

Multivariate GRBF-Netzwerke und Systeme lokaler Experten: Neue Konzepte und ihre Anwendung bei komplexen Regelungsaufgaben

**Dissertation der Fakultät für Physik
der Ludwig-Maximilians-Universität München**

**vorgelegt von
Sebastian Albrecht
aus Hamburg**

27. September 1999

Erster Gutachter: Prof. Dr. P. Tavan
Zweiter Gutachter: Prof. Dr. H. Lesch
Tag der mündlichen Prüfung: 24. Februar 2000

Zusammenfassung

Solange der Mensch seit Beginn der modernen Wissenschaft versucht, seine kognitiven Fähigkeiten durch anatomische, physiologische und psychologische Untersuchungen zu verstehen, werden diese Forschungen auch von der Entwicklung mathematischer Modelle begleitet. Dies geschieht in der Hoffnung, zu einem tieferen Verständnis der Gehirnfunktionen zu gelangen und in jüngster Zeit mit dem Ziel, neuartige mathematische Verfahren, z.B. zur Mustererkennung und Funktionenapproximation, zu erhalten. Im Rahmen dieses Ansatzes wurde vor etwa 10 Jahren das *radiale Basisfunktionen* (RBF)-Netzwerk eingeführt, welches bestimmte Strukturen im cerebellaren Cortex modelliert.

In früheren Arbeiten wurden tiefgehende Beziehungen zwischen diesem dreischichtigen Netzwerkmodell und der *maximum likelihood* (ML)-Schätzung von empirischen Datenverteilungen durch Mischungen *univariater* Normalverteilungen aufgedeckt. Solche Netzwerke eignen sich zur datengetriebenen Funktionenapproximation und zur Lösung von Klassifikationsaufgaben. Ausgehend von diesen Beobachtungen wird in der vorliegenden Arbeit das RBF-Modell stufenweise verallgemeinert.

Zunächst wird mit dem generalisierten radialen Basisfunktionen (GRBF)-Netzwerk ein Modell vorgestellt, dessen Parameter sich aus ML-Schätzungen von Datenverteilungen durch Mischungen *multivariater* Normalverteilungen ableiten lassen. Damit wird erstmals ein Verfahren eingeführt, mit dem alle Netzwerkparameter simultan optimiert werden können. Ein deterministisches Abkühlschema sorgt dabei für die sichere Konvergenz des zugehörigen sequentiellen stochastischen Lernprozesses.

Anschließend wird ein neues Modell zur Funktionenapproximation, der sogenannte *LLM-Approximator*, vorgestellt, das ebenfalls auf Dichteschätzungen durch Mischungen multivariater Normalverteilungen beruht und sich in Spezialfällen auf das GRBF-Netzwerk reduziert. Im LLM-Verfahren wird die zu approximierende Funktion durch eine Interpolation lokaler linearer Regressionsmodelle dargestellt. In Verallgemeinerung dieser Verfahren wird schließlich ein Konstruktionsprinzip für *Systeme lokaler Experten* formuliert, das sowohl Wettbewerb als auch Kooperation unterschiedlicher Experten zur Lösung einer gemeinsamen Aufgabe organisiert.

Die Arbeitsweisen des LLM-Approximators als auch des Systems lokaler Experten werden am Beispiel von Regelungsproblemen illustriert. Zunächst wird die Regelung eines virtuellen Bioreaktors mit Hilfe des LLM-Approximators vorgestellt. Anschließend wird das System lokaler Experten für die Regelung einer realen, komplexen industriellen Anlage verwendet. Dabei handelt es sich um die Anlage zur Rückstandsverbrennung im Werk Burghausen der Wacker-Chemie GmbH.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motorische Kontrolle	1
1.2. Industrielle Regelung	5
1.3. Ein neuronaler Ansatz zur industriellen Kontrolle	8
1.4. Aufbau der Arbeit	10
2. Gehirn, Datenanalyse und neuronale Netzwerke	13
2.1. Überblick	13
2.2. Gehirn und Kleinhirnrinde	15
2.2.1. Nervenzellen und Synapsen	15
2.2.2. Kleinhirnrinde	19
2.3. Datenanalyse	22
2.3.1. Funktionenapproximation	22
2.3.2. Mustererkennung	27
2.3.3. Dichteschätzung	35
2.4. Künstliche neuronale Netze	43
2.4.1. Perzeptron	46
2.4.2. GRBF-Netzwerk	49
3. GRBF-Netzwerke, Dichteschätzung und Funktionenapproximation	57
3.1. Erweiterung des GRBF-Netzwerks	58
3.2. Lernregeln für das multivariate GRBF-Netzwerk	62
3.2.1. Durchführung des Lernverfahrens	67
3.3. Funktionenapproximation	70
3.4. Diskussion und Beispiele	73

4. Systeme lokaler Experten	83
4.1. Warum sind lokale Experten vorteilhaft?	84
4.2. Verallgemeinerung des GRBF-Netzwerks auf ein System lokaler Experten	85
4.3. Approximation einer multifunktionalen Datenfolge	90
4.3.1. mGRBF-System für eine multifunktionale Datenfolge	91
4.3.2. LLM-Approximator für eine multifunktionale Datenfolge	93
5. Regelungstheorie	97
5.1. Grundbegriffe der Regelungstheorie	97
5.2. Modellbasierte Regelung mit einem PID-Regler	99
5.3. Erfahrungsbasierte Regelung mit einem <i>Fuzzy</i> -Regler	103
5.4. Datenbasierte Regelung mit neuronalen Netzen	109
6. Regelung eines Bioreaktors im Computerexperiment	115
6.1. Dynamik des Bioreaktors	115
6.2. Regelung des Bioreaktors	121
6.2.1. Simulation des Bioreaktors	121
6.2.2. Kriterien zur Beurteilung einer Regelung	122
6.2.3. Drei Regler für den Bioreaktor	124
6.3. Ergebnisse für verschiedene Szenarien und Regler	128
7. Regelung der Rückstandsverbrennung in einer industriellen Anlage	133
7.1. Die Anlage zur Rückstandsverbrennung	134
7.1.1. Öfen und Brennstrecken	135
7.1.2. Randbedingungen und Sicherheitsaspekte	138
7.1.3. Prozeßleitsystem der Rückstandsverbrennung	139
7.1.4. Probleme bei der Regelung der Rückstandsverbrennung	140
7.2. Entwurf einer neuen Regelung	145
7.2.1. Zielsetzung und Konzeption	145
7.2.2. Programm PCS zur Regelung industrieller Prozesse	148

7.2.3. Experten zur Regelung der Flüssigkeitszufuhr	151
7.2.4. Zwei spezielle, bedarfsorientierte Experten	152
7.2.5. Experten zur Regelung der Abgaszufuhr	158
7.3. Beispiele und Diskussion	159
A. Anhang	165
A.1. Der LLM-Approximator	165
A.1.1. Herleitung des LLM-Approximators	165
A.1.2. Klassifikation mit einem LLM-Approximator	168
A.2. Dynamik des Bioreaktors	169
A.2.1. Fixpunkte des Bioreaktors	169
A.2.2. Stabilitätsanalyse der Fixpunkte	171
Literaturverzeichnis	175

1. Einleitung



Mit nur wenigen Strichen skizziert Sempé in der obigen Abbildung das Phänomen des Lernens [Sem85]. Das Kind muß seine ganze Aufmerksamkeit auf die neue Bewegungsform des Radfahrens richten. Dagegen hat die Mutter, die gutmütig ihr Kind beobachtet, keinerlei Schwierigkeiten mit der Kontrolle des Fahrrads. Auf welche Weise hat die Mutter ihre Fähigkeit erworben? Was sind die Mechanismen des Lernens? Wie werden in der Biologie so komplexe Aufgaben wie die Koordination der Skelettmuskulatur während des Radfahrens gelöst?

Dies sind Fragen, um deren Beantwortung sich viele Forschungsgebiete bemühen. Auf der einen Seite gibt es Biologen, Mediziner und Psychologen, die Lernprozesse in biologischen Systemen untersuchen und beschreiben, auf der anderen Seite Ingenieure und Regelungstechniker, die für ihre alltägliche Arbeit Regelungsmethoden zur Lösung verschiedenartiger technischer Probleme benötigen. Zwischen diesen durchaus gegensätzlichen Wissenschaftsgebieten finden sich Forschungszweige wie die der künstlichen Intelligenz, Neuroinformatik, *Fuzzy*-Logik, genetischen Algorithmen, nichtlinearen Optimierung und der Selbstorganisation. Die vorliegende Arbeit ist im skizzierten Zwischenbereich angesiedelt und unternimmt den Versuch, aus dem Aufbau natürlicher Regelungssysteme zu lernen, um die dort erkennbaren Organisationsprinzipien in die Architektur technischer Regelungssysteme zu übertragen.

1.1. Motorische Kontrolle

Was sind nun die Prinzipien, auf Grund derer es der Natur gelingt, Regelungsprobleme, die in der Technik große Schwierigkeiten bereiten, ohne Mühe zu lösen? Um die wichtigsten Aspekte anschaulich herausarbeiten zu können, soll als Beispiel die motorische Kontrolle beim Menschen, wie sie z.B. während des Radfahrens notwendig ist, betrachtet werden.

In Abbildung 1.1 ist das motorische System schematisch dargestellt. Seine Bestandteile werden dabei als Komponenten eines technischen Regelkreises interpretiert. Den auffälligsten Teil des motorischen Systems bildet der Bewegungsapparat, d.h. Knochen, Sehnen, Bänder und Muskeln [Sch96]. Die Aufgabe der durch das *zentrale Nervensystem* (ZNS) zu leistenden Regelung besteht in der Kontrolle dieses Bewegungsapparates durch eine

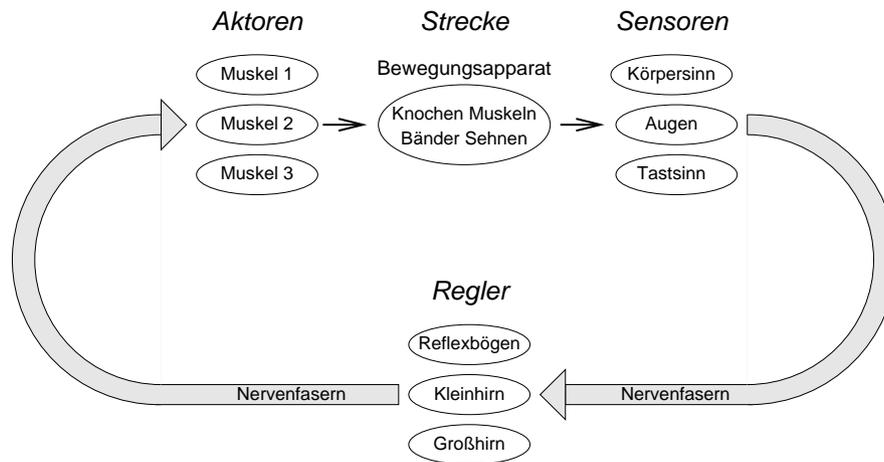


Abbildung 1.1.: **Das motorische System.** Die beteiligten Komponenten lassen sich den in der Technik üblichen Begriffen (Aktoren, Sensoren, Regler, Strecke) zuordnen. Die Informationsübertragung erfolgt über die Nervenfasern. Die Gesamtstruktur entspricht derjenigen eines Regelkreises.

Vielzahl von Muskeln. Die Muskeln sind also die *Aktoren* des Systems. Sie erhalten von den Motorneuronen des Gehirns Befehle zur Kontraktion. Informationen über die jeweils aktuelle Position der Gliedmaßen werden dem ZNS über verschiedene Sinnesmodalitäten zugeführt. Die wichtigste Rolle spielt hierbei der Körpersinn, der zum einen auf der Längenmessung der Muskelfasern durch die Muskelspindeln und zum anderen auf der Spannungsmessung der Muskeln durch die Sehnenorgane beruht [ST85]. Zusätzliche Informationen erhält das Gehirn durch die optische Wahrnehmung der Gliedmaßen und den taktilen Sinn auf der Hautoberfläche. Diese drei Sinnesmodalitäten bilden das *sensorische System* des Bewegungsapparats. Die sensorischen Signale repräsentieren die Eingabedaten für den eigentlichen *Regler* innerhalb des ZNS, der sich aus verschiedenen Reflexbögen, dem Kleinhirn und bestimmten Teilen der Großhirnrinde zusammensetzt.

Das skizzierte System zeigt einige Besonderheiten. Die große Anzahl verschiedener Muskeln und der damit verbundene hohe Freiheitsgrad an körperlichen Bewegungen läßt ahnen, welche Schwierigkeiten die Koordination so vieler Aktoren bereitet. Auch ist bemerkenswert, daß die Sensoren aus gänzlich verschiedenen Sinnen bestehen, die zudem eine Vielzahl verschiedener, zum Teil redundanter Informationen liefern. Diese Redundanzen geben uns die Möglichkeit, Bewegungen auch unter erschwerten Bedingungen auszuführen (blind zu gehen, mit dicken Handschuhen etwas zu greifen), selbst wenn uns dies ohne Übung nicht mit der gewohnten Präzision gelingt. Bei der Informationsübertragung der Sensordaten in das Gehirn ist dieses Prinzip der Redundanz ebenfalls sichtbar. Viele Stränge von Nervenfasern ziehen von den Gliedmaßen über das Rückenmark in das Gehirn. Die Information wird dabei nicht über ein besonderes Protokoll¹ übermittelt; vielmehr ist in der

¹ In der Technik würde bei einer solchen protokollbasierten Informationsübertragung die Nachricht "Thermometer mißt 435 °C und Manometer mißt 0.17 bar" über genau *eine* Leitung gesendet werden.

Natur jeder Nervenfasers ein bestimmter Informationsgehalt zugeordnet. So wird z.B. ein Reiz in einer sehr kleinen Hautregion durch eine Erregung weniger – nur diesem Hautbereich zugeordneter – Nervenzellen in das Gehirn übertragen. Liegt kein solcher Reiz vor, so fällt es uns schwer, diese Hautregion bewußt wahrzunehmen. Dieser Umstand läßt sich nun so beschreiben, daß jeder Nervenfasers eine *Bedeutung* zugeordnet wird. Erst durch die Erregung der Nervenfasers wird die zugeordnete Bedeutung wahrgenommen und so für das Gehirn *relevant*.

