: RETTBERG, Achim (Hrsg.) ; ZANELLA, Mauro C. (Hrsg.) ; RAMMIG, Franz J. (Hrsg.): *From Specification to Embedded Systems Application, International Embedded Systems Symposium (IESS)*. Manaus, Brazil, 15-17 August 2005, S. 201–210
- [ER05a] EICKHOFF, Ralf ; RÜCKERT, Ulrich: Fault-Tolerance of Basis Function Networks using Tensor Product Stabilizers. In: *Proceedings of the International Conference on Systems, Man and Cybernetics, IEEE SMC 2005*. Hawaii, USA, 10 - 12 oct 2005
- [ER05b] EICKHOFF, Ralf ; RÜCKERT, Ulrich: Robustness of Radial Basis Functions. In: CABESTANY, Joan (Hrsg.) ; PRIETO, Alberto (Hrsg.) ; SANDOVAL, Francisco (Hrsg.): *Proceedings of the 8th International Work-Conference on Artificial Neural Networks (IWANN)* Bd. 3512. Barcelona, Spain : Springer Berlin/Heidelberg, 8-10 jun 2005 (Lecture Notes in Computer Science), 264–271
- [ER05c] EICKHOFF, Ralf ; RÜCKERT, Ulrich: Tolerance of Radial Basis Functions Against Stuck-At-Faults. In: DUCH, Włodzisław (Hrsg.) ; KACPRZYK, Janusz (Hrsg.) ; OJA, Erkki (Hrsg.) ; ZADROZNY, Sławomir (Hrsg.): *Proceedings of the 15th International Conference on Artificial Neural Networks (ICANN)* Bd. 3697. Warsaw, Poland : Springer Berlin/Heidelberg, 11-15 sep 2005 (Lecture Notes in Computer Science), 1003–1008
- [ER06a] EICKHOFF, Ralf ; RÜCKERT, Ulrich: Enhancing Fault Tolerance of Radial Basis Functions. In: *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN)*. Vancouver, Canada, 16-21 jul 2006, S. on CD
- [ER06b] EICKHOFF, Ralf ; RÜCKERT, Ulrich: Pareto-optimal noise and approximation properties of RBF networks. In: KOLLIAS, Stefanos (Hrsg.) ; STAFYLOPATIS, Andreas (Hrsg.) ; DUCH, Włodzisław (Hrsg.) ; OJA, Erkki (Hrsg.): *Proceedings of the 16th International Conference on Artificial Neural Networks (ICANN)*. Athens, Greece : Springer Berlin/Heidelberg, 10-14 sep 2006 (Lecture Notes in Computer Science). – ISBN 3–540–38625–4, S. 993–1002

- [ER06c] EICKHOFF, Ralf ; RÜCKERT, Ulrich: Robustness of Radial Basis Functions. In: *Neurocomputing* (2006). – to appear
- [ER07] EICKHOFF, Ralf ; RÜCKERT, Ulrich: Controlling complexity of RBF networks by similarity. In: VERLEYSSEN, Michel (Hrsg.): *Proceedings of the 15th European Symposium on Artificial Neural Networks (ESANN)*. Evere, Belgium : D-Side Publications, 25-27 April 2007, S. 181–186. – akzeptiert
- [ES07] EICKHOFF, Ralf ; SITTE, Joaquin: Significance measure of Local Cluster Neural Networks. In: *Proceedings of the 2007 International Joint Conference on Neural Networks (IJCNN)*. Orlando, Florida, USA, 12-17 aug 2007, S. on CD
- [ESR06] EICKHOFF, Ralf ; SITTE, Joaquin ; RÜCKERT, Ulrich: Robust Local Cluster Neural Networks. In: VERLEYSSEN, Michel (Hrsg.): *Proceedings of the 14th European Symposium on Artificial Neural Networks (ESANN)*. Evere, Belgium : D-Side Publications, 26-28 April 2006. – ISBN 2-930307-06-4, S. 119–124

## **Das Heinz Nixdorf Institut – Interdisziplinäres Forschungszentrum für Informatik und Technik**

Das Heinz Nixdorf Institut ist ein Forschungszentrum der Universität Paderborn. Es entstand 1987 aus der Initiative und mit Förderung von Heinz Nixdorf. Damit wollte er Ingenieurwissenschaften und Informatik zusammenzuführen, um wesentliche Impulse für neue Produkte und Dienstleistungen zu erzeugen. Dies schließt auch die Wechselwirkungen mit dem gesellschaftlichen Umfeld ein.

Die Forschungsarbeit orientiert sich an dem Programm „Dynamik, Mobilität, Vernetzung: Auf dem Weg zu den technischen Systemen von morgen“. In der Lehre engagiert sich das Heinz Nixdorf Institut in vielen Studiengängen der Universität. Hier ist das übergeordnete Ziel, den Studierenden die Kompetenzen zu vermitteln, auf die es in der Wirtschaft morgen ankommt.

Heute wirken am Heinz Nixdorf Institut sieben Professoren mit insgesamt 200 Mitarbeiterinnen und Mitarbeitern. Etwa ein Viertel der Forschungsprojekte der Universität Paderborn entfallen auf das Heinz Nixdorf Institut und pro Jahr promovieren hier etwa 30 Nachwuchswissenschaftlerinnen und Nachwuchswissenschaftler.

## **Heinz Nixdorf Institute – Interdisciplinary Research Centre for Computer Science and Technology**

The Heinz Nixdorf Institute is a research centre within the University of Paderborn. It was founded in 1987 initiated and supported by Heinz Nixdorf. By doing so he wanted to create a symbiosis of computer science and engineering in order to provide critical impetus for new products and services. This includes interactions with the social environment.

Our research is aligned with the program “Dynamics, Mobility, Integration: Enroute to the technical systems of tomorrow.” In training and education the Heinz Nixdorf Institute is involved in many programs of study at the University of Paderborn. The superior goal in education and training is to communicate competencies that are critical in tomorrows economy.

Today seven Professors and 200 researchers work at the Heinz Nixdorf Institute. The Heinz Nixdorf Institute accounts for approximately a quarter of the research projects of the University of Paderborn and per year approximately 30 young researchers receive a doctorate.