Abbildung 1.1 zeigt, daß auch im motorischen Kontrollsystem eine redundante Aufgabenverteilung existiert [ST85]. Sehr schnelle, unter Umständen lebensrettende Reaktionsbewegungen werden über die sog. *monosynaptischen Reflexbögen* gesteuert, bei denen sensorische Signale schon im Rückenmark an motorische Nervenzellen weitergegeben werden. Im Gegensatz zu den unwillkürlichen Reflexbewegungen stehen die willkürlichen, geplanten Bewegungen, wie z.B. das Greifen nach einer Kaffeetasse. Eine solche Aktion wird zunächst im motorischen Feld der Großhirnrinde (*motorischer Cortex*) geplant. Die Feinkoordination aller beteiligten Muskeln wird dagegen vor allem vom Kleinhirn ausgeführt [Ste88]. Das Kleinhirn ist dem Großhirn aber nicht einfach nur nachgeschaltet, sondern agiert parallel zu diesem. So können sich Menschen mit einer schweren Verletzung des Kleinhirns durchaus noch bewegen, nur zeichnen sich diese Bewegungen durch eine “falsche” Dosierung (*Asynergie*), einen *Tremor* oder durch einen zu niedrigen Muskeltonus aus [ST85]. Viele leichtere Verletzungen des Kleinhirns können aber nahezu perfekt durch das Großhirn kompensiert werden. Dagegen führt eine Läsion im Bereich der motorischen Cortex zu einer Lähmung des betroffenen Muskelbereichs, verbunden mit einem zu hohen Muskeltonus (*spastische Hemiplegie*) [ST85]. Wir werden den funktionellen Aufbau des Kleinhirns im Verlauf dieser Arbeit noch genauer diskutieren.

Diese – wenn auch nur sehr grobe – Skizze des motorischen Systems zeigt wesentliche Merkmale eines jeden biologischen Regulationssystems. Offensichtlich existiert eine gewisse Aufgabenteilung zwischen verschiedenen spezialisierten Teilen, wie z.B. den Reglern und Sensoren aus Abbildung 1.1, die von nun an als *Experten* bezeichnet werden sollen. Die Grenzen zwischen den einzelnen Experten sind aber nicht so stark abgesteckt, daß nicht ein Experte die Aufgaben eines anderen wenigstens teilweise übernehmen könnte. An Stelle einer starren Aufgabenteilung ist die Zusammenarbeit zwischen den Experten vielmehr von *Kooperation* und *Konkurrenz* geprägt. Die Kooperation wird im obigen Beispiel im Wechselspiel zwischen Grobkoordination und Feinkoordination deutlich. Eine unwillkürliche Reflexbewegung, entgegen einer andersartig geplanten Bewegung, zeigt dagegen die Konkurrenz zwischen den einzelnen Experten. Auch diese Architektur eines Systems von kooperierenden und konkurrierenden Experten wird für die weitere Argumentation von zentraler Bedeutung sein.

In dem hier beschriebenen System zur motorischen Kontrolle lassen sich drei Zeitskalen identifizieren, die für viele biologische Regulationssysteme typisch sind. Die Rad fahrende Mutter aus dem Eingangsbeispiel korrigiert zügig geringe Abweichungen von ihrem geplanten Weg und wird sich auch nicht von plötzlichen Situationsänderungen (z.B. durch

einen über den Weg springenden Hund) irritieren lassen. Ihr motorisches Kontrollsystem arbeitet vor allem auf einer *kurzen* Zeitskala. Kurzfristige Anpassungen gelingen ihrem Kind hingegen wesentlich schlechter. Es befindet sich noch in der *Lernphase*, in der sein Kontrollsystem auf einer mittelfristigen Zeitskala erst aus vielen Situationen typische Bewegungsmuster, die zum sicheren Radfahren notwendig sind, extrahiert. In dieser Phase stellt die Begegnung mit einem Hund noch ein großes Problem dar. Mutter, Kind und Hund verweisen auch auf die evolutionären Mechanismen, die das motorische Kontrollsystem erzeugt haben. Das Radfahren ist sicherlich keine Fortbewegungsform, die dem Menschen angeboren ist. Trotzdem können wir uns die dafür notwendigen – wie auch andere, viel schwierigere – Bewegungen auf Grund unserer Lernfähigkeit durch Übung aneignen. Diese Flexibilität des motorischen Systems ist die Folge einer sehr langfristigen Entwicklung auf einer Zeitskala von vielen Millionen Jahren.

Die Klärung der Fragen, wie sich die verschiedenen Regelungsmechanismen im Laufe von Jahrtausenden evolutionär herausgebildet haben und was die treibenden Kräfte für diese Evolution sind, ist Aufgabe der Phylogenie. In Anlehnung an die dort aufgedeckten Entwicklungsmechanismen der Natur existiert eine eigene Forschungsrichtung, die sich mit der Erforschung sog. evolutionärer Algorithmen zur Optimierung und Regelung technischer Systeme beschäftigt [KS97, BS93, Bäck96]. Im Folgenden werden Umformungen und Entwicklungen, die auf einer sehr langen Zeitskala ablaufen, aber nicht berücksichtigt. Vielmehr wird die Existenz sinnvoller biologischer Konstruktionen hier einfach als gegeben hingenommen. Für die Konstruktion biologisch inspirierter Algorithmen bedeutet dies, daß verschiedene Varianten eines Verfahrens nicht durch das langfristige Überleben in einer “feindlichen” Umgebung ausgewählt werden (*survival of the fittest*). Im von uns angenommenen Szenario muß vielmehr der Entwickler diesen Auswahlprozeß vornehmen, indem er Kosten definiert, die bei Über- bzw. Unterschreiten zur Verwerfung bzw. zur Weiterentwicklung einer Idee führen.

Was kann man aus der obigen Betrachtung lernen, wenn man Regelungssysteme entwickeln will, die am Vorbild des ZNS orientiert und gleichzeitig auch in einer technischen Umgebung anwendbar sein sollen? Vier Punkte kristallisieren sich heraus, die für den weiteren Verlauf dieser Arbeit von zentraler Bedeutung sind:

- Die Informationsübertragung erfolgt in einem doppelten Datenformat. Die Nervenfasern als solche hat zwar eine Bedeutung, aber erst die Aktivierung dieser Nervenfasern verleiht dieser Bedeutung aktuelle Relevanz.
- Die Gesamtaufgabe eines Regelungssystems wird in eine Reihe von Teilaufgaben zerlegt und diese werden spezialisierten Untereinheiten zugewiesen. Diese Experten lösen die gestellte Gesamtaufgabe in einem Wechselspiel aus Kooperation und Konkurrenz.
- Regelungssysteme zeigen einen modularen Aufbau. Das motorische Kontrollsystem gliedert sich in Sensoren, Regler und Aktoren. Das Gehirn als Regler ist in viele,

teils hierarchische, teils parallel arbeitende und voneinander abhängige Funktionsgruppen strukturiert. Dieses Strukturprinzip ist bis zur untersten Ebene hinab, den Nervenzellen, sichtbar.

- Bei der Konstruktion eines technischen Regelungssystems muß der Gesamtaufbau von einem Entwickler unter Berücksichtigung von Kosten vorgegeben werden, da die langfristige selbstorganisierende Auswahl einer konkreten Systemarchitektur durch evolutionäre Algorithmen zum einen zu lange dauern, zum anderen hohe technische Risiken verursachen würde. In dem von uns verfolgten Szenario soll ein solches System mittel- und kurzfristige Anpassungen selbständig vornehmen können.

Unter Berücksichtigung dieser vier Vorgaben wird in diesem Text ein neuronal motiviertes Architekturprinzip für komplexe Regelungssysteme entwickelt, theoretisch erforscht und an Hand einer industriellen Anlage auf seine praktische Realisierbarkeit untersucht. Bei dieser Anlage handelt es sich um jenen Teil des Werks Burghausen der Wacker-Chemie GmbH, in welchem die Rückstände der chemischen Produktion verbrannt werden. Die Kontrollprobleme dieser Teilanlage waren es, welche diese Arbeit motiviert haben. Um in die entsprechenden Fragestellungen einzuführen, gibt der folgende Abschnitt zunächst einen Überblick über den allgemeinen Aufbau einer industriellen Produktionskontrolle, um anschließend die Kontrollprobleme bei der Rückstandsverbrennung im Werk Burghausen zu skizzieren.

1.2. Industrielle Regelung

Abbildung 1.2 zeigt ein Luftbild des Werks Burghausen der Wacker-Chemie GmbH. Auf diesem etwa 100 ha großen Gelände werden in etwa 30 Produktionsbetrieben vor allem Grundstoffe für die chemische Industrie hergestellt [WCG99]. Die kontrollierte Führung einer technischen Anlage dieser Größe erfordert besondere Strategien. Wie in Abbildung 1.3 dargestellt ist, ist eine ausgeprägte Hierarchie der Organisationsstruktur notwendig, die sich auch in der informationstechnischen Ausstattung widerspiegelt. Die Unternehmensführung trifft vor allem langfristige betriebswirtschaftliche Entscheidungen, die sich nur schlecht durch Informationssysteme automatisieren lassen. Auf der Ebene der Produktionsführung werden mittelfristige Planungen für die Produktionsabläufe vorgenommen, die durch die Führungen der Teilbetriebe umgesetzt werden sollen. Diese beiden mittleren Ebenen werden in letzter Zeit zunehmend durch Informationssysteme unterstützt. Die auf der untersten Ebene angesiedelte Prozeßführung ist lediglich für den korrekten Ablauf des jeweiligen technischen Verfahrens verantwortlich. Schon lange werden hier Prozeßleitsysteme für die Automatisierung und Regelung zu Hilfe genommen.

In den letzten Jahren haben im Bereich der industriellen Leittechnik starke Veränderungen stattgefunden. Waren frühere Leitsysteme proprietäre Lösungen des jeweiligen Herstellers,



Abbildung 1.2.: **Werk Burghausen der Wacker-Chemie GmbH.** Das Gelände wird vom Alz-Salzach-Kanal unterteilt, dessen Bau um 1900 für den Standort des Werks ausschlaggebend war. Der Kreis markiert die Anlage für die Rückstandsverbrennung (aus [[WCG99](#)]).

so ist derzeit ein deutlicher Trend hin zur Standardisierung der verschiedenen Komponenten zu beobachten [[Bir99](#)]. Diese Entwicklung ist aber noch im Gange und wird stark von den Fortschritten der Netzwerktechnik beeinflusst². Die zunehmend einfachere Verfügbarkeit von Informationen zu Produktions- und Prozeßabläufen im gesamten Werk wird gegenwärtig vor allem zur Analyse und Archivierung von Betriebsdaten verwendet. Zur eigentlichen Prozeßführung, also zur Kontrolle des technischen Ablaufes, werden diese neuen informationstechnischen Möglichkeiten dagegen noch nicht systematisch genutzt. Dafür verantwortlich ist ein gewisser Mangel an Methoden, mit denen die vielen verschiedenen Informationen zur Kontrolle eines Prozesses verknüpft werden können. Etabliert sind dagegen Verfahren, die durch Messung einer einzelnen Größe (z.B. Druck) und durch Variation einer zugehörigen Stellgröße (z.B. Ventilstellung) einen Sollwert einregeln.

Ende 1994 kam im Rahmen eines Vortrags von Paul Tavan über Methoden der Neuroinformatik ein Kontakt mit dem Regelungsingenieur Thomas List zustande. T. List war damals im Werk Burghausen der Wacker-Chemie GmbH für die Optimierung bestehender Anlagenteile zuständig und daher auch immer auf der Suche nach neuen, erfolgverspre-

²Es ist zu erwarten, daß das für das *internet* zentrale Übertragungsprotokoll TCP/IP und die plattformunabhängige Programmiersprache *java* auch in der Leittechnik immer wichtiger werden.

Funktion	System
Unternehmensführung	
Produktionsführung	Produktionsleitsystem
Betriebsführung	Betriebsleitsystem
Prozeßführung	Prozeßleitsystem

Abbildung 1.3.: **Hierarchieebenen eines industriellen Betriebs.** Links sind die verschiedenen Organisationsebenen dargestellt und daneben die informationstechnischen Systeme, die zu Unterstützung eingesetzt werden (nach [Bir99]).

chenden Kontrollverfahren.

Eine seiner damaligen Aufgaben war die Behebung bestehender Probleme in der Rückstandsverbrennungsanlage des Werks. Dieser Betriebsteil dient der Verbrennung von Reststoffen, die innerhalb der verschiedenen Produktionsanlagen des Werks anfallen. Die Anlage zur Rückstandsverbrennung umfaßt drei verschieden große Öfen, in denen unter Einhaltung diverser Grenzwerte die Abfälle (Gase, Flüssigkeiten und Feststoffe) bei hoher Temperatur verbrannt werden. Wegen der gestiegenen Produktion des Werks und des damit verbundenen Anstiegs an Abfallstoffen drohte die Kapazität der Anlage zur Rückstandsverbrennung an ihre Grenzen zu stoßen. Es war daher das Ziel von Herrn T. List, Möglichkeiten einer Kapazitätserhöhung zu untersuchen und gegebenenfalls entsprechende Maßnahmen zu ergreifen. Der Einsatz konventioneller Regelungstechnik zur Lösung dieser Aufgabe war jedoch wegen der relativ komplexen Struktur der Rückstandsverbrennungsanlage mit großen Schwierigkeiten verbunden.

Diese Probleme führten zu der Idee, in einem staatlich geförderten Forschungsprojekt die Rückstandsverbrennung als Demonstrationsanlage für die Anwendbarkeit von Regelungsmethoden der Neuroinformatik zu nutzen [DLT95]. Insbesondere sollte in diesem Projekt ein ganzheitlicher Entwurf eines Systems zur Produktions-, Betriebs- und Prozeßführung versucht werden. Sparmaßnahmen der staatlichen Förderstellen ließen ein Projekt dieser Größenordnung aber leider nicht zu. Trotzdem wurde im März 1995 eine – von der Wacker-Chemie GmbH finanzierte – “kleine” Kooperation vereinbart, in deren Rahmen Teile der vorliegenden Arbeit durchgeführt wurden. Der Ablauf dieses Projekts soll nun kurz skizziert werden.