## Bände der HNI-Verlagsschriftenreihe

- Bd. 1 FAHRWINKEL, U.: Methoden zur Modellierung und Analyse von Geschäftsprozessen zur Unterstützung des Business Process Reengineering. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 1, 1995 – ISBN 3-931466-00-0
- Bd. 2 HORNBOSTEL, D.: Methode zur Modellierung der Informationsverarbeitung in Industrieunternehmen. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 2, 1995 – ISBN 3-931466-01-9
- Bd. 3 STEMANN, V.: Contention Resolution in Hashing Based Shared Memory Simulations. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 3, 1995 – ISBN 3-931466-02-7
- Bd. 4 KETTERER, N.: Beschreibung von Datenaustausch eines verteilten Fertigungssteuerungssystems. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 4, 1995 – ISBN 3-931466-03-5
- Bd. 5 HARTMANN, T.: Spezifikation und Klassifikation von Methoden zur Definition hierarchischer Abläufe. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 5, 1995 – ISBN 3-931466-04-3
- Bd. 6 WACHSMANN, A.: Eine Bibliothek von Basisdiensten für Parallelrechner: Routing, Synchronisation, gemeinsamer Speicher. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 6, 1995 – ISBN 3-931466-05-1
- Bd. 7 GAUSEMEIER, J. (Hrsg.): Die Szenariotechnik – Werkzeug für den Umgang mit einer multiplen Zukunft. 1. Paderborner Szenario-Workshop, 14. November 1995, HNI-Verlagsschriftenreihe, Paderborn, Band 7, 1995 – ISBN 3-931466-06-X
- Bd. 8 CZUMAJ, A.: Parallel Algorithmic Techniques: PRAM Algorithms and PRAM Simulations. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 8, 1995 – ISBN 3-931466-07-8
- Bd. 9 HUMPERT, A.: Methodische Anforderungsverarbeitung auf Basis eines objektorientierten Anforderungsmodells. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 9, 1995 – ISBN 3-931466-08-6
- Bd. 10 AMEUR, F.: Space-Bounded Learning Algorithms. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 10, 1995 – ISBN 3-931466-09-4
- Bd. 11 PAUL, M.: Szenariobasiertes Konzipieren neuer Produkte des Maschinenbaus auf Grundlage möglicher zukünftiger Technologieentwicklungen. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 11, 1996 – ISBN 3-931466-10-8
- Bd. 12 HOLL, F.: Ordnungsmäßigkeit von Informations- und Kommunikationssystemen. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 12, 1996 – ISBN 3-931466-11-6
- Bd. 13 GAUSEMEIER, J. (Hrsg.): First European Workshop on Global Engineering Networking - organized by GLENet e.V., HNI-Verlagsschriftenreihe, Paderborn, Band 13, 1996 – ISBN 3-931466-12-4
- Bd. 14 PETRI, K.: Vergleichende Untersuchung von Berechnungsmodellen zur Simulation der Dynamik von Fahrleitung-Stromabnehmer-Systemen. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 14, 1996 – ISBN 3-931466-13-2
- Bd. 15 LESCHKA, S.: Fallbasiertes Störungsmanagement in flexiblen Fertigungssystemen. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 15, 1996 – ISBN 3-931466-14-0
- Bd. 16 SCHNEIDER, U.: Ein formales Modell und eine Klassifikation für die Fertigungssteuerung - Ein Beitrag zur Systematisierung der Fertigungssteuerung. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 16, 1996 – ISBN 3-931466-15-9

---

Bezugsadresse:  
Heinz Nixdorf Institut  
Universität Paderborn  
Fürstenallee 11  
33102 Paderborn

## Bände der HNI-Verlagsschriftenreihe

- Bd. 17 FELSER, W.: Eine Methode zur Erstellung von Fertigungssteuerungsverfahren aus Bausteinen. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 17, 1996 – ISBN 3-931466-16-7
- Bd. 18 GAUSEMEIER, J.; ALEXANDER FINK, A.: Neue Wege zur Produktentwicklung – Erfolgspotentiale der Zukunft. HNI-Verlagsschriftenreihe, Paderborn, Band 18, 1996– ISBN 3-931466-17-5
- Bd. 19 DANGELMAIER, W.; GAUSEMEIER, J.: Fortgeschrittene Informationstechnologie in der Produktentwicklung und Fertigung. 2. Internationales Heinz Nixdorf Symposium, HNI-Verlagsschriftenreihe, Paderborn, Band 19, 1996 – ISBN 3-931466-18-3
- Bd. 20 HÜLLERMEIER, E.: Reasoning about Systems based on Incomplete and Uncertain Models. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 20, 1997 – ISBN 3-931466-19-1
- Bd. 21 GAUSEMEIER, J.: International Symposium on Global Engineering Network - Antwerp, Belgium, HNI-Verlagsschriftenreihe, Paderborn, Band 21, 1997 – ISBN 3-931466-20-5
- Bd. 22 BURGER, A.: Methode zum Nachweis der Wirtschaftlichkeit von Investitionen in die Rechnerintegrierte Produktion. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 22, 1997 – ISBN 3-931466-21-3
- Bd. 23 GAUSEMEIER, J.: Entwicklung und Transfer von Entwicklungssystemen der Mechatronik - Paderborner Workshop TransMechatronik. HNI-Verlagsschriftenreihe, Paderborn, Band 23, 1997 – ISBN 3-931466-22-1
- Bd. 24 GERDES, K.-H.: Architekturkonzeption für Fertigungsleitsysteme der flexiblen automatischen Fertigung. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 24, 1997 – ISBN 3-931466-23-X
- Bd. 25 EBBESMEYER, P.: Dynamische Texturwände - Ein Verfahren zur echtzeitorientierten Bildgenerierung für Virtuelle Umgebungen technischer Objekte. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 25, 1997 – ISBN 3-931466-24-8
- Bd. 26 FRANK, G.: Ein digitales Hardwaresystem zur echtzeitfähigen Simulation biologienaher neuronaler Netze. Dissertation, Fachbereich für Elektrotechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 26, 1997 – ISBN 3-931466-25-6
- Bd. 27 DITTRICH, W.: Communication and I/O Efficient Parallel Data Structures. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 27, 1997 – ISBN 3-931466-26-4
- Bd. 28 BÄUMKER, A.: Communication Efficient Parallel Searching. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 28 1997 – ISBN 3-931466-27-2
- Bd. 29 PINTASKE, C.: System- und Schaltungstechnik neuronaler Assoziativspeicher. Dissertation, Fachbereich für Elektrotechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 29, 1997 – ISBN 3-931466-28-0
- Bd. 30 HENKEL, S.: Ein System von Software-Entwurfsmustern für die Propagation von Ereignissen in Werkzeugen zur kooperativen Fabrikmodellierung. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 30, 1997 – ISBN 3-931466-29-9
- Bd. 31 DANGELMAIER, W.: Vision Logistik – Logistik wandelbarer Produktionsnetze. HNI-Verlagsschriftenreihe, Paderborn, Band 31, 1997 – ISBN 3-931466-30-2
- Bd. 32 BREXEL, D.: Methodische Strukturmodellierung komplexer und variantenreicher Produkte des integrativen Maschinenbaus. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 32, 1997 – ISBN 3-931466-31-0

## Bände der HNI-Verlagsschriftenreihe

- Bd. 33 HAHN, A.: Integrationsumgebung für verteilte objektorientierte Ingenieursysteme. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 33, 1997 – ISBN 3-931466-32-9
- Bd. 34 SABIN, A.: Semantisches Modell zum Aufbau von Hilfsorientierungsdiensten in einem globalen Engineering Netzwerk. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 34, 1997 – ISBN 3-931466-33-7
- Bd. 35 STROTHMANN, W.-B.: Bounded Degree Spanning Trees. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 35, 1997 – ISBN 3-931466-34-5
- Bd. 36 MÜLLER, W.; RAMMIG, F.-J.: Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen. HNI-Verlagsschriftenreihe, Paderborn, Band 36, 1998 – ISBN 3-931466-35-3
- Bd. 37 SCHNEIDER, W.: Anwenderorientierte Integration von CAE-Systemen. Ein Verfahren zur Realisierung eines durchgehenden Informationsflusses entlang des Produktentwicklungsprozesses. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 37, 1998 – ISBN 3-931466-36-1
- Bd. 38 DEMEL, W.; SCHMITZ, G. (Hrsg.): Entwicklung und Transfer von Entwicklungssystemen der Mechatronik. Aachener Workshop TransMechatronik, 26. Juni 1998, Technologiezentrum am Europaplatz Aachen, HNI-Verlagsschriftenreihe, Paderborn, Band 38, 1998 – ISBN 3-931466-37-X
- Bd. 39 GROBBEL, R.; LANGEMANN, T.: Leitfaden PPS-Systeme: Auswahl und Einführung in der Möbelindustrie. HNI-Verlagsschriftenreihe, Paderborn, Band 39, 1998 – ISBN 3-931466-38-8
- Bd. 40 REHBEIN, P.: Tribologische Untersuchung von hochfrequent schwingenden Gleitkontakten für den Einsatz in Reibkraftschlüssigen Antrieben. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 40, 1998 – ISBN 3-931466-39-6
- Bd. 41 DANGELMAIER, W.: KOMNET – Kommunikationsplattform für KMU-Netzwerke. HNI-Verlagsschriftenreihe, Paderborn, Band 41, 1998 – ISBN 3-931466-40-X
- Bd. 42 KALLMEYER, F.: Eine Methode zur Modellierung prinzipieller Lösungen mechatronischer Systeme. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 42, 1998 – ISBN 3-931466-41-8
- Bd. 43 TRAPP, R.: Stereoskopische Korrespondenzbestimmung mit impliziter Detektion von Okklusionen. Dissertation, Fachbereich für Elektrotechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 43, 1998 – ISBN 3-931466-42-6
- Bd. 44 GAUSEMEIER, J.; FINK, A.; SCHLAKE, O.: Grenzen überwinden - Zukünfte gestalten. 2. Paderborner Konferenz für Szenario-Management, HNI-Verlagsschriftenreihe, Paderborn, Band 44, 1998 – ISBN 3-931466-43-4
- Bd. 45 wird noch vergeben!
- Bd. 46 VÖCKING, B.: Static and Dynamic Data Management in Networks. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 46, 1998 – ISBN 3-931466-45-0
- Bd. 47 SCHEKELMANN, A.: Materialflußsteuerung auf der Basis des Wissens mehrerer Experten. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 47, 1999 – ISBN 3-931466-46-9
- Bd. 48 GECK-MÜGGE, K.: Herleitung und Spezifikation generischer Bausteine zur einheitlichen Modellierung von Fertigungsinformationen für die Fertigungssteuerung. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 48, 1999 – ISBN 3-931466-47-7
- Bd. 49 WALLASCHEK, J.; LÜCKEL, J.; LITTMANN, W.: Heinz Nixdorf Symposium on Mechatronics and Advanced Motion Control. 3. Internationales Heinz Nixdorf Symposium, HNI-Verlagsschriftenreihe, Paderborn, Band 49, 1999 – ISBN 3-931466-48-5

## Bände der HNI-Verlagsschriftenreihe

- Bd. 50 FINK, A.: Szenariogestützte Führung industrieller Produktionsunternehmen. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 50, 1999 – ISBN 3-931466-49-3
- Bd. 51 HOLTkamp, R.: Ein objektorientiertes Rahmenwerk zur Erstellung individueller, verteilter Fertigungslenkungssysteme. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 51, 1999 – ISBN 3-931466-50-7
- Bd. 52 KUHN, A.: Referenzmodelle für Produktionsprozesse zur Untersuchung und Gestaltung von PPS-Aufgaben. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 52, 1999 – ISBN 3-931466-51-5
- Bd. 53 SIEBE, A.: Systematik der Umsetzung von IT-orientierten Veränderungsprojekten in dynamischen Umfeldern. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 53, 1999 – ISBN 3-931466-52-3
- Bd. 54 KLAHOLD, R. F.: Dimensionierung komplexer Produktionsnetzwerke. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 54, 1999 – ISBN 3-931466-53-1
- Bd. 55 SCHÜRHOlz, A.: Synthese eines Modells zur simulationsgestützten Potentialanalyse der Distribution. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 55, 1999 – ISBN 3-931466-54-X
- Bd. 56 GEHNEN, G.: Integriertes Netzwerk zur Fertigungssteuerung und –automatisierung. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 56, 1999 – ISBN 3-931466-55-8
- Bd. 57 KRESS, S.: Architektur eines workflow-basierten Planungsinstruments für die technische Auftragsbearbeitung unter besonderer Berücksichtigung des Einsatzes der Telearbeit. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 57, 1999 – ISBN 3-931466-56-6
- Bd. 58 THIELEMANN, F.: Integrierte Methodik zur Gestaltung von Leistungserstellungsprozessen mittels Workflowmanagement. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 58, 1999 – ISBN 3-931466-57-4
- Bd. 59 KROME, J.: Modelle zur Untersuchung des Schwingungsverhaltens von Statoren für piezoelektrische Ultraschall-Wanderwellen-Motoren. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 59, 1999 – ISBN 3-931466-58-2
- Bd. 60 DEMEL, W.; SCHMITZ, G. (Hrsg.): Entwicklung und Transfer von Entwicklungssystemen der Mechatronik. Krefelder Workshop TransMechatronik, 24. August 1999 Fachhochschule Niederrhein, HNI-Verlagsschriftenreihe, Paderborn, Band 60, 1999 – ISBN 3-931466-59-0
- Bd. 61 LANGEMANN, T.: Modellierung als Kernfunktion einer systemorientierten Analyse und Bewertung der diskreten Produktion. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 61, 1999 – ISBN 3-931466-60-4
- Bd. 62 KÜMMEL, M.: Integration von Methoden und Werkzeugen zur Entwicklung von mechatronischen Systemen. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 62, 1999 – ISBN 3-931466-61-2
- Bd. 63 LUKOVszKI, T.: New Results on Geometric Spanners and Their Applications. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 63, 1999 – ISBN 3-931466-62-0
- Bd. 64 LÖFFLER, A.; MONDADA, F.; RÜCKERT, U. (Hrsg.): Experiments with the Mini-Robot Khepera, Proceedings of the 1st International Khepera Workshop. HNI-Verlagsschriftenreihe, Paderborn, Band 64, 1999 – ISBN 3-931466-63-9
- Bd. 65 SCHÄFERMEIER, U.; BISCHOFF, C.: KMUnet - Ein Konzept zur ablauforganisatorischen Gestaltung der Lieferanteneinbindung. HNI-Verlagsschriftenreihe, Paderborn, Band 65, 2000 – ISBN 3-931466-64-7