Nach einer Analyse der Anlage und vorhandener archivierter Daten wurden verschiedene Problemfelder und Optimierungsmöglichkeiten deutlich. Eine zentrale Aufgabe der Anlage ist die zuverlässige und ständige Verbrennung der über lange Rohrleitungen angelieferten Abgase aus den Produktionsbetrieben. Da diese Abgase kaum gepuffert werden, ist eine ständige Verfügbarkeit der Rückstandsverbrennung notwendig. Ein längerer Ausfall der Anlage würde die Einstellung von Produktionsprozessen erzwingen. In Hinblick auf das

Kontrollproblem kommt erschwerend hinzu, daß viele Abgase nicht kontinuierlich, sondern – abhängig vom jeweiligen Produktionsverfahren – stoßartig herangeführt werden. Derartige sehr plötzlich auftretende Belastungsspitzen können zu Über- bzw. Unterschreitungen von Grenzwerten, somit zu einer erzwungenen Notabschaltung von Verbrennungsöfen und in der Folge auch zum Stillstand in bestimmten Produktionsbetrieben führen.

Ein weiteres Problemfeld betrifft die Verbrennung von flüssigen Abfällen. Deren chemische und physikalische Eigenschaften sind in aller Regel nicht bekannt und somit ist kaum Wissen über Verbrennungsenthalpien und Viskositäten verfügbar. Dies macht ihre kontrollierte Dosierung zu einem schwierigen Unterfangen. Da sowohl die Ventilcharakteristiken wie auch die Durchflußmessungen auf Grund der unbekanntenen Viskositäten mit sehr großen Fehlern behaftet sind, muß häufig allein aus der Reaktion des Verbrennungsvorgangs auf die Verbrennungsenthalpien der jeweiligen Flüssigkeiten geschlossen werden.

Da die Anlage zur Rückstandsverbrennung seit etwa 1950 ständig erweitert wurde, finden sich in ihr sehr viele historisch bedingte Eigenarten und Konventionen³, die zu einer gewissen Unübersichtlichkeit führen. Das in der Anlage installierte Prozeßleitsystem stammt aus den achtziger Jahren und hat sich für eine Optimierung der Regelung als großes Hindernis erwiesen, da seine sehr schwerfällige Programmierung die Implementierung neuer Methoden fast unmöglich macht. Erst die Programmierung einer Schnittstelle zu einer modernen *workstation* hat die für die Entwicklung neuartiger Regelungsverfahren notwendige Flexibilität erzeugt.

Der wichtigste Regelkreis, der im Prozeßleitsystem implementiert ist, versucht den Druck in den Abgasleitungen innerhalb gewisser Grenzen auf einen konstanten Wert zu halten. Diese Regelung ist aber nicht die Hauptaufgabe des Prozeßleitsystems. Sie besteht vielmehr darin, die Visualisierung zahlreicher Meßwerte und die Fernsteuerung ebenso zahlreicher Ventile zu ermöglichen. Mit der Beobachtung und Bedienung des Leitsystems sind die Anlagenfahrer der jeweiligen Schicht für die kontrollierte Verbrennung der Gase und Flüssigkeiten verantwortlich. Ihre wesentliche Strategie besteht darin, die Öfen in einem sehr "sicheren" Bereich zu belassen, d.h. den Betriebszustand der Öfen möglichst weit entfernt von Grenzwerten, deren Überschreitung zur automatischen Notabschaltung führen würde, zu halten. Diese Strategie führt zwar zu einer wünschenswerten Reduktion der Ausfälle; andererseits wird aber dadurch auch die mittlere Verbrennungskapazität der Anlage verringert.

1.3. Ein neuronaler Ansatz zur industriellen Kontrolle

Die Zusammenarbeit mit der Wacker-Chemie GmbH von März 1995 bis Dezember 1997 bot die seltene Gelegenheit "echte" industrielle Daten zu analysieren und auch regelnd in

³So werden die 12 Abgasleitungen von 1 bis 13 durchnummeriert. Leitung sieben existiert nicht.

einen realen, komplexen Prozeß einzugreifen. In vielen Produktionsanlagen werden verfahrenstechnische Abläufe zum Schutz vor Industriespionage vertraulich behandelt. Daher ist in solchen Anlagen ein detaillierter Einblick werksfremder Personen sehr unüblich.

Dies mag auch ein Grund dafür sein, weshalb zwar viele Konzepte zur intelligenten Kontrolle mit Methoden der Neuroinformatik publiziert werden [MSW92, DeS95], aber bislang nur wenige dieser Konzepte an tatsächlich existierenden Anlagen erprobt wurden. Das in der Literatur häufig skizzierte Vorgehen besteht vielmehr darin, neue Methoden an Hand der Simulation von Modellsystemen zu evaluieren. Dabei werden aber Aspekte wie Verfügbarkeit, Zuverlässigkeit, Pflegbarkeit und praktische Realisierbarkeit häufig vernachlässigt. Die wenigen zugänglichen Beispiele für industrielle Anwendungen von Regelungsmethoden der Neuroinformatik können in zwei Gruppen eingeteilt werden.

Zum einen finden sich in der Literatur seit etwa 1987 Arbeiten zum Einsatz (z.B. [SR87, LBD⁺90]) von sogenannten *multi layer perceptrons* (MLP), die mit dem *backpropagation*-Algorithmus trainiert wurden [RHW86, And97]. Das Training dieses Netzwerktyps erfolgte dabei häufig mit einem der inzwischen allgemein verfügbaren Programmpaketen [Uni99, Mat99]. Die mathematisch komplizierte und biologisch wenig plausible MLP-Struktur erlaubt es aber nicht, aus diesen häufig sehr speziellen Arbeiten Rückschlüsse auf den generellen Nutzen dieser Netzwerkarchitektur zu ziehen [Cri89]. Andererseits werden seit etwa 15 Jahren zunehmend neuronale Netzwerkmodelle verwendet [Mal81, RMS92], die auf Konzepten der biologischen Selbstorganisation beruhen. Insbesondere wurden hier sogenannte *neuronale Karten* [Koh82, RT89, Koh90, MBS93], Netzwerke mit *radialen Basisfunktionen* (RBF) [MD89, PG90a, ABK⁺99] oder mit *lokalen Experten* [JJ93a, JJ92] intensiv untersucht. Solche selbstorganisierenden Modelle wurden auch zur Lösung ingenieurtechnischer Fragen eingesetzt, wie z.B. zur Kontrolle einer Walzstraße [MGPS96], zur Kontrolle von Robotern [WMS91] und von autonomen Fahrzeugen [MSW92]. Die vorliegende Arbeit ordnet sich in diesen Diskussionszusammenhang ein.

Angesichts der relativ geringen Anzahl tatsächlicher Anwendungen von Kontrollmethoden der Neuroinformatik in der industriellen Praxis und des Mangels eigener Erfahrungen auf diesem Gebiet stellte die oben skizzierte Kooperation mit der Wacker-Chemie GmbH ein Wagnis dar. Andererseits repräsentierten die damit verbundenen Aufgabestellungen auch eine Herausforderung, da sie eine neue Form des interdisziplinären Arbeitens erforderten. Auf der einen Seite waren die Einarbeitung in Verfahrens- und Leitsystemtechnik sowie der Austausch mit Anlagenfahrern und Technikern vor Ort notwendig. Auf der anderen Seite mußten konzeptionelle Entwicklungen geleistet, Tests an Modellbeispielen durchgeführt und schließlich neue Regelungsmethoden in ein zuverlässiges *software*-System implementiert werden.

In Anlehnung an die weiter oben skizzierten Prinzipien der motorischen Kontrolle bei Lebewesen, die Architektur von RBF-Netzwerken sowie an Konzepte von M. Jordan und R. Jacobs [JJ92, JJ93a] wurde für die Regelung der Anlage zur Rückstandsverbrennung ein hierarchisches System lokaler Experten entwickelt. Jeder Experte ist dabei für eine bestimmte Teilaufgabe zuständig. So gibt es z.B. einen Experten, der für die Einhaltung des

vorgegebenen Drucks in den Abgaszuleitungen verantwortlich ist, und andere Experten, die nur bei Gefahr einer Notabschaltung durch Unter- bzw. Überschreitung von Grenzwerten in den Prozeß eingreifen. Im Gesamtsystem arbeiten etwa 20 Experten bei der Kontrolle der Rückstandsverbrennung zusammen, deren Zusammenarbeit als Wechselspiel von kompetitiven und kooperativen Prozessen organisiert ist.

Es ist ein Ziel dieses Textes, dieses an biologischen Vorbildern angelehnte Expertensystem zu begründen. Entsprechend ist sie in einen theoretischen Teil und einen Abschnitt, der noch genauer auf das geschilderte Industrieprojekt eingehen wird, gegliedert. Weitere Gliederungsaspekte ergeben sich aus der Notwendigkeit, in Hinblick auf die skizzierte Kontrollaufgabe sowohl Methoden der Neuroinformatik wie auch der Regelungstechnik zu untersuchen.

1.4. Aufbau der Arbeit

Die vorliegende Arbeit ist in drei Blöcke aufgeteilt. Sie umfaßt einführende Kapitel, theoretische Entwicklungen und die Demonstration der neuen Methoden an Hand von Beispielen:

- In Kapitel 2 wird zunächst ein Überblick über den Aufbau des Gehirns und die Struktur biologischer Neuronennetze gegeben. Daraus werden eine Reihe von Forderungen an die mathematischen Eigenschaften entsprechender Modelle abgeleitet, d.h. es werden die Eigenschaften *künstlicher neuronaler Netzwerke* (KNN) spezifiziert. Dabei werden die KNN-Modelle als Varianten klassischer mathematischer Verfahren zur Mustererkennung, Funktionenapproximation und Dichteschätzung interpretiert. Nach der Beschreibung einiger, für die Entwicklung der Neuroinformatik wichtiger, Modelle neuronaler Netzwerke wird der Schwerpunkt der weiteren Betrachtung auf das *radiale Basisfunktionen Netzwerk* (RBF) und das *generalisierte radiale Basisfunktionen Netzwerk* (GRBF) gelenkt, die spezielle Varianten der *feed forward* Netzwerke sind.

Abbildung 1.4 zeigt den Aufbau eines solchen dreischichtigen neuronalen Netzwerks. Dem Netzwerk werden über die *Eingabeschicht* Daten zugeführt, die durch die *Zwischen-* und *Ausgabeschicht* verschiedenen Transformationen oder Umkodierungen unterworfen werden. Die drei Schichten sind jeweils aus künstlichen Neuronen aufgebaut, deren Reizantwortcharakteristik der Funktionsweise biologischer Nervenzellen nachgebildet ist. Die Neuronen einer Schicht geben über Verbindungen Informationen an die Neuronen der nachfolgenden Schicht weiter. Die Gewichte dieser Verbindungen stellen freie Parameter des Netzwerks dar. Weitere freie Parameter gehen in die Reizantwortcharakteristiken der Neuronen ein. Die Anpassung dieser Parameter durch einen geeigneten Algorithmus wird als *Lernen* oder *Training* bezeichnet. Dazu wird ein *Trainingsdatensatz* benötigt, in dem zu jedem Satz von Eingabedaten die zu realisierenden Ausgaben vorgegeben sind.

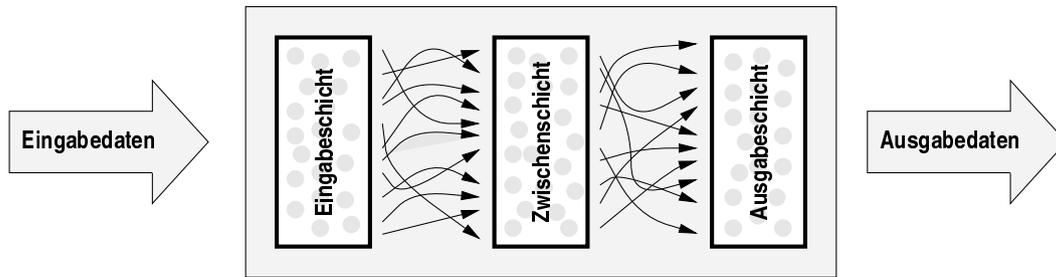


Abbildung 1.4.: **Dreischichtiges neuronales Netzwerk.** Die drei Schichten werden jeweils durch künstliche Neuronen gebildet (graue Kreise). Diese können über parametrisierbare Verbindungen (dünne Pfeile) Informationen an die Neuronen der nachfolgenden Schicht weitergeben.

- In Kapitel 3 wird erläutert, warum für das optimale *Training* eines GRBF-Netzwerks eine sogenannte parametrische Dichteschätzung der Verteilung der Eingabedaten erforderlich ist. Für den Fall der Dichteschätzung mit einer *Mischung multivariater Normalverteilungen* werden Lernregeln für die freien Parameter hergeleitet, welche die *likelihood* der Dichteschätzung bei schneller und sicherer Konvergenz maximieren [ABK⁺99]. Anschließend wird dieses Konzept zum Zweck der Funktionenapproximation durch ein GRBF-Netzwerk verallgemeinert, indem die gemeinsame Dichte der Ein- und Ausgabedatenverteilungen einer Schätzung unterzogen wird. Man erhält so eine neuartige Funktionenapproximation durch *lokale lineare Modelle* (LLM). Die Arbeitsweise dieses LLM-Approximators wird an einigen einfachen Beispielen demonstriert.

Allen Typen von GRBF-Netzwerken ist eigen, daß die Kodierung der Eingabedaten auf der Zwischenschicht des Netzes stark lokalisiert ist. Entsprechend können die Neuronen der Zwischenschicht als lokale Experten für einen Teil der Eingabedaten aufgefaßt werden. Diese Eigenschaft wird in Kapitel 4 dazu genutzt, ein allgemeines Konzept für ein System zu entwerfen, welches aus lokalen Experten aufgebaut ist, die über klar definierte, aber möglicherweise wechselseitig überlappende Zuständigkeiten verfügen. Die Experten berechnen neben ihrer Ausgabe noch eine *Kompetenz*, mit der sie die Relevanz ihrer Ausgabe quantifizieren. Diese zusätzliche Angabe ihrer eigenen Kompetenz ermöglicht, wie gezeigt werden wird, die Einführung von kompetitiven und kooperativen Wechselwirkungen zwischen den Experten.

- Als eine anspruchsvolle Anwendung von neuronalen Netzen, welche die Lösung von Aufgaben wie die der Klassifikation, der Funktionenapproximation und des Lernens einschließt, gilt die Regelung von technischen Systemen. Kapitel 5 führt die in diesem Zusammenhang wichtigsten Begriffe ein und demonstriert an Hand dreier prinzipiell verschiedener Möglichkeiten, wie eine solche Regelung realisiert werden kann.

Mit dem analytischen Modell des sogenannten Bioreaktors ist von verschiedenen

Autoren ein *benchmark*-System für den Vergleich von Regelungssystemen eingeführt worden. In Kapitel 6 werden zunächst die Differentialgleichungen diskutiert, welche diesen biochemischen Prozeß beschreiben, und es werden damit die Schwierigkeiten der Regelung dieses Systems aufgezeigt. Anschließend werden ein klassischer PID-Regler, eine *Fuzzy*-Regelung und ein Regler, der auf einem LLM-Approximator beruht, am Beispiel dieses Bioreaktors getestet und verglichen.

Kapitel 7 beschreibt schließlich am Beispiel der schon angesprochenen Rückstandsverbrennung jene Erfordernisse und Probleme, die bei der Umsetzung neuer Regelungskonzepte in die Praxis des industriellen Alltags auftreten. Viele der in Kapitel 3 und 4 beschriebenen Verfahren wurden verwendet, um die Regelung der Rückstandsverbrennung erfolgreich zu verbessern. Nachdem Aufgabe und Aufbau dieser Anlage dargestellt und die *software*-technischen Probleme der Anwendung geschildert sind, wird auf verschiedene Aspekte des realisierten Gesamtsystems eingegangen.

2. Gehirn, Datenanalyse und neuronale Netzwerke

1987 wurden von M. Powell [Pow87] Netzwerke aus radialen Basisfunktionen (RBF) als Werkzeuge zur Interpolation hochdimensionaler Datensätze ausführlich untersucht und so der wissenschaftlichen Öffentlichkeit bekannt gemacht. In der Folge hat sich aus dieser zunächst rein mathematisch begründeten Methode das *RBF-Netzwerk* als ein weithin beachteter Netzwerktyp der Neuroinformatik entwickelt und etabliert. Eine große Zahl von Veröffentlichungen hat sich mit verschiedenen Varianten, Lernalgorithmen und Anwendungen von RBF-Netzwerken beschäftigt (siehe z.B. [CCG91, MAC⁺95, BFG95, BF95, KL98]), so daß diese Netzwerke heute Alternativen zum ursprünglich sehr weit verbreiteten, aber biologisch nur schwer interpretierbaren *multi layer*-Perzeptron [RHW86, And97] darstellen. Wegen der zentralen Stellung, die das RBF-Netzwerkmodell in der vorliegenden Arbeit hat, sollen vorab die Fragen diskutiert werden, (i) warum RBF-Netzwerke in der Neuroinformatik eine derart wichtige Rolle spielen, (ii) in wie weit sie neurobiologische Beobachtungen widerspiegeln und, (iii) in welchem Zusammenhang ihre mathematischen Eigenschaften mit den Fähigkeiten des Gehirns stehen. Für ein tiefergehendes Verständnis ist es dabei notwendig, etwas weiter auszuholen und vorab einige Fakten, Methoden und Konzepte aus den Forschungsgebieten der Neuroanatomie, statistischen Datenanalyse und Neuroinformatik zu referieren. Darauf aufbauend kann anschließend der Prototyp eines RBF-Netzwerks [MD88, MD89] vorgestellt, begründet und untersucht werden.

2.1. Überblick

Ziel der Neuroinformatik ist es, die Funktionsweise des Gehirns durch die Konstruktion vereinfachter mathematischer Modelle zu verstehen. Dabei wird das zentrale Nervensystem (ZNS) als informationsverarbeitendes System aufgefaßt. Ausgehend vom heutigen Kenntnisstand wird angenommen, daß seine informationsverarbeitenden Fähigkeiten – wie in der Einleitung am Beispiel der motorischen Kontrolle skizziert – auf der Arbeit der Nervenzellen und der Informationsübertragung durch die Nervenfasern beruhen.

Um empirische Grundlagen für die Bildung von Modellen in der Neuroinformatik, insbesondere der RBF-Netzwerke, zu erarbeiten, werden in Abschnitt 2.2 zunächst einige jener

anatomischen und physiologischen Aspekte des ZNS dargestellt, die für die weiteren Ausführungen von Bedeutung sind. Dabei wird genauer auf das Kleinhirn eingegangen, welches für die Feinkoordination im motorischen System verantwortlich ist. Dieser Teil des ZNS zeigt eine relativ einfache anatomische Struktur, deren Verschaltung von J.C. Eccles und Mitarbeitern [EIS67, Ecc73] entschlüsselt wurde. 1969 wurde von D. Marr [Mar69] ein erstes Funktionsmodell für die Kleinhirnrinde entwickelt und anschließend von J. Albus [Alb71] verfeinert. Die Anatomie und Funktionsweise der Kleinhirnrinde soll hauptsächlich deshalb relativ detailliert besprochen werden, da sie das biologische Vorbild für RBF-Netzwerke ist [PG89].

Da im Laufe dieser Arbeit verschiedene mathematische Aspekte des von uns entwickelten generalisierten RBF-Netzwerks (GRBF) untersucht werden sollen (vgl. [Ado92, KT97, ABK⁺99]), wird mit Abschnitt 2.3 ein mathematischer Teil eingeschoben, in dem Begriffe eingeführt werden, die für das weitere Verständnis wichtig sind. GRBF-Netzwerke dienen – wie viele andere neuronale Netzwerkmodelle auch – der Approximation eines gegebenen Datensatzes durch eine Modellfunktion. Entsprechend soll diese Aufgabe zunächst allgemein mathematisch skizziert werden. Ferner soll gezeigt werden, daß solche Funktionenapproximatoren in praktischen Anwendungen dazu genutzt werden können Klassifikationsprobleme zu lösen, wie sie z.B. bei der Spracherkennung auftreten. Zur Vorbereitung der Analyse in Kapitel 3, die tiefgreifende Zusammenhänge [Klo96] zwischen unserem GRBF-Netzwerk [ABK⁺99] und etablierten statistischen Verfahren der parametrischen Dichteschätzung [DH73] ausnutzen wird, sollen die oben genannten Konzepte eingeführt werden. Die Auswahl der in Abschnitt 2.3 dargestellten mathematischen Verfahren ist also vor allem dadurch motiviert, daß viele dieser Algorithmen bei der Diskussion von RBF-Netzwerken von grundlegender Bedeutung sind.

Nach der Einführung anatomischer und mathematischer Begriffe steht ein Teil jener Mittel bereit, die zur neurobiologischen Begründung der Formulierung unseres GRBF-Netzwerks benötigt werden. Es fehlen aber noch einige Vorstellungen, die im Bereich der Neuroinformatik mittlerweile wohletabliert sind.

Die Grundlagen dieses Wissenschaftsgebiets wurden Ende des letzten Jahrhunderts gelegt. Damals waren die Mikroskopie- und Präparationstechniken ausreichend weit entwickelt, um die anatomische Struktur des Gehirns untersuchen [yC52, Gol23] und so auch seinen funktionellen Aufbau erforschen zu können. Seitdem ist die Vorstellung vom Gehirn als einem Netzwerk von Nervenzellen, welche über Nervenfasern elektrische Signale austauschen, allgemein akzeptiert. Eines der zentralen Paradigmen der Neuroinformatik ist daher die Überzeugung, daß sich die kognitiven Fähigkeiten des Gehirns auf ein dynamisches, sich selbst organisierendes Netzwerk von neuronalen “Prozessoren” zurückführen läßt. Auf der Basis dieser auch als *Konnektionismus* bezeichneten Hypothese stellten 1943 W. McCulloch und W. Pitts [MP43] erstmals ein Netzwerk *formaler* Neuronen vor, auf das, wie sie zeigen konnten, jede beliebige Operation der binären Logik abgebildet werden kann. In den letzten Jahrzehnten hat sich daraus das Forschungsgebiet der Neuroinformatik entwickelt, welches inzwischen in viele verschiedene Teilgebiete gegliedert

ist und dementsprechend viele Konzepte und Theorien hervorgebracht hat. Im dritten und letzten Abschnitt 2.4 dieses Kapitels werden daher diejenigen Modelle der Neuroinformatik beschrieben, die für das Verständnis unseres GRBF-Netzwerks notwendig sind. Den Abschluß bildet die Darstellung jenes modifizierten RBF-Netzwerks, welches 1988 von J. Moody und C. Darken [MD88, MD89] vorgeschlagen wurde, und welches von uns zum GRBF-Netzwerk verallgemeinert wurde [ABK⁺99].

2.2. Gehirn und Kleinhirnrinde

Das Gehirn des Menschen und anderer höherer Lebewesen ist ein ungemein kompliziertes Gebilde, das sich in eine große Anzahl anatomischer Strukturen gliedern läßt [Sch96]. Abbildung 2.1 zeigt einen Blick auf das menschliche Gehirn zusammen mit den Bezeichnungen der auffälligsten Strukturen. Aus naturwissenschaftlicher Sicht fungiert dieses Gebilde als informationsverarbeitendes System, das seine Eingabeinformationen aus der Aktivität der verschiedenen Sinne bezieht. Die Sinnesreize rufen im Gehirn höchst komplexe und im Einzelnen größtenteils noch unbekannte Abläufe hervor, die letztendlich zu Aktionen der Muskeln führen.

In diesem Jahrhundert und speziell im vergangenen Jahrzehnt wurden außerordentlich große Anstrengungen unternommen, um dieses zunächst geheimnisvoll erscheinende Organ und die Prinzipien seiner Arbeitsweise systematisch zu erforschen. Als Ergebnis dieser Untersuchungen zur anatomischen, physiologischen und funktionellen Struktur des Gehirns hat sich inzwischen ein sehr breites und zum Teil auch tiefgehendes Wissen angesammelt, welches insbesondere auch in die Modellbildungen der Neuroinformatik eingegangen ist. Dabei werden die zu einem Nervennetzwerk verbundenen Nervenzellen als die wesentlichen Elementarbausteine der Hirnfunktion aufgefaßt. Entsprechend sollen im nächsten Abschnitt zunächst die für die Bildung von Funktionsmodellen wichtigsten Eigenschaften dieser Bausteine skizziert werden.

2.2.1. Nervenzellen und Synapsen

Um die Jahrhundertwende konnten Ramón y Cajal und C. Golgi auf Grund spezieller Präparationstechniken erstmals einzelne Nervenzellen innerhalb sehr dünner Gehirnschnitte unter dem Mikroskop nachweisen [yC52, Gol23]. Für diese Leistung wurden sie 1906 gemeinsam mit dem Nobelpreis ausgezeichnet. Schnell wurde deutlich, daß das Gehirn aus einer Unzahl von Nervenzellen aufgebaut ist, die, verbunden durch die Nervenfasern, ein sehr kompliziertes Geflecht bilden. Heute wird geschätzt, daß sich unser Gehirn aus etwa $10^{10} - 10^{11}$ Nervenzellen zusammensetzt. Dabei ist jede dieser Zellen durch Nervenfasern mit etwa 1000 anderen Zellen verbunden [ST85]. Diese grundlegenden¹ Bausteine des

¹Die Nervenzellen werden meist zusätzlich von Begleitzellen umhüllt. Diese Gliazellen, die zahlreicher als die Neuronen sind, nehmen mehr als die Hälfte des Raums im Gehirn ein und scheinen vor allem eine

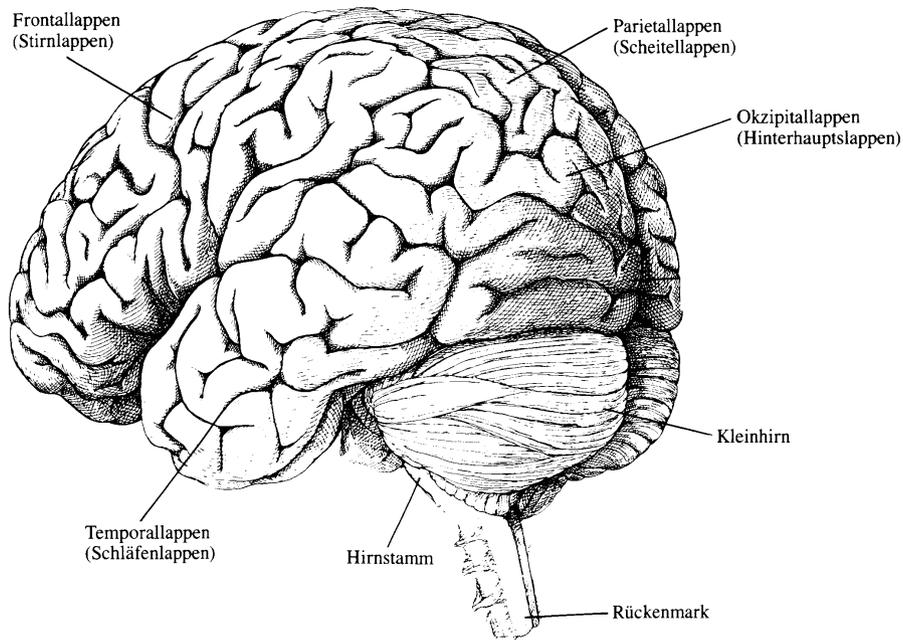


Abbildung 2.1.: **Das menschliche Gehirn.** Die stark gefaltete Großhirnrinde überdeckt die stammesgeschichtlich älteren Strukturen des Kleinhirnes, des Hirnstammes und des hier nicht sichtbaren Mittelhirns (aus [Hub90]).

Gehirns treten, abhängig von ihrem anatomischen Umfeld, in vielen Variationen auf. Im folgenden sollen aber zunächst einige gemeinsame anatomische Merkmale von Nervenzellen besprochen werden [ST85].