## Bände der HNI-Verlagsschriftenreihe

- Bd. 66 HOLTHÖFER, N.: Regeln in einer Mengenplanung unter Ausbringungsgrenzen. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 66, 2000 – ISBN 3-931466-69-8
- Bd. 67 SCHLAKE, O.: Verfahren zur kooperativen Szenario-Erstellung in Industrieunternehmen. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 67, 2000 – ISBN 3-931466-66-3
- Bd. 68 LEWANDOWSKI, A.: Methode zur Gestaltung von Leistungserstellungsprozessen in Industrieunternehmen. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 68, 2000 – ISBN 3-931466-67-1
- Bd. 69 SCHMIDTMANN, A.: Eine Spezifikations-sprache für die Fertigungslenkung. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 69, 2000 – ISBN 3-931466-68-X
- Bd. 70 GROBBEL, R.: Eine Referenzarchitektur für Kooperationsbörsen. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 70, 2000 – ISBN 3-931466-69-8
- Bd. 71 WESSEL, R.: Modelocked Waveguide Lasers in Lithium Niobate. Dissertation, Fachbereich für Physik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 71, 2000 – ISBN 3-931466-70-1
- Bd. 72 LÖFFLER, A.: Energetische Modellierung neuronaler Signalverarbeitung. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 72, 2000 – ISBN 3-931433-71-X
- Bd. 73 LUDWIG, L. A.: Computational Intelligence in der Produktionswirtschaft. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 73, 2000 – ISBN 3-931466-72-8
- Bd. 74 WENSKI, R.: Eine objektorientierte Systemkomponente zur Workflow-Modellierung und -Ausführung unter besonderer Berücksichtigung der Telekooperation. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 74, 2000 – ISBN 3-931466-73-6
- Bd. 75 GRASMANN, M.: Produktkonfiguration auf Basis von Engineering Data Management-Systemen. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 75, 2000 – ISBN 3-931466-74-4
- Bd. 76 DITZE, C.: Towards Operating System Synthesis. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 76, 2000 – ISBN 3-931466-75-2
- Bd. 77 KÖRNER, T.: Analog VLSI Implementation of a Local Cluster Neural Network. Dissertation, Fachbereich für Elektrotechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 77, 2000 – ISBN 3-931466-76-0
- Bd. 78 SCHEIDELER, C.: Probabilistic Methods for Coordination Problems. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 78, 2000 – ISBN 3-931466-77-9
- Bd. 79 GAUSEMEIER, J.; LINDEMANN, U.; REINHART, G.; WIENDAHL, H.-P.: Kooperatives Produktengineering - Ein neues Selbstverständnis des ingenieurmäßigen Wirkens. HNI-Verlagsschriftenreihe, Paderborn, Band 79, 2000 – ISBN 3-931466-78-7
- Bd. 80 GAUSEMEIER, J.; LÜCKEL, J.: Entwicklungsumgebungen Mechatronik - Methoden und Werkzeuge zur Entwicklung mechatronischer Systeme. HNI-Verlagsschriftenreihe, Paderborn, Band 80, 2000 – ISBN 3-931466-79-5
- Bd. 81 RIEPING, I.: Communication in Parallel Systems-Models, Algorithms and Implementations. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 81, 2000 – ISBN 3-931466-80-9

## Bände der HNI-Verlagsschriftenreihe

- Bd. 82 GAUSEMEIER, J; LÜCKEL, J.: Auf dem Weg zu den Produkten für die Märkte von morgen. 4. Internationales Heinz Nixdorf Symposium, HNI-Verlagsschriftenreihe, Paderborn, Band 82, 2000 – ISBN 3-931466-81-7
- Bd. 83 DEL CASTILLO, G.: The ASM Workbench - A Tool Environment for Computer-Aided Analysis and Validation of Abstract State Machine Models. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 83, 2000 – ISBN 3-931466-82-5
- Bd. 84 SCHÄFERMEIER, U.: Eine Methode zur systemorientierten organisatorischen Gestaltung der Zweckaufgabenverrichtung in kooperativen Verbänden; Klassifikation, Aufgabenzuordnung. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 84, 2000 – ISBN 3-931466-83-3
- Bd. 85 KRÜGER, J.: Ganzheitliche Beherrschung von Abläufen in und zwischen soziotechnischen Systemen: Ein Beitrag zur Modellbildung und zum paradigmatischen Verständnis von Industrieunternehmen zur Integration von Mensch und Maschine; Prozess und Struktur. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 85, 2000 – ISBN 3-931466-84-1
- Bd. 86 BARTSCHER, T.: Methoden des Integrierten Workflowmanagements (IWFm). Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 86, 2000 – ISBN 3-931466-85-X
- Bd. 87 QUINTANILLA, J.: Ein Verifikationsansatz für eine netzbasierte Modellierungsmethode für Fertigungssteuerungssysteme. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 87, 2000 – ISBN 3-931466-86-8
- Bd. 88 PREIS, R.: Analyses and Design of Efficient Graph Partitioning Methods. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 88, 2001 – ISBN 3-931466-87-6
- Bd. 89 wird noch vergeben!
- Bd. 90 WESTERMANN, M.: Caching in Networks: Non-Uniform Algorithms and Memory Capacity Constraints. Dissertation, Fachbereich für Informatik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 90, 2001 – ISBN 3-931466-89-2
- Bd. 91 LEMKE, J.: Nutzenorientierte Planung des Einsatzes von CAD- / CAE-Systemen. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 91, 2001 – ISBN 3-935433-00-X
- Bd. 92 VON BOHUSZEWICZ, O.: Eine Methode zur Visualisierung von Geschäftsprozessen in einer virtuellen Umgebung. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 92, 2001 – ISBN 3-935433-01-8
- Bd. 93 BÖRNCHEN, T.: Zur Entwicklung dynamischer Komponenten für variables Kraftfahrzeug-Scheinwerferlicht. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 93, 2001 – ISBN 3-935433-02-6
- Bd. 94 WINDELER, I.: Auswahl von Restrukturierungsprojekten in Forschungs- und Entwicklungsorganisationen der Automobilindustrie. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 94, 2001 – ISBN 3-935433-03-4
- Bd. 95 WOLFF, C.: Parallele Simulation großer pulscodierter neuronaler Netze. Dissertation, Fachbereich für Elektrotechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 95, 2001 – ISBN 3-935433-04-2
- Bd. 96 HENKE, A.: Modellierung, Simulation und Optimierung piezoelektrischer Stellsysteme. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 96, 2001 – ISBN 3-935433-05-0
- Bd. 97 RÜCKERT, U.; SITTE, J.; WITKOWSKI, U. (Hrsg.): Autonomous Minirobots for Research and Edutainment AMiRE2001. 5. Internationales Heinz Nixdorf Symposium, HNI-Verlagsschriftenreihe, Paderborn, Band 97, 2001 – ISBN 3-935433-06-9

## Bände der HNI-Verlagsschriftenreihe

- Bd. 98 LI, P.: Datenkonversion für den Datenaustausch in verteilten Fertigungs-Lenkungssystemen. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 98, 2001 – ISBN 9-935433-07-7
- Bd. 99 BRANDT, C.: Eine modellbasierte Methode zum strukturierten Entwurf virtueller Umgebungen. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 99, 2001 – ISBN 9-935433-08-5
- Bd. 100 WLEKLINSKI, C.: Methode zur Effektivitäts- und Effizienzbewertung der Entwicklung maschinenbaulicher Anlagen. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 100, 2001 – ISBN-3-935433-09-3
- Bd. 101 HEMSEL, T.: Untersuchung und Weiterentwicklung linearer piezoelektrischer Schwingungsantriebe. Dissertation, Fachbereich für Maschinentechnik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 101, 2001 – ISBN 3-935433-10-7
- Bd. 102 MAUERMANN, H.: Leitfaden zur Erhöhung der Logistikqualität durch Analyse und Neugestaltung der Versorgungsketten. Dissertation, Fachbereich für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 102, 2001 – ISBN 3-935433-11-5
- Bd. 103 WAGENBLAßT, D.: Eine Analyse- und Beurteilungsmethode zur Beurteilung der Funktionssicherheit von gemischt analog-digitalen Schaltungen. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 103, 2002 – ISBN 3-935433-12-3
- Bd. 104 PORRMANN, M.: Leistungsbewertung eingebetteter Neurocomputersysteme. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 104, 2002 – ISBN 3-935433-13-1
- Bd. 105 SEIFERT, L.: Methodik zum Aufbau von Informationsmodellen für Electronic Business in der Produktentwicklung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 105, 2002 – ISBN 3-935433-14-X
- Bd. 106 SOETEBEER, M.: Methode zur Modellierung, Kontrolle und Steuerung von Produktstrategien. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 106, 2002 – ISBN 3-935433-15-8
- Bd. 107 GAUSEMEIER, J.; GRAFE, M. (Hrsg.): 1. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung. HNI-Verlagsschriftenreihe, Paderborn, Band 107, 2002 – ISBN 3-935433-16-6
- Bd. 108 FLATH, M.: Methode zur Konzipierung mechatronischer Produkte. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 108, 2002 – ISBN 3-935433-17-4
- Bd. 109 AVENARIUS, J.: Methoden zur Suche und Informationsbereitstellung von Lösungselementen für die Entwicklung mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 109, 2002 – ISBN 3-935433-18-2
- Bd. 110 HELMKE, S.: Eine simulationsgestützte Methode für Budgetentscheidungen im Kundenbindungsmanagement. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 110, 2002 – ISBN 3-935433-19-0
- Bd. 111 CZUBAYKO, R.: Rechnerinterne Repräsentation von informationsverarbeitenden Lösungselementen für die verteilte kooperative Produktentwicklung in der Mechatronik. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 111, 2002 – ISBN 3-935433-20-4
- Bd. 112 GOLDSCHMIDT, S.: Anwendung mengenorientierter numerischer Methoden zur Analyse nichtlinearer dynamischer Systeme am Beispiel der Spurführungsdynamik von Schienenfahrzeugen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 112, 2002 – ISBN 3-935433-21-2

## Bände der HNI-Verlagsschriftenreihe

- Bd. 113 LEHMANN, T.: Towards Device Driver Synthesis. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 113, 2002 – ISBN 3-935433-22-0
- Bd. 114 HÄRTEL, W.: Issueorientierte Frühaufklärung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 114, 2002 – ISBN 3-935433-23-9
- Bd. 115 ZIEGLER, M.: Zur Berechenbarkeit reeller geometrischer Probleme. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 115, 2002 – ISBN 3-935433-24-7
- Bd. 116 SCHMIDT, M.: Neuronale Assoziativspeicher im Information Retrieval. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 116, 2003 – ISBN 3-935433-25-5
- Bd. 117 EL-KEBBE, D. A.: Towards the MaSHReC Manufacturing System under Real-Time Constraints. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 117, 2003 – ISBN 3-935433-26-3
- Bd. 118 PUSCH, R.: Personalplanung und -entwicklung in einem integrierten Vorgehensmodell zur Einführung von PDM-Systemen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 118, 2003 – ISBN 3-935433-27-1
- Bd. 119 SOHLER, C.: Property Testing and Geometry. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 119, 2003 – ISBN 3-935433-28-X
- Bd. 120 KESPOHL, H.: Dynamisches Matching – Ein agentenbasiertes Verfahren zur Unterstützung des Kooperativen Produktengineering durch Wissens- und Technologietransfer. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 120, 2003 – ISBN 3-935433-29-8
- Bd. 121 MOLT, T.: Eine domänenübergreifende Softwarespezifikationstechnik für automatisierte Fertigungsanlagen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 121, 2003 – ISBN 3-935433-30-1
- Bd. 122 GAUSEMEIER, J.; LÜCKEL, J.; WALLASCHEK, J. (Hrsg.): 1. Paderborner Workshop Intelligente mechatronische Systeme. HNI-Verlagsschriftenreihe, Paderborn, Band 122, 2003 – ISBN 3-935433-31-X
- Bd. 123 GAUSEMEIER, J.; GRAFE, M. (Hrsg.): 2. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung. HNI-Verlagsschriftenreihe, Paderborn, Band 123, 2003 – ISBN 3-935433-32-8
- Bd. 124 LITTMANN, W.: Piezoelektrische resonant betriebene Ultraschall-Leistungswandler mit nichtlinearen mechanischen Randbedingungen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 124, 2003 – ISBN 3-935433-33-6
- Bd. 125 WICKORD, W.: Zur Anwendung probabilistischer Methoden in den frühen Phasen des Systementwurfs. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 125, 2003 – ISBN 3-935433-34-4
- Bd. 126 HEITTMANN, A.: Ressourceneffiziente Architekturen neuronaler Assoziativspeicher. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 126, 2003 – ISBN 3-935433-35-2
- Bd. 127 WITKOWSKI, U.: Einbettung selbstorganisierender Karten in autonome Miniroboter. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 127, 2003 – ISBN 3-935433-36-0
- Bd. 128 BOBDA, C.: Synthesis of Dataflow Graphs for Reconfigurable Systems using Temporal Partitioning and Temporal Placement. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 128, 2003 – ISBN 3-935433-37-9