Wie in Abbildung 2.2 schematisch dargestellt ist, besteht eine Nervenzelle aus drei Funktionsgruppen: dem Zellsoma, den Dendriten und dem Axon. Die Zellmembran, die als guter elektrischer Isolator wirkt, umschließt diese Funktionsgruppen und grenzt so die Zelle gegen ihre Umgebung ab. Die Dendriten sind häufig mehrfach verzweigt und erstrecken sich in die nähere Umgebung des Zellsomas. Das Axon, welches oft auch als Nervenfasern bezeichnet wird, entspringt am Axonhügel aus dem Soma und ist typischerweise relativ lang, sehr dünn und am Ende verzweigt. Axone werden von präsynaptischen Endigungen abgeschlossen.

Bei einem Kontakt eines Axons mit der Zellmembran einer anderen Nervenzelle kann sich eine *Synapse* ausbilden, welche die Übertragung von elektrischen Signalen ermöglicht. Diese Kontaktstellen bestehen aus der präsynaptischen Endigung des Axons, dem synaptischen Spalt und der postsynaptischen Membran, welche zumeist an den Dendriten oder am Zellkörper, manchmal aber auch am Axon der nachgeschalteten Zelle lokalisiert ist. Man kann daher zwischen axo-dendritischen, axo-somatischen und axo-axonischen Synapsen unterscheiden.

Versorgungs- und Schutzfunktion für die Nervenzellen zu haben [ST85].

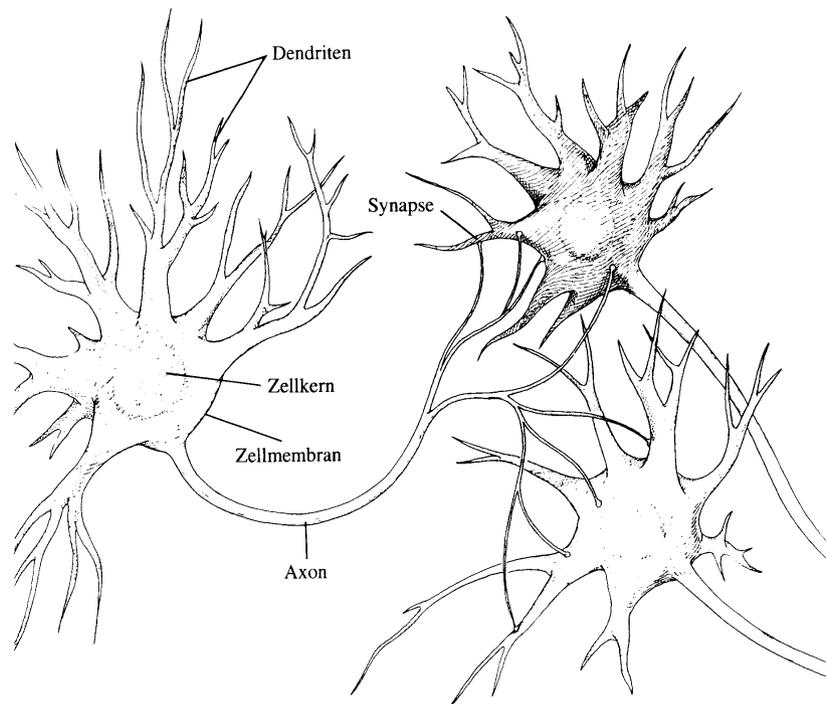


Abbildung 2.2.: Vereinfachter Aufbau von Nervenzellen (aus [Hub90]). Im Gehirn hat jede Zelle durchschnittlich mit etwa 1000 anderen Zellen synaptische Verbindungen. Die typische Größe einer Nervenzelle liegt bei 5-100 μm . Das Axon kann bis zu einem Meter lang werden.

Vom Standpunkt der Informationsverarbeitung betrachtet, fungieren die Nervenzellen im Organismus zugleich als Prozessoren wie auch als Leiterbahnen. Entsprechend lassen sich ihre speziellen Aufgaben einteilen in die Aufnahme von Information, ihre Verarbeitung, die Fortleitung der verarbeiteten Informationen über größere Distanzen, sowie deren Weitergabe an eine oder mehrere nachgeschaltete Zellen. Betrachtet man den Organismus als Ganzes, so erfolgt die Aufnahme von Information durch die Sinneszellen, die als Sensoren auf die Detektion verschiedener physikalischer Umgebungseinflüsse spezialisiert sind. Diese Zellen kodieren die Ergebnisse ihrer jeweiligen Messungen zunächst durch graduelle Veränderungen ihres Membranpotentials [Ste88]. Diese Veränderungen können von nachgeschalteten Nervenzellen abgegriffen werden und dort in einer Art Analog-Digital-Umwandlung Sequenzen pulsformiger Depolarisationen des Membranpotentials auslösen. Diese *spikes* werden schließlich über die Axone dieser Zellen weitergeleitet.

Die biophysikalischen Mechanismen des axonalen *spike*-Transports sind im Prinzip seit den Pionierarbeiten von A. Hodgkin und A. Huxley (Nobelpreis zusammen mit J.C. Eccles 1963, [HH52]) bekannt. Ihre biochemischen und molekularen Grundlagen wurden seitdem im Detail aufgeklärt. Die Ausbreitungsgeschwindigkeit der etwa 1 ms andauernden *spikes* liegt zwischen 1 und 100 m/s. Insgesamt scheint bei der Umwandlung der an den Sinneszellen zumeist noch in analoger Form repräsentierten Meßwerte in Pulsfolgen der nachgeschalteten Zellen die sog. Frequenzkodierung wirksam zu sein: Starke Sinnesreize

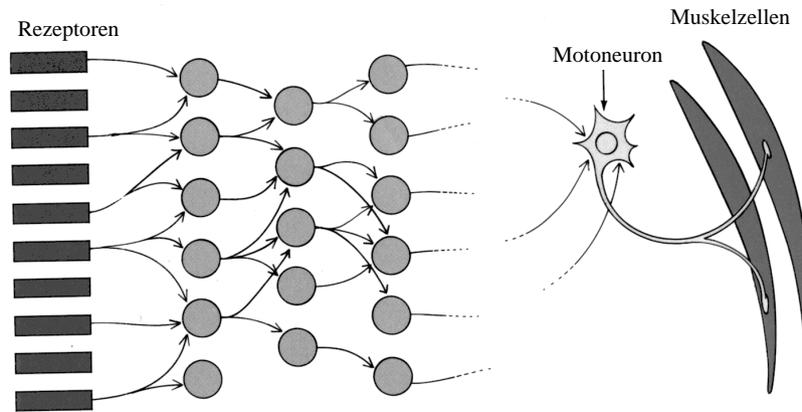


Abbildung 2.3.: **Stark vereinfachte Struktur eines Netzwerks aus Nervenzellen** (nach [Hub90]). An vielen motorischen Regelkreisen im Gehirn sind nur 4-10 Schichten von Nervenzellen beteiligt. Die Zahl der Verbindungen zwischen den Nervenzellen ist in einem realen Netzwerk aber um Größenordnungen höher.

führen zu hochfrequenten *spike*-Folgen [ST85].

Wie oben erwähnt wurde, erfolgt die Signalübertragung zwischen verschiedenen Zellen an den Synapsen. Bei Einlaufen eines elektrischen Signals in die präsynaptische Endigung eines Axons werden dort chemische Transmitterstoffe ausgeschüttet, die durch den synaptischen Spalt diffundieren und an der postsynaptischen Membran ein elektrisches Signal verursachen [ST85]. Das Signal einer Zelle, welches durch das Axon an eine Synapse herangeführt wird, kann auf die postsynaptische Nervenzelle eine erregende (exzitatorische) oder hemmende (inhibitorische) Wirkung haben. Die Stärke der Kopplung zwischen verschiedenen Zellen hängt von der Zahl und Effektivität der synaptischen Verbindungen ab.

Im Zellsoma einer Zelle werden die zumeist über die Dendriten von vielen Seiten eingehenden Signale näherungsweise einfach aufsummiert, wenn man von ebenfalls möglichen, nichtlinearen Interaktionen der Signale absieht. An der Somamembran führt das Überschreiten einer bestimmten Gesamtdepolarisation, die zu einem gegebenen Zeitpunkt auf Grund der Erregung der vielen synaptischen Kontakte vorliegt, zur Ausbildung eines Depolarisationspulses und somit zu einem Ausgabesignal der Zelle, welches über das Axon weitergeleitet wird [ST85]. Die Erregbarkeit einer Nervenzelle ist dabei von vielen Faktoren, wie der Neuronengröße, dem Neuronentyp und der Entfernung der jeweiligen Synapse vom Zellsoma, abhängig.

Abbildung 2.3 faßt die wichtigsten Funktionsbausteine des Nervensystems schematisch zusammen. Man erkennt, wie sensorische Daten über die Rezeptoren einem Netzwerk von Nervenzellen zugeführt werden. Jede solche Zelle empfängt über die Synapsen – hier dargestellt als Pfeilspitzen – Informationen von anderen Nervenzellen. Diese Signale können die Nervenzelle zum Aussenden eigener Signale aktivieren, die über das Axon – die Pfeile – an andere Neuronen weitergegeben werden. Dabei entscheiden die Übertragungseigen-

schaften der Synapsen, ob und in welchem Maße die nachgeschalteten Neuronen aktiviert oder gehemmt werden. Das hier skizzierte Verschaltungsschema ist eine der zentralen Annahmen der Neuroinformatik, nach der sich die Funktionsweise des Gehirns durch die sinnvolle Verschaltung von sehr vielen informationsverarbeitenden Elementen – den Neuronen – erklären läßt [And97].

Im Gegensatz zu der schematischen Darstellung in Abbildung 2.3 sind die Verbindungen innerhalb des Gehirns nicht ausschließlich “vorwärts”, d.h. von den sensorischen Eingängen zu den motorischen Ausgängen, gerichtet. Vielmehr gibt es dort zusätzliche laterale Verbindungen innerhalb der Zellschichten und reziproke Verbindungen zwischen diesen Schichten. Dies trifft auch für die relativ übersichtliche und gut untersuchte Kleinhirnrinde zu, der sich der nächste Abschnitt aus den folgenden Gründen widmet: Einerseits soll durch die Darstellung der Anatomie und Funktion der Kleinhirnrinde eine Vorstellung von der Komplexität der Verschaltung der Nervenzellen im ZNS vermittelt werden. Andererseits dienen, wie oben erwähnt, Aufbau und Funktion der Kleinhirnrinde als biologisches Vorbild bei der Konstruktion der uns interessierenden RBF-Netzwerke [PG89].

2.2.2. Kleinhirnrinde

Das Kleinhirn (*Cerebellum*) dient hauptsächlich der Kontrolle und Feinabstimmung der Körpermotorik. Es ist stammesgeschichtlich älter als das Großhirn (*Cerebrum*) und gliedert sich im wesentlichen in zwei Teile. Die innenliegende *weiße Masse* besteht vor allem aus Nervenfasern. Dort werden die über das Mittelhirn und Großhirn eingehenden Informationen zu geplanten motorischen Aktionen auf die außenliegende Kleinhirnrinde (*cerebellarer Cortex*) verteilt [Sch96]. Diese *graue Masse*, welche in Abbildung 2.4 im Querschnitt dargestellt ist, hat eine stark faltige Struktur und stellt im ausgebreiteten Zustand eine Schicht von etwa 1500 cm^2 dar. Die Kleinhirnrinde ist über weite Gebiete sehr einheitlich aufgebaut und weist eine gewisse Schichtstruktur auf, die durch verschiedene Zelltypen entsteht [Sch96].

Abbildung 2.5 zeigt das Verschaltungsschema der wichtigsten Zelltypen innerhalb der Kleinhirnrinde nach J.C. Eccles und Mitarbeitern [EIS67, Ecc73]. Für die Funktion dieser Verschaltung wurde von D. Marr [Mar69] und J. Albus [Alb71] eine erste Interpretation geliefert.

Nach diesem Modell wird eine in der Großhirnrinde generierte *Bewegungsskizze* über verschiedene Stufen an die Kleinhirnrinde gesendet und schließlich über die sog. Moosfasern an die Körnerzellen herangeführt. Dabei hat eine Moosfaser Verbindungen mit ca. 400 Körnerzellen; eine Körnerzelle hat aber nur etwa vier Verbindungen mit verschiedenen Moosfasern. Die Körnerzellen (beim Menschen etwa 10^{10}) verarbeiten die über die Moosfasern einlaufenden Eingaben und geben das Ergebnis über ihre Axone, die Parallelfasern, in Richtung der Molekularschicht (vgl. Abbildung 2.4) weiter, in der die Ausgabeinformation über eine Umgebung von ca. 1–2 mm Durchmesser verteilt wird. Die Parallelfasern

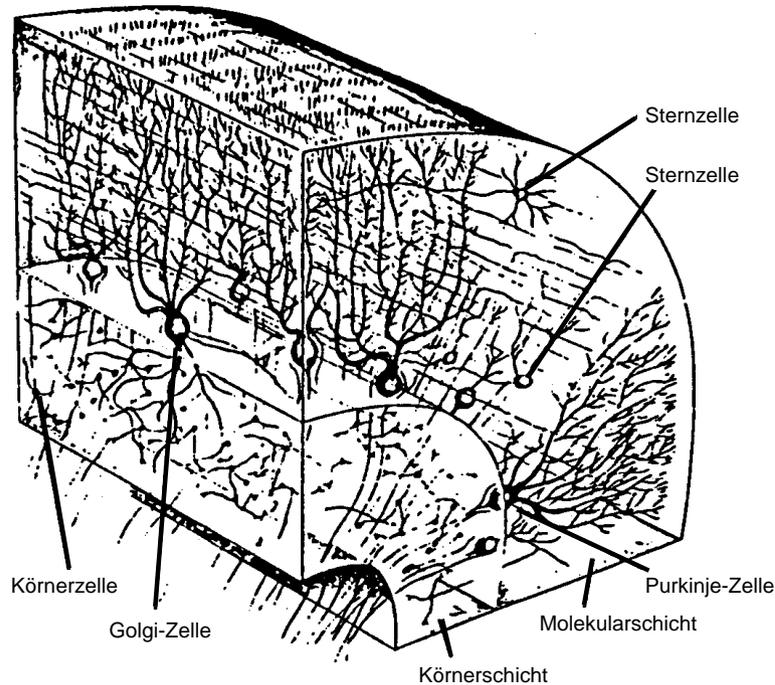


Abbildung 2.4.: **Schnitt durch die Kleinhirnrinde.** Die Dicke des dargestellten Schnitts beträgt ca. 1 mm. Die im Kleinhirn innenliegende weiße Masse befindet sich in der Abbildung links unten (aus [Tho90]).

durchqueren hierbei die Dendritenbäume aller anderen Neuronen und bilden Synapsen mit den Golgi-, Stern-, Korb- und Purkinjezellen [ST85]. Dabei lassen sich zwei Informationsströme unterscheiden:

Zum einen wird über die Golgizellen eine Rückkopplung zu den Körnerzellen aufgebaut. Die Golgizellen sammeln die über die Parallelfasern vermittelten Aktivitäten der Körnerzellen ein, verarbeiten diese und bilden mit ihren Axonen an den Kontaktstellen zwischen Moosfasern und Körnerzellen inhibitorische axon-axonische Synapsen. Diese negative Rückkopplung führt zu einer Regulierung der Körnerzellaktivität und damit der Parallelfaseraktivität.