## Bände der HNI-Verlagsschriftenreihe

- Bd. 129 HELLER, F.: Wissensbasiertes Online-Störungsmanagement flexibler, hoch automatisierter Montagesysteme. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 129, 2003 – ISBN 3-935433-38-7
- Bd. 130 KÜHN, A.: Systematik des Ideenmanagements im Produktentstehungsprozess. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 130, 2003 – ISBN 3-935433-39-5
- Bd. 131 KEIL-SLAWIK, R.; BRENECKE, A.; HOHENHAUS, M.: ISIS -Installationshandbuch für lernförderliche Infrastrukturen. HNI-Verlagsschriftenreihe, Paderborn, Band 131, 2003 – ISBN 3-935433-40-9
- Bd. 132 OULD HAMADY, M.: Ein Ansatz zur Gestaltung des operativen Fertigungsmanagements innerhalb der Lieferkette. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 132, 2003 – ISBN 3-935433-41-7
- Bd. 133 HOLTZ, C.: Theoretical Analysis of Unsupervised On-line Learning through Soft Competition. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 133, 2003 – ISBN 3-935433-42-5
- Bd. 134 UEBEL, M.: Ein Modell zur Steuerung der Kundenbearbeitung im Rahmen des Vertriebsmanagements. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 134, 2003 – ISBN 3-935433-43-3
- Bd. 135 BRINKMANN, A.: Verteilte Algorithmen zur Datenplatzierung und zum Routing in gegnerischen Netzwerken. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 135, 2003 – ISBN 3-935433-44-1
- Bd. 136 FRÜND, E.: Aktive Kompensation von periodischen Schwingungen an rotierenden Walzen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 136, 2003 – ISBN 3-935433-45-X
- Bd. 137 KEIL-SLAWIK, R. (Hrsg.): Digitale Medien in der Hochschule: Infrastrukturen im Wandel. HNI-Verlagsschriftenreihe, Paderborn, Band 137, 2004 – ISBN 3-935433-46-8
- Bd. 138 STORCK, H.: Optimierung der Kontaktvorgänge bei Wanderwellenmotoren. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 138, 2004 – ISBN 3-935433-47-6
- Bd. 139 KALTE, H.: Einbettung dynamisch rekonfigurierbarer Hardwarearchitekturen in eine Universalprozessorumgebung. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 139, 2004 – ISBN 3-935433-48-4
- Bd. 140 ISKE, B.: Modellierung und effiziente Nutzung aktiver Infrarotsensorik in autonomen Systemen. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 140, 2004 – ISBN 3-935433-49-2
- Bd. 141 BÄTZEL, D.: Methode zur Ermittlung und Bewertung von Strategiealternativen im Kontext Fertigungstechnik. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 141, 2004 – ISBN 3-935433-50-6
- Bd. 142 BÖKE, C.: Automatic Configuration of Real-Time Operating Systems and Real-Time Communication Systems for Distributed Embedded Applications. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 142, 2004 – ISBN 3-935433-51-4
- Bd. 143 KÖCKERLING, M.: Methodische Entwicklung und Optimierung der Wirkstruktur mechatronischer Produkte. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 143, 2004 – ISBN 3-935433-52-2
- Bd. 144 HENZLER, S.: Methodik zur Konzeption der Struktur und der Regelung leistungsverzweigter Getriebe mit Toroidvariator. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 144, 2004 – ISBN 3-935433-53-0

## Bände der HNI-Verlagsschriftenreihe

- Bd. 145 GAUSEMEIER, J.; LÜCKEL, J.; WALLASCHEK, J. (Hrsg.): 2. Paderborner Workshop Intelligente mechatronische Systeme. HNI-Verlagsschriftenreihe, Paderborn, Band 145, 2004 – ISBN 3-935433-54-9
- Bd. 146 LESSING, H.: Prozess zur multivariaten Prognose von Produktionsprogrammen für eine effiziente Kapazitätsplanung bei typisierten Dienstleistungen. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 146, 2004 – ISBN 3-935433-55-7
- Bd. 147 HAMOUDIA, H.: Planerische Ablaufgestaltung bei prozessorientierten Dienstleistungen. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 147, 2004 – ISBN 3-935433-56-5
- Bd. 148 BUSCH, A.: Kollaborative Änderungsplanung in Unternehmensnetzwerken der Serienfertigung – ein verhandlungsbasierter Ansatz zur interorganisationalen Koordination bei Störungen. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 148, 2004 – ISBN 3-935433-57-3
- Bd. 149 GAUSEMEIER, J.; GRAFE, M. (Hrsg.): 3. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung. HNI-Verlagsschriftenreihe, Paderborn, Band 149, 2004 – ISBN 3-935433-58-1
- Bd. 150 MEYER, B.: Value-Adding Logistics for a World Assembly Line. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 150, 2004 – ISBN 3-935433-59-X
- Bd. 151 GRIENITZ, V.: Methodik zur Erstellung von Technologieszenarien für die strategische Technologieplanung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 151, 2004 – ISBN 3-9354 33-60-3
- Bd. 152 FRANKE, H.: Eine Methode zur unternehmensübergreifenden Transportdisposition durch synchron und asynchron kommunizierende Agenten. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 152, 2004 – ISBN 3-9354 33-61-1
- Bd. 153 SALZWEDEL, K. A.: Data Distribution Algorithms for Storage Networks. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 153, 2004 – ISBN 3-935433-62-X
- Bd. 154 RÄCKE, H.: Data Management and Routing in General Networks. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 154, 2004 – ISBN 3-935433-63-8
- Bd. 155 FRANK, U.; GIESE, H.; KLEIN, F.; OBERSCHELP, O.; SCHMIDT, A.; SCHULZ, B.; VÖCKING, H.; WITTING, K.; GAUSEMEIER, J. (Hrsg.): Selbstoptimierende Systeme des Maschinenbaus – Definitionen und Konzepte. HNI-Verlagsschriftenreihe, Paderborn, Band 155, 2004 – ISBN 3-935433-64-6
- Bd. 156 MÖHRINGER, S.: Entwicklungsmethodik für mechatronische Systeme. Habilitation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 156, 2004 – ISBN 3-935433-65-4
- Bd. 157 FAHRENTHOLZ, M.: Konzeption eines Betriebskonzepts für ein bedarfsgesteuertes schienengebundenes Shuttle-System. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 157, 2004 – ISBN 3-935433-66-2
- Bd. 158 GAJEWSKI, T.: Referenzmodell zur Beschreibung der Geschäftsprozesse von After-Sales-Dienstleistungen unter besonderer Berücksichtigung des Mobile Business. Dissertation Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 158, 2004 – ISBN 3-935433-67-0
- Bd. 159 RÜTHER, M.: Ein Beitrag zur klassifizierenden Modularisierung von Verfahren für die Produktionsplanung. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 159, 2004 – ISBN 3-935433-68-9