Zum anderen wird die Aktivität der Körnerzellen an die großen Purkinjezellen weitergegeben. Deren Zahl ist viel kleiner als jene der Körnerzellen (1 : 80000); ihre Axone bilden den einzigen "Ausgang" der Kleinhirnrinde. Diese Ausgabesignale erreichen nach Verschaltungen im Thalamus wieder die motorische Großhirnrinde und über das Rückenmark die verschiedenen Muskeln.

Zu den Purkinjezellen scheint sich eine weitere, hier aber anscheinend positive Rückkopplung aufzubauen. Man findet in der weißen Masse kleine Gebiete mit Anhäufungen von Zellkörpern, die *inferior olivia*. Aus dieser Region gibt es über die Kletterfasern Ver-

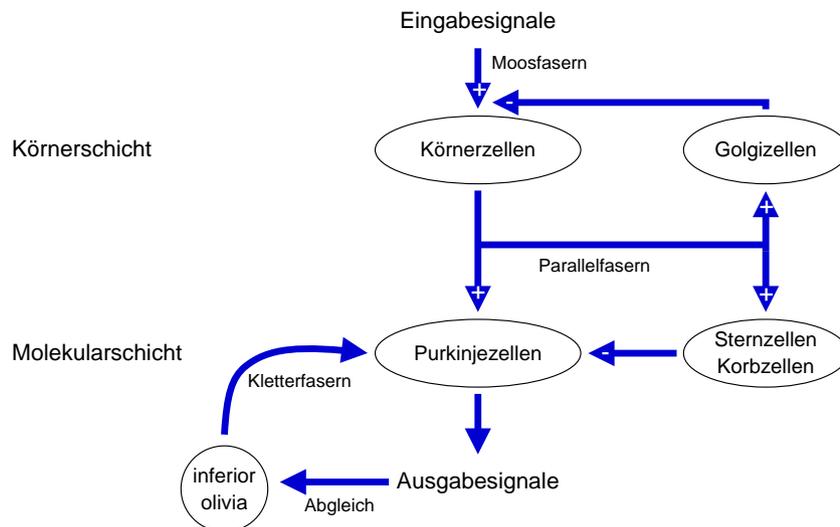


Abbildung 2.5.: **Schematische neuronale Verschaltung im Cerebellum.** Die verschiedenen Nervenzellen sind zusammen mit ihren Dendritenbäumen als Ellipsen dargestellt. Die Pfeile symbolisieren die Axone, die von dem jeweiligen Neuron ausgehen, und die Pfeilspitzen die synaptischen Verbindungen. Eine exzitatorische bzw. inhibitorische Verbindung wird durch ein + bzw. – in der Pfeilspitze charakterisiert (nach [ST85]).

bindungen zu den Dendriten der Purkinjezellen. Dabei “klettern” die Verzweigungen der Kletterfasern an den Ästen der Dendritenbäume der Purkinjezellen hoch und bilden dort zahlreiche exzitatorische Synapsen aus. Jede Purkinjezelle wird von nur einer Kletterfaser erreicht, aber jede Kletterfaser versorgt 10–15 Purkinjezellen. Ein einzelner Impuls in einer Kletterfaser führt daher zu einer mehrfachen Entladung der aktivierten Purkinjezellen.

Sieht man von den durch die Parallelfasern aktivierten Purkinjezellen ab, so sind die Wirkungen der übrigen, von diesen Fasern aktivierten Zellen inhibitorisch. Beispielsweise hemmen die Golgizellen die Körnerzellen (*feedback*-Hemmung), während die Sternzellen und die Korbzellen hemmend auf die Purkinjezellen wirken (*feedforward*-Hemmung). Bis auf die Körner- und Purkinjezellen, sowie die Zellen der inferior olivaria, wirken also die meisten Neuronen der Kleinhirnrinde inhibitorisch. An keiner anderen Stelle des Zentralnervensystems ist ein derartiges Überwiegen der Hemmung bekannt [ST85].

Nach der Interpretation von D. Marr [Mar69] und J. Albus [Alb71] kann man sich die große Überzahl an Körnerzellen gegenüber den Moosfasern als ein Abbildung der über die Moosfasern gelieferten Eingabe in einen höherdimensionalen Raum vorstellen. Die hochdimensionale Ausgabe der Körnerzellen stellt somit eine sehr redundante Transformation der Eingabedaten dar, welche zusätzlich durch die inhibitorische Rückkopplung über die Golgizellen moduliert wird. Die Signale der Purkinjezellen spannen schließlich wegen ihrer geringeren Anzahl einen Raum von deutlich niedrigerer Dimension auf, auf den die Signale der Körnerzellen abgebildet werden. Die Rückkopplung über die inferior olivaria hat dabei die Eigenschaft, daß die exzitatorischen Verbindungen zwischen Parallelfasern und

Purkinjezellen verstärkt werden, wenn gleichzeitig ein Signal über die Kletterfasern und den Parallelfasern anliegt [Mas84]. Eine derartige Verstärkung synaptischer Verbindungen auf Grund gleichzeitiger Aktivitäten von prä- und postsynaptischen Neuronen wurde erstmals im Jahre 1949 von D. Hebb als Lernregel für die Modifikation synaptischer Kontakte postuliert [Heb49]. Inzwischen gibt es Hinweise auf die Wirksamkeit dieser Hebbschen Regel im zentralnervösen Gewebe [KGB86]. Im gegebenen Fall des Cerebellums “belohnt” die inferior olivaria also richtige Aktionen, die durch die Purkinjezellen ausgelöst werden, durch Wachstum der zugehörigen Synapsenstärken.

1989 diskutierten T. Poggio und F. Girosi [PG89] erstmals den Zusammenhang zwischen einem RBF-Netzwerk und dem Modell des Cerebellums von D. Marr und J. Albus. Dieser Zusammenhang und mithin die biologische Motivierung von RBF-Netzwerken wird in Abschnitt 2.4 noch genauer erläutert werden. Um die mathematischen Eigenschaften von RBF-Netzwerken gründlich diskutieren zu können, ist es jedoch zunächst notwendig sich mit einer Reihe von mathematischen Konzepten zu befassen. Diese Konzepte stammen aus den Gebieten der statistischen Datenanalyse, der Mustererkennung, der Wahrscheinlichkeitstheorie und der Optimierung. Wie an Hand des RBF-Modells gezeigt werden soll, hat die Natur mit der Konstruktion des Cerebellums anscheinend ein mathematisches Instrument geschaffen, dessen selbstorganisierende Verschaltungs- und Funktionsdynamik eine Vielzahl wohlbekannter, schwieriger Probleme aus diesen klassischen Wissenschaftszweigen zu lösen vermag.

2.3. Datenanalyse

Viele der intelligenten Fähigkeiten höherer Lebewesen beruhen auf einer Analyse der von den Sinnesorganen eingehenden Informationen: Vögeln gelingt eine punktgenaue Landung auf einem schwankenden Ast, Lebewesen können aus vielen Beobachtungen einen Erfahrungsschatz aufbauen und Menschen haben die bemerkenswerte Fähigkeit Gesichter auch nach langer Zeit wieder zu erkennen. Die Grundlagen für alle diese Fähigkeiten sind zwar angeboren. Viele dieser Fähigkeiten entfalten sich aber erst durch intensives Training. Es sollen nun einige jener Begriffe eingeführt werden, mit deren Hilfe sich die Fähigkeiten zur Analyse und Speicherung sensorischer Informationen, sowie die zugehörigen Lernprozesse, mathematisch formulieren lassen. Dabei werden wir insbesondere Ansätze aus der klassischen statistischen Datenanalyse genauer darstellen und mit dem Problem der Funktionenapproximation beginnen.

2.3.1. Funktionenapproximation

Der Begriff der *Funktionenapproximation* bezeichnet mathematische Verfahren, die darauf abzielen, den lediglich an Hand von Messungen zugänglichen Zusammenhang zwischen Eingaben $x \in \mathbb{R}^D$ und Ausgaben $y \in \mathbb{R}^K$ eines beliebigen Systems durch eine

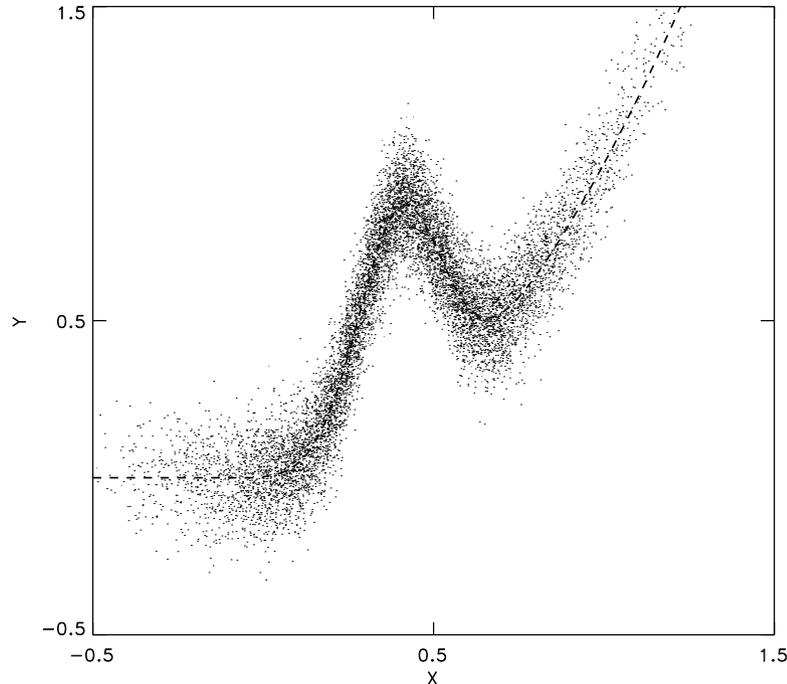


Abbildung 2.6.: **Funktionenapproximation.** Die Punkte zeigen alle Elemente eines Datensatzes \mathcal{Z} , welcher einen funktionalen Zusammenhang zwischen den jeweils eindimensionalen Eingabedaten \mathcal{X} und Ausgabedaten \mathcal{Y} aufweist. Dieser Datensatz wurde erzeugt, indem zu der Funktion $f(x) = x^2 + 0.7 \exp[-\frac{1}{2} \frac{(x-0.4)^2}{0.015}]$, die als gestrichelte Linie eingetragen ist, ein Rauschterm η addiert wurde. Der Datensatz \mathcal{X} ist im Intervall $[-0.5, 1.5]$ nicht gleichverteilt, sondern entsprechend einer Normalverteilung gewürfelt. Diese Normalverteilung hat bei $x = 0.5$ ihr Maximum und fällt bis $x = -0.5$ bzw. $x = 1.5$ stark ab.

analytische Funktion zu approximieren. Abbildung 2.6 zeigt ein einfaches, niedrigdimensionales Beispiel für diese Aufgabenstellung. Die Mengen der beobachteten Eingabedaten $\mathcal{X} = \{\mathbf{x}_k | k = 1 \dots N\} \subset \mathbb{R}^D$ und Ausgabedaten $\mathcal{Y} = \{\mathbf{y}_k | k = 1 \dots N\} \subset \mathbb{R}^K$ werden als gegeben vorausgesetzt und zum Gesamtdatensatz $\mathcal{Z} = \{\mathcal{X}, \mathcal{Y}\}$ zusammengefaßt. Es wird angenommen, daß der Datensatz \mathcal{Z} einer Abbildung

$$\mathbf{y}_k = \mathbf{f}(\mathbf{x}_k) + \boldsymbol{\eta}, \quad k = 1 \dots N \quad (2.1)$$

gehört. Dabei ist $\mathbf{f}(\mathbf{x})$ der dem Datensatz \mathcal{Z} zu Grunde liegende funktionale Zusammenhang und $\boldsymbol{\eta}$ ein nicht näher bekannter Rauschterm, der etwa Meßfehler bei der Beobachtung eines Datenpunktes $(\mathbf{x}_k, \mathbf{y}_k)$ berücksichtigt. Gesucht ist nun eine Funktion $\hat{\mathbf{f}}(\mathbf{x}; \Theta)$, die den im Datensatz \mathcal{Z} verborgenen funktionalen Zusammenhang $\mathbf{f}(\mathbf{x})$ möglichst gut widerspiegelt. Mit einem zu definierenden Ähnlichkeitsmaß sollte also

$$\hat{\mathbf{f}}(\mathbf{x}; \Theta) \approx \mathbf{f}(\mathbf{x}) \quad (2.2)$$

gelten. Hier ist Θ die Menge aller Parameter, die in die Funktion $\hat{f} : \mathbb{R}^D \rightarrow \mathbb{R}^K$ eingehen. Die Bestimmung eines Parametersatzes Θ , für den (2.2) gilt, ist damit ein Optimierungsproblem, an welches die folgenden Anforderungen gestellt werden: Die Funktion $\hat{f}(\mathbf{x}; \Theta)$ soll einerseits über eine hohe *Approximierungsqualität* verfügen, um den Zusammenhang zwischen \mathcal{X} und \mathcal{Y} im Datensatz \mathcal{Z} gut wiedergeben zu können. Es sollte also gelten

$$\hat{f}(\mathbf{x}_k; \Theta) \approx \mathbf{y}_k, \quad k = 1 \dots N. \quad (2.3)$$

Andererseits soll $\hat{f}(\mathbf{x}; \Theta)$ im Sinne von (2.2) auch *generalisierungsfähig* sein. Der Rauschterm $\boldsymbol{\eta}$ aus (2.1) sollte daher möglichst weitgehend eliminiert werden.

Bei der Berechnung des Parametersatzes Θ kann zwischen parametrischen und nicht-parametrischen Verfahren unterschieden werden [Bis95]. Bei einem nicht-parametrischen Verfahren werden alle Parameter Θ einfach aus \mathcal{Z} abgelesen und nicht durch einen speziellen Algorithmus berechnet.

Betrachten wir als Beispiel für ein nicht-parametrisches Verfahren die lineare Interpolation eines eindimensionalen Datensatzes ($D = 1, K = 1$). Hier wird die Funktion $f(x)$ approximiert, indem der Funktionswert $\hat{f}(x; \Theta)$ an einer im Datensatz nicht auffindbaren Stelle x , die zwischen direkt benachbarten Datenpunkten x_{k_1} und x_{k_2} liegt, gemäß

$$\hat{f}(x; \Theta) = y_{k_1} + \frac{y_{k_2} - y_{k_1}}{x_{k_2} - x_{k_1}}(x - x_{k_1}) \quad (2.4)$$

aus den zugehörigen y -Werten y_{k_1} und y_{k_2} berechnet wird.

Die Approximierungsqualität dieses nicht-parametrischen Verfahrens ist wegen $\hat{f}(x_k; \Theta) \equiv y_k, k = 1 \dots N$, optimal. Allerdings ist seine Generalisierungsfähigkeit gering, da keinerlei Anstrengung unternommen wurde, das Rauschen $\boldsymbol{\eta}$ zu unterdrücken. Der Vorteil, daß die Parameter Θ nicht aus \mathcal{Z} berechnet werden müssen, sondern durch \mathcal{Z} unmittelbar gegeben sind, wird durch den Nachteil erkauft, daß die Parameterzahl $|\Theta|$ genau so groß ist wie der Datensatz \mathcal{Z} .

Eine Alternative sind parametrische Methoden, die versuchen mit sehr viel kleineren Parametersätzen auszukommen und eine höhere Generalisierungsfähigkeit erreichen. Dabei stellt sich zunächst die nicht-triviale Aufgabe der Wahl einer geeigneten analytischen Darstellung der Funktionenschar $\hat{f}(\mathbf{x}; \Theta)$. Wenn das zu untersuchende System einer bestimmten Problemklasse angehört, so kann häufig mittels einer Analyse des Systems eine geeignete Funktionenschar $\hat{f}(\mathbf{x}; \Theta)$ ausgewählt werden². Soll der parametrische Funktionenapproximator jedoch auf eine große Klasse von Systemen anwendbar sein, so ist die Wahl

²Als Beispiel sei die wiederholte Messung des Strömungswiderstandes F eines Körpers in einer turbulenten Strömung bei verschiedenen Geschwindigkeiten v genannt. Da für den turbulenten Strömungswiderstand der Zusammenhang $F = c_w A \frac{\rho}{2} v^2$ bekannt ist [LG88], ist es auch sinnvoll eine Funktionenapproximation mit $F(v; \Theta = \{c_w\})$ auszuführen. A ist hierbei der größte der Strömung entgegenstehende Körperquerschnitt und ρ die Dichte des strömenden Mediums. Der Widerstandsbeiwert c_w ist ein Parameter und kann mit einem Regressionsverfahren aus dem Datensatz \mathcal{Z} bestimmt werden.

der Funktionenschar $\hat{\mathbf{f}}(\mathbf{x}; \Theta)$ schwieriger. Es müssen dann Regularisierungsannahmen, wie etwa Glattheitsannahmen, gemacht werden, welche in allgemeiner Form versuchen, die Eigenschaften der jeweiligen Systemklassen zu beschreiben. Ist eine Funktionenschar $\hat{\mathbf{f}}(\mathbf{x}; \Theta)$ festgelegt, müssen schließlich die Parameter Θ an den Datensatz \mathcal{Z} angepaßt werden. Hier stellt sich die Frage, nach welchem Optimierungskriterium dies erfolgen und mit welchem Algorithmus die Parameter Θ berechnet werden sollen.

Eine verbreitete Form zur Parameterbestimmung ist die *Regression* [MKB94]. Das Optimierungskriterium ist hier die Forderung nach der Minimierung der mittleren quadratischen Abweichung

$$E(\Theta; \mathcal{Z}) = \frac{1}{N} \sum_{k=1}^N |\hat{\mathbf{f}}(\mathbf{x}_k; \Theta) - \mathbf{y}_k|^2 \quad (2.5)$$

der Approximation des Funktionswerts $\hat{\mathbf{f}}(\mathbf{x}_k; \Theta)$ am Eingabedatenpunkt \mathbf{x}_k vom zugehörigen Ausgabedatenpunkt \mathbf{y}_k für den gesamten Datensatz \mathcal{Z} . Mit der Definition des Mittelungssymbols $\langle \cdot \rangle$ durch

$$\langle \mathbf{g}(\mathbf{x}) \rangle_{\mathcal{X}} \equiv \frac{1}{N} \sum_{k=1}^N \mathbf{g}(\mathbf{x}_k), \quad \mathcal{X} = \{\mathbf{x}_k | k = 1 \dots N\} \quad (2.6)$$

kann dies auch einfacher als

$$E(\Theta; \mathcal{Z}) = \langle |\hat{\mathbf{f}}(\mathbf{x}_k; \Theta) - \mathbf{y}_k|^2 \rangle_{\mathcal{Z}} \quad (2.7)$$

geschrieben werden. Die Stationaritätsbedingungen

$$\frac{\partial}{\partial \theta} E(\Theta; \mathcal{Z}) = 0 \quad \forall \theta \in \Theta \quad (2.8)$$

repräsentieren dann die notwendigen Voraussetzungen für das Vorliegen eines Minimums des Fehlerfunktionals $E(\Theta; \mathcal{Z})$.

Regression mit Polynomen

Eine häufige Wahl für die Funktionenschar $\hat{\mathbf{f}}(\mathbf{x}_k; \Theta)$ sind bei eindimensionalen Ein- und Ausgabedatensätzen die Polynome

$$\hat{f}(x; \Theta) = \sum_{j=0}^L w_j x^j \quad (2.9)$$

der Ordnung L . Dieser Ansatz führt mit (2.8) zur *polynomialen Regression*, bei der die Parameter $\Theta = \{w_0 \dots w_L\}$ durch das lineare Gleichungssystem

$$\sum_{j=0}^L w_j \langle x^{j+l} \rangle = \langle y x^l \rangle, \quad l = 0 \dots L \quad (2.10)$$

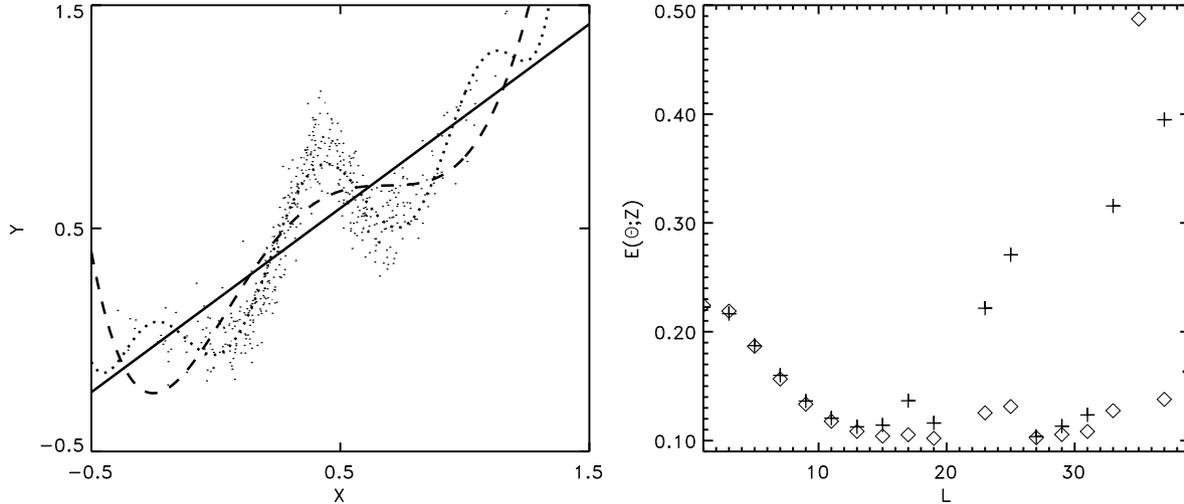


Abbildung 2.7.: **Polynomiale Regression.** Links ist das Ergebnis einer polynomialen Regression der Daten aus Abbildung 2.6 mit einem Polynom der Ordnung $L = 1$ (durchgezogene Linie), $L = 5$ (gestrichelt) und $L = 10$ (gepunktet) gezeigt. Rechts ist der Fehler (2.7) in Abhängigkeit von der Polynomordnung L dargestellt. Dabei wird zwischen $E(\Theta; \mathcal{Z})$ (\diamond) und $E(\Theta; \mathcal{Z}_{\text{test}})$ (+) unterschieden.

bestimmt sind. Für den trivialen Fall der Approximation durch eine konstante Funktion $\hat{f}(x; \Theta = w_0) = w_0$, also $L = 0$, erhält man aus (2.10) $w_0 = \langle y \rangle$. Der gesuchte Zusammenhang zwischen Ein- und Ausgabedatensatz wird also gerade durch den Mittelwert des Ausgabedatensatzes \mathcal{Y} genähert. Die *lineare Regression* mit einer Ausgleichsgeraden ($L = 1$) liefert die Funktionenapproximation

$$\hat{f}(x; \Theta) = w_0 + w_1 x \quad (2.11)$$

mit

$$w_1 = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2} \quad \text{und} \quad w_0 = \langle y \rangle - w_1 \langle x \rangle, \quad (2.12)$$

in die, neben den Mittelwerten $\langle x \rangle$, $\langle y \rangle$, die Korrelation $\langle xy \rangle - \langle x \rangle \langle y \rangle$ der Ein- und Ausgabedaten sowie die Varianz $\langle x^2 \rangle - \langle x \rangle^2$ der Eingabedaten eingehen. Die beiden Beispiele für $L = 0$ und $L = 1$ verdeutlichen, wie sich der Parametersatz Θ aus der Berechnung statistischer Momente ergibt.

Abbildung 2.7 zeigt für den Datensatz aus Abbildung 2.6 das Ergebnis der polynomialen Regression mit Polynomen der Ordnungen $L = 1, 5$ und 10 . Ebenfalls dargestellt sind die Fehler (2.7) dieser Regressionen in Abhängigkeit von der Ordnung L der Polynome. Dabei wird zwischen dem Fehler für den Datensatz \mathcal{Z} , an Hand dessen die Parameter nach (2.10) bestimmt wurden, und dem Fehler für einen weiteren Datensatz $\mathcal{Z}_{\text{test}}$, der durch

denselben stochastischen Prozeß generiert wurde, unterschieden. Bei \mathcal{Z} spricht man auch vom *Trainingsdatensatz* und bei $\mathcal{Z}_{\text{test}}$ vom *Testdatensatz*.

Für den Trainingsdatensatz ist in Abbildung 2.7 zu erkennen, wie die steigende Polynomordnung L allmählich zu einer Abnahme des Fehlers $E(\Theta; \mathcal{Z})$ führt und somit die Qualität der Approximation zunimmt. Andererseits wird der Fehler $E(\Theta; \mathcal{Z}_{\text{test}})$ für den Testdatensatz bei $L > 15$ wieder größer. Wenn der Datensatz durch ein Polynom zu hoher Ordnung approximiert wird, nimmt also die Generalisierungsfähigkeit ab. Abbildung 2.7 zeigt aber auch, daß die Qualität der Funktionenapproximation für große L nicht unbedingt mit L zunimmt. Der Grund dafür ist in der Notwendigkeit zu sehen, daß für zunehmende L immer höhere statistische Momente berechnen werden müssen. Für einen Datensatz \mathcal{Z} mit endlich vielen Elementen sind diese aber mit einem immer größeren statistischen Fehler behaftet. Eine Erhöhung der Polynomordnung führt also nicht zwingend zu einer Verbesserung der Approximierungsqualität und Generalisierungsfähigkeit [Bis95].

Für den allgemeineren³ Fall einer hochdimensionalen Eingabe ($D \neq 1$) und einer eindimensionalen Ausgabe ($K = 1$) kann die polynomiale Regression natürlich auch auf multivariable Polynome angewendet werden. Die Bestimmung der Parameter Θ beruht dann wieder auf der Berechnung verschiedener statistischer Momente von \mathcal{X} und \mathcal{Y} . Für ein Polynom dritter Ordnung müßte z.B.

$$\hat{f}(\mathbf{x}; \Theta) = w_0 + \sum_{i_1=1}^D w_{i_1} x_{i_1} + \sum_{i_1=1}^D \sum_{i_2=1}^D w_{i_1 i_2} x_{i_1} x_{i_2} + \sum_{i_1=1}^D \sum_{i_2=1}^D \sum_{i_3=1}^D w_{i_1 i_2 i_3} x_{i_1} x_{i_2} x_{i_3}$$

geschrieben werden. Bei einer solchen Abbildung gilt für die Parameterzahl $|\Theta| \sim D^L$. Bei einem gegebenen hochdimensionalen Datensatz \mathcal{Z} hat diese rasante Zunahme der Parameterzahl zur Folge, daß mit zunehmender Ordnung L für die Berechnung der einzelnen Parameter immer weniger Datenpunkte zur Verfügung stehen. Daher nimmt der statistische Fehler der Parameterbestimmung mit steigendem L rasch zu. Für hochdimensionale Abbildungen müssen daher sehr große Trainingsdatensätze zur Verfügung stehen, um eine gute Qualität der Approximation zu erhalten. Auch wird die Fehlerfunktion (2.7) für viele Parameter immer komplexer und ihre Optimierung immer schwieriger [Bis95].