## Bände der HNI-Verlagsschriftenreihe

- Bd. 160 MUECK, B.: Eine Methode zur benutzerstimulierten detaillierungsvarianten Berechnung von diskreten Simulationen von Materialflüssen. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 160, 2004 – ISBN 3-935433-69-7
- Bd. 161 LANGEN, D.: Abschätzung des Ressourcenbedarfs von hochintegrierten mikroelektronischen Systemen. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 161, 2005 – ISBN 3-935433-70-0
- Bd. 162 ORLIK, L.: Wissensbasierte Entscheidungshilfe für die strategische Produktplanung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 162, 2005 – ISBN 3-935433-71-9
- Bd. 163 GAUSEMEIER, J.; RAMMIG, F.; SCHÄFER, W.; WALLASCHEK, J. (Hrsg.): 3. Paderborner Workshop Intelligente mechatronische Systeme. HNI-Verlagsschriftenreihe, Paderborn, Band 163, 2005 – ISBN 3-935433-72-7
- Bd. 164 FISCHER, M.: Design, Analysis, and Evaluation of a Data Structure for Distributed Virtual Environments. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 164, 2005 – ISBN 3-935433-73-5
- Bd. 165 MATYSCZOK, C.: Dynamische Kantenextraktion - Ein Verfahren zur Generierung von Tracking-Informationen für Augmented Reality-Anwendungen auf Basis von 3D-Referenzmodellen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 165, 2005 – ISBN 3-935433-74-3
- Bd. 166 JANIA, T.: Änderungsmanagement auf Basis eines integrierten Prozess- und Produktdatenmodells mit dem Ziel einer durchgängigen Komplexitätsbewertung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 166, 2005 – ISBN 3-935433-75-1
- Bd. 167 GAUSEMEIER, J.; GRAFE, M. (Hrsg.): 4. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung. HNI-Verlagsschriftenreihe, Paderborn, Band 167, 2005 – ISBN 3-935433-76-X
- Bd. 168 VOLBERT, K.: Geometric Spanners for Topology Control in Wireless Networks. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 168, 2005 – ISBN 3-935433-77-8
- Bd. 169 ROSLAK, J.: Entwicklung eines aktiven Scheinwerfersystems zur blendungsfreien Ausleuchtung des Verkehrsraumes. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 167, 2005 – ISBN 3-935433-78-6
- Bd. 170 EMMRICH, A.: Ein Beitrag zur systematischen Entwicklung produktorientierter Dienstleistungen. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 170, 2005 – ISBN 3-935433-79-4
- Bd. 171 NOWACZYK, O.: Explorationen: Ein Ansatz zur Entwicklung hochgradig interaktiver Lernbausteine. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 171, 2005 – ISBN 3-935433-80-8
- Bd. 172 MAHMOUD, K.: Theoretical and experimental investigations on a new adaptive duo servo drum brake with high and constant brake shoe factor. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 172, 2005 – ISBN 3-935433-81-6
- Bd. 173 KLIEWER, G.: Optimierung in der Flugplanung: Netzwerkentwurf und Flottenzuweisung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 173, 2005 – ISBN 3-935433-82-4
- Bd. 174 BALÁŽOVÁ, M.: Methode zur Leistungsbewertung und Leistungssteigerung der Mechatronikentwicklung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 174, 2005 – ISBN 3-935433-83-2
- Bd. 175 FRANK, U.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 175, 2005 – ISBN 3-935433-84-0

## Bände der HNI-Verlagsschriftenreihe

- Bd. 176 BERGER, T.: Methode zur Entwicklung und Bewertung innovativer Technologiestrategien. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 176, 2005 – ISBN 3-935433-85-9
- Bd. 177 BERSSENBRÜGGE, J.: Virtual Nightdrive - Ein Verfahren zur Darstellung der komplexen Lichtverteilungen moderner Scheinwerfersysteme im Rahmen einer virtuellen Nachtfahrt. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 177, 2005 – ISBN 3-935433-86-7
- Bd. 178 GAUSEMEIER, J. (Hrsg.): Vorausschau und Technologieplanung. 1. Symposium für Vorausschau und Technologieplanung Heinz Nixdorf Institut, 3. und 4. November 2005, Schloß Neuhardenberg, HNI-Verlagsschriftenreihe, Paderborn, Band 178, 2005 – ISBN 3-935433-87-5
- Bd. 179 FU, B.: Piezoelectric actuator design via multiobjective optimization methods. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 179, 2005 – ISBN 3-935433-88-3
- Bd. 180 WALLASCHEK, J.; HEMSEL, T.; MRACEK, M.: Proceedings of the 2nd International Workshop on Piezoelectric Materials and Applications in Actuators. HNI-Verlagsschriftenreihe, Paderborn, Band 180, 2005 – ISBN 3-935433-89-1
- Bd. 181 MEYER AUF DER HEIDE, F.; MONIEN, B. (Hrsg.): New Trends in Parallel & Distributed Computing. 6. Internationales Heinz Nixdorf Symposium, 17. und 18. Januar 2006, HNI-Verlagsschriftenreihe, Paderborn, Band 181, 2006 – ISBN 3-939350-00-1
- Bd. 182 HEIDENREICH, J.: Adaptierbare Änderungsplanung der Mengen und Kapazitäten in Produktionsnetzwerken der Serienfertigung. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 182, 2006 – ISBN 3-939350-01-X
- Bd. 183 PAPE, U.: Umsetzung eines SCM-Konzeptes zum Liefermanagement in Liefernetzwerken der Serienfertigung. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 183, 2006 – ISBN 3-939350-02-8
- Bd. 184 BINGER, V.: Konzeption eines wissensbasierten Instruments für die strategische Vorausschau im Kontext der Szenariotechnik. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 184, 2006 – ISBN 3-939350-03-6
- Bd. 185 KRIESEL, C.: Szenarioorientierte Unternehmensstrukturoptimierung – Strategische Standort- und Produktionsplanung. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 185, 2006 – ISBN 3-939350-04-4
- Bd. 186 KLEIN, J.: Efficient collision detection for point and polygon based models. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 186, 2006 – ISBN 3-939350-05-2
- Bd. 187 WORTMANN, R.: Methodische Entwicklung von Echtzeit 3D-Anwendungen für Schulung und Präsentation. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 187, 2006 – ISBN 3-939350-06-0
- Bd. 188 GAUSEMEIER, J.; GRAFE, M. (Hrsg.): 5. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung. HNI-Verlagsschriftenreihe, Paderborn, Band 188, 2006 – ISBN 3-939350-07-9
- Bd. 189 GAUSEMEIER, J.; RAMMIG, F.; SCHÄFER, W.; TRÄCHTLER, A.; WALLASCHEK, J. (Hrsg.): 4. Paderborner Workshop Entwurf mechatronischer Systeme. HNI-Verlagsschriftenreihe, Paderborn, Band 189, 2006 – ISBN 3-939350-08-7
- Bd. 190 DAMEROW, V.: Average and Smoothed Complexity of Geometric Structures. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 190, 2006 – ISBN 3-939350-09-5
- Bd. 191 GIESE, H.; NIGGEMANN, O. (Hrsg.): Postworkshop Proceedings of the 3rd Workshop on Object-oriented Modeling of Embedded Real-Time Systems (OMER 3), HNI-Verlagsschriftenreihe, Paderborn, Band 191, 2006 – ISBN 3-939350-10-9

## Bände der HNI-Verlagsschriftenreihe

- Bd. 192 RADKOWSKI, R.: Anwendung evolutionärer Algorithmen zur Unterstützung des Entwurfs selbstoptimierender Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 192, 2006 – ISBN 3-939350-11-7
- Bd. 193 SHEN, Q.: A Method for Composing Virtual Prototypes of Mechatronic Systems in Virtual Environments. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 193, 2006 – ISBN 3-939350-12-5
- Bd. 194 REDENIUS, A.: Verfahren zur Planung von Entwicklungsprozessen für fortgeschrittene mechatronische Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 194, 2006 – ISBN 3-939350-13-3
- Bd. 195 KÜHL, P.: Anpassung der Lichtverteilung des Abblendlichtes an den vertikalen Straßenverlauf. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 195, 2006 – ISBN 3-939350-14-1
- Bd. 196 MICHELS, J. S.: Integrative Spezifikation von Produkt- und Produktionssystemkonzeptionen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 196, 2006 – ISBN 3-939350-15-X
- Bd. 197 RIPS, S.: Adaptive Steuerung der Lastverteilung datenparalleler Anwendungen in Grid-Umgebungen. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 197, 2006 – ISBN 3-939350-16-8
- Bd. 198 GAUSEMEIER, J. (Hrsg.): Vorausschau und Technologieplanung. 2. Symposium für Vorausschau und Technologieplanung Heinz Nixdorf Institut, 9. und 10. November 2006, Schloß Neuhardenberg, HNI-Verlagsschriftenreihe, Paderborn, Band 198, 2006 – ISBN 3-939350-17-6
- Bd. 199 FRANKE, W.: Wiederverwendungsorientierte Herleitung von Inter-Fachkomponentenkonzepten für Lagerverwaltungssysteme. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 199, 2006 – ISBN 978-3-939350-18-7
- Bd. 200 SCHEIDELER, P.: Ein Beitrag zur erfahrungsbasierten Selbstoptimierung einer Menge technisch homogener fahrerloser Fahrzeuge. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 200, 2006 – ISBN 978-3-939350-19-4
- Bd. 201 KÖSTERS, C.: Ein ontologiebasiertes Modell zur Beschreibung der Abläufe in einem Produktionssystem unter besonderer Berücksichtigung einer diskreten Produktion. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 201, 2006 – ISBN 978-3-939350-20-0
- Bd. 202 HALFMEIER, S.: Modellierung und Regelung von Halbtoroidvariationen in leistungsverzweigten Getriebestrukturen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 202, 2006 – ISBN 978-3-939350-21-7
- Bd. 203 RÜHRUP, S.: Position-based Routing Strategien. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 203, 2006 – ISBN 978-3-939350-22-4
- Bd. 204 SCHMIDT, A.: Wirkungsmuster zur Selbstoptimierung – Konstrukte für den Entwurf selbstoptimierender Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 204, 2006 – ISBN 978-3-939350-23-1
- Bd. 205 IHMOR, S.: Modeling and Automated Synthesis of Reconfigurable Interfaces. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 205, 2006 – ISBN 978-3-939350-24-8

## Bände der HNI-Verlagsschriftenreihe

- Bd. 206 ECKES, R.: Augmented Reality – basiertes Verfahren zur Unterstützung des Anlaufprozesses von automatisierten Fertigungssystemen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 206, 2007 – ISBN 978-3-939350-25-5
- Bd. 207 STEFFEN, D.: Ein Verfahren zur Produktstrukturierung für fortgeschrittene mechatronische Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 207, 2007 – ISBN 978-3-939350-26-2
- Bd. 208 LAROQUE, C.: Ein mehrbenutzerfähiges Werkzeug zur Modellierung und richtungsoffenen Simulation von wahlweise objekt- und funktionsorientiert gegliederten Fertigungssystemen. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 208, 2007 – ISBN 978-3-939350-27-9
- Bd. 209 GAUSEMEIER, J.; GRAFE, M. (Hrsg.): 6. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung. HNI-Verlagsschriftenreihe, Paderborn, Band 209, 2007 – ISBN 978-3-939350-28-6
- Bd. 210 GAUSEMEIER, J.; RAMMIG, F.; SCHÄFER, W.; TRÄCHTLER, A.; WALLASCHEK, J. (Hrsg.): 5. Paderborner Workshop Entwurf mechatronischer Systeme. HNI-Verlagsschriftenreihe, Paderborn, Band 210, 2007 – ISBN 978-3-939350-29-3
- Bd. 211 KAUSCHKE, R.: Systematik zur lichttechnischen Gestaltung von aktiven Scheinwerfern. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 211, 2007 – ISBN 978-3-939350-30-9
- Bd. 212 DU, J.: Zellen-basierte Dienst-Entdeckung für Roboternetzwerke. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 212, 2007 – ISBN 978-3-939350-31-6
- Bd. 213 DANNE, K.: Real-Time Multitasking in Embedded Systems Based on Reconfigurable Hardware. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 213, 2007 – ISBN 978-3-939350-32-3

---

Bezugsadresse:  
Heinz Nixdorf Institut  
Universität Paderborn  
Fürstenallee 11  
33102 Paderborn