2.3.2. Mustererkennung

Eine herausragende Leistung vieler Lebewesen ist die Mustererkennung bzw. Klassifikation. Beim Menschen wird diese Fähigkeit beispielsweise bei der Gesichts- oder der Spracherkennung deutlich. Hierbei wird einem sehr hochdimensionalen⁴ Eingabesignal, welches sich aus optischen oder akustischen Sinneswahrnehmungen zusammensetzt, eine Person oder eine bestimmte sprachliche Äußerung zugeordnet. Bemerkenswert ist, daß diese

³Tatsächlich ist dies der allgemeine Fall, da jede Abbildung $f : \mathbb{R}^D \rightarrow \mathbb{R}^K$ in K Einzelabbildungen $f_k : \mathbb{R}^D \rightarrow \mathbb{R}$ zerlegt werden kann.

⁴Die Netzhaut des menschlichen Auges enthält ungefähr 10^8 lichtempfindliche Rezeptoren. Nach einer Vorverarbeitung innerhalb der Netzhaut wird die visuelle Wahrnehmung über den Sehnerv, welcher et-

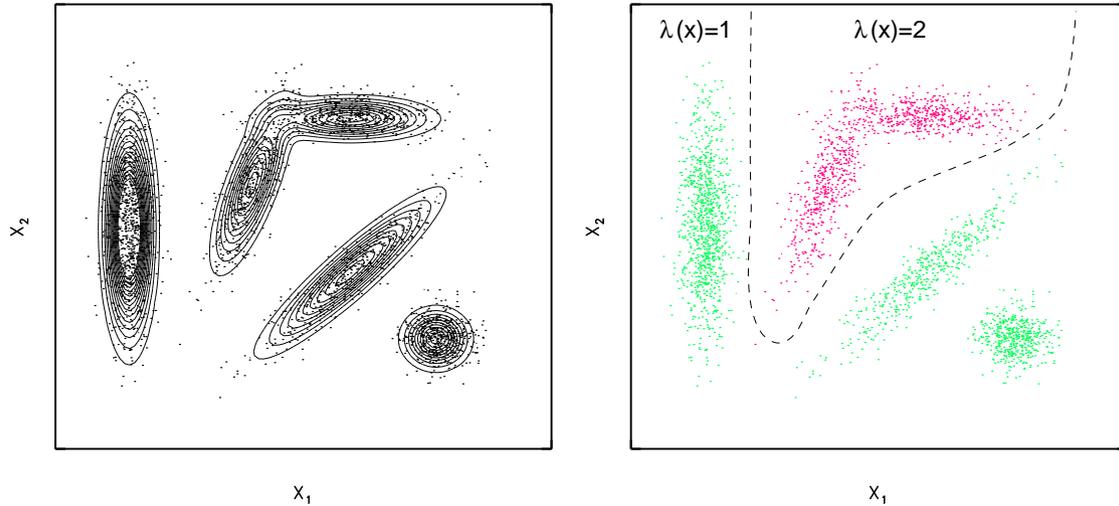


Abbildung 2.8.: **Mustererkennung.** Links sind $N = 3000$ Merkmalsvektoren \mathbf{x}_k eines zwei-dimensionalen Datensatzes \mathcal{X} dargestellt. Zusätzlich ist die Wahrscheinlichkeitsdichte $p(\mathbf{x})$, nach der die gezeigten Punkte gewürfelt wurden, in einer Höhenliniendarstellung eingetragen. Die Daten gliedern sich in fünf Häufungen. Im ganz linken Haufen befinden sich 1000 Punkte und in den restlichen vier jeweils 500 Punkte. Im rechten Teilbild ist jeder Merkmalsvektor des Datensatzes \mathcal{X} einer von zwei Klassen $\lambda \in \{1, 2\}$ zugeordnet; die Klassenzugehörigkeiten λ_k werden durch die Farbkodierungen rot und grün unterschieden. Die gestrichelt eingezeichnete Linie markiert eine mögliche Grenze zwischen den beiden Klassen. Durch eine solche Grenze werden zwei Gebiete des Merkmalsraums abgegrenzt, auf denen die zugehörige *Klassifikationsfunktion* $\lambda(\mathbf{x})$ jeweils die Werte $\lambda(\mathbf{x}) = 1$ (grün) und $\lambda(\mathbf{x}) = 2$ (rot) annimmt. Verwendet man diese Funktion zur Klassifikation der Daten, so werden nur wenigen Objekten aus der Klasse “rot” irrtümlich der Funktionswert $\lambda(\mathbf{x}) = 1$ zugeordnet.

Erkennung auch auf der Basis unvollständiger Eingabedaten relativ gut ausgeführt werden kann: Ein Gesicht ist von der Seite zu erkennen, auch über schlechte Telefonleitungen ist ein Gespräch möglich.

Typische Mustererkennungsprobleme sind die Spracherkennung [Alb95, Der95] oder die Gesichtserkennung [Küh94, LVB⁺93]. Abbildung 2.8 illustriert an Hand eines sehr einfachen Beispiels die Aufgabenstellung, die bei der Mustererkennung zu bewältigen ist. Dem Beispiel liegt ein fiktives Szenario zugrunde, welches so oder ähnlich in jedem Lehrbuch zur Mustererkennung anzutreffen ist (z.B. [DH73]). Dabei wird angenommen, daß Objekte k aus einer Menge \mathcal{K} durch zwei Sensoren meßtechnisch erfaßt werden. Für jedes Objekt k werden beide Sensorsignale x_{1k} und x_{2k} zu einem *Merkmalsvektor* \mathbf{x}_k zusammengefaßt, der als Ortsvektor in einem zugehörigen *Merkmalsraum* interpretiert wird. Ein Objekt wird

wa 10^6 Nervenfasern umfaßt, in den visuellen Cortex übertragen [Hub90]. Auch wenn die Signale der vielen Nervenfasern stark korrelieren, so hat der durch diese Signale aufgespannte Raum eine gewaltige Dimension.

so durch die Messung ausgewählter Merkmale charakterisiert. Jedes sensorisch beobachtete Objekt kann daher als Punkt im Merkmalsraum dargestellt werden, wobei die Dimension dieses Raums durch die Zahl der Sensoren gegeben ist. Entsprechend wird die Menge \mathcal{K} durch einen Datensatz $\mathcal{X} = \{\mathbf{x}_k | k = 1 \dots N\}$ und mithin durch eine Punktverteilung im Merkmalsraum repräsentiert. Das linke Teilbild von Abbildung 2.8 zeigt ein Beispiel für eine solche Verteilung. Zusätzlich ist bei diesem numerisch generierten Beispiel noch die bei der Erzeugung verwendete Modelldichte $p(\mathbf{x})$ angegeben. Bei tatsächlichen Klassifikationsproblemen ist diese Dichte selbstverständlich nicht bekannt.

Im Beispiel aus Abbildung 2.8 wird ferner angenommen, daß die Objekte aus zwei Klassen $\lambda = 1, 2$ stammen. Die Aufgabe eines mustererkennenden Systems oder *Klassifikators* ist es nun, die Objekte ausschließlich an Hand der sensorischen Beobachtung \mathbf{x} ihrer jeweiligen Klasse $\lambda(\mathbf{x})$ zuzuordnen. In Abbildung 2.8 ist eine mögliche Klassifikationsfunktion $\lambda(\mathbf{x})$ durch die Angabe einer Grenzlinie definiert, bei der $\lambda(\mathbf{x})$ vom Wert $\lambda(\mathbf{x}) = 1$ auf den Wert $\lambda(\mathbf{x}) = 2$ springt. Als Wissensbasis für die Konstruktion eines solchen Klassifikators dient ein Trainingsdatensatz $\mathcal{Z} = \{\mathcal{X}, \mathcal{Y}\}$, der neben den Meßergebnissen \mathbf{x}_k auch noch die Klassenzugehörigkeiten λ_k enthält. In Abbildung 2.8, rechts wurde der Datensatz \mathcal{X} aus dem linken Teilbild zu einem solchen Trainingsdatensatz \mathcal{Z} erweitert, wobei zur Angabe der Klassenzugehörigkeiten λ_k die Farbkodierung grün für $\lambda_k = 1$ und rot für $\lambda_k = 2$ verwendet wurde.

Aus dem angeführten Beispiel kann nun zunächst geschlossen werden, daß die Mustererkennung als Spezialfall der Funktionenapproximation aufgefaßt werden kann. Gilt es doch hier, aus Beispieldaten $\{\mathbf{x}_k, \lambda_k | k = 1 \dots N\}$ eine Klassifikationsfunktion $\lambda(\mathbf{x})$ zu extrahieren. Andererseits scheint der Fall besonders einfach zu sein, da die Klassifikationsfunktion nur endlich viele Werte annehmen kann und über weite Bereiche des Merkmalraums konstant sein sollte. Eine solche Klassifikationsfunktion führt so zu einer Aufteilung des Merkmalsraums in Raumgebiete, die mit den verschiedenen Objektklassen assoziiert sind. Man spricht bei diesem Vorgang auch von der *Partitionierung* des Merkmalsraums. Die Trennflächen zwischen den verschiedenen Partitionen werden durch die *Klassengrenzen* gebildet. In Abbildung 2.8, rechts, ist eine mögliche Klassengrenze eingezeichnet, die zu einer sehr guten Klassifikation führen würde.

In den nächsten beiden Abschnitten werden zwei Methoden zur Mustererkennung diskutiert, die sich unter anderem in der Art der Partitionierung unterscheiden. Bei dem *nächsten Nachbar Erkennen* wird eine *scharfe* Partitionierung des Merkmalsraums vorgenommen. Dies bedeutet, daß jeder Raumpunkt genau einer Partition zugeordnet wird. Der anschließend dargestellte *Bayessche Erkennen* beruht dagegen auf einer *unscharfen* Partitionierung, bei der ein Raumpunkt mehreren Partitionen zugeordnet werden kann.

Nächster Nachbar Erkennen

Ein naheliegendes Konzept zur Konstruktion eines Erkenners besteht in der geeigneten Bestimmung einer geringen Anzahl $M < N$ von *prototypischen* Punkten $\mathcal{C} = \{\mathbf{c}_r | r =$

$1 \dots M\} \subset \mathbb{R}^D$, deren Klassenzugehörigkeit λ_r bekannt ist [Rip96]. Wird ein zu klassifizierender Merkmalsvektor \mathbf{x} dem Erkenner präsentiert, so wird zunächst entsprechend einer Metrik $|\cdot|$ und der Vorschrift

$$r' = \arg \min_{r=1 \dots M} |\mathbf{c}_r - \mathbf{x}| \quad (2.13)$$

der zu \mathbf{x} nächste prototypische Punkt $\mathbf{c}_{r'}$ bestimmt und anschließend die Klassenzugehörigkeit $\lambda_{r'}$ dieses nächsten Nachbarn als erkannte Klasse $\lambda(\mathbf{x})$ angenommen. Die Schwierigkeit dieses Verfahrens liegt in der Auswahl der Punktmenge \mathcal{C} , welche die prototypischen Punkte \mathbf{c}_r umfaßt.

Die Bestimmung einer solchen Punktmenge \mathcal{C} wird auch als *Vektorquantisierung* (VQ) oder *clustering* [DH73, Der95] bezeichnet. Diese Methoden zielen auf eine optimale Beschreibung eines großen Datensatzes \mathcal{X} durch eine kleinere Menge \mathcal{C} von *repräsentativen*, *typischen* Datenpunkten ab. \mathcal{C} bezeichnet man üblicherweise als *Kodebuch* und ihre Elemente \mathbf{c}_r als *Kodebuchvektoren* oder *Prototypen*. Viele technische Anwendungen im Bereich der Datenverarbeitung erfordern eine solche reduzierte Darstellung in Form eines Kodebuchs, um eine effiziente Speicherung, Übermittlung und Verarbeitung von Daten zu erreichen.

In der Literatur gibt es eine Vielzahl von Vorschlägen für *clustering*-Verfahren, die unterschiedlich begründet werden und jeweils ihre Vorzüge und Nachteile haben. So wurden aus ingenieurstechnischer Sicht algorithmisch einfach zu handhabende Methoden wie etwa der LBG-Algorithmus [LBG80] oder das *k-means clustering* [Mac67] entwickelt, die auf die Optimierung unterschiedlicher Fehlermaße zur Repräsentation von Daten zielen. Diese Verfahren weisen jedoch alle sehr schwerwiegende Konvergenzprobleme auf (siehe z.B. [Küh94] für einen Überblick). Aus Überlegungen zu selbstorganisierenden neuronalen Netzen stammen dagegen *clustering*-Methoden wie die selbstorganisierenden Merkmalskarten von T. Kohonen [Koh82, RMS92, DT94a] oder der *neural-gas* Algorithmus von Martinetz et al. [MS91], die zwar sicher konvergieren, deren Optimierungsziel aber häufig nicht klar definiert ist. Im folgenden wird daher ein Verfahren vorgestellt, das sowohl über gute Konvergenzeigenschaften als auch über eine klare mathematische Begründung verfügt.

RGF-Algorithmus

K. Rose, E. Gurewitz und G. Fox haben 1990 ein *clustering*-Verfahren vorgeschlagen, das auf Konzepten aus der statistischen Mechanik beruht [RGF90]. Als Fehlermaß, das durch ein optimales \mathcal{C} minimiert werden soll, wählten sie

$$U(\mathcal{C}, \mathcal{V}) = \sum_{k=1}^N \sum_{r=1}^M v_{kr} (\mathbf{x}_k - \mathbf{c}_r)^2. \quad (2.14)$$

Dabei beschreibt die Menge $\mathcal{V} = \{v_{kr} \in \{0, 1\} | k = 1 \dots N, r = 1 \dots M\}$ die Zuordnungen der Datenpunkte \mathbf{x}_k zu jeweils genau einem Kodebuchvektor \mathbf{c}_r , d.h. es gilt die Nebenbedingung $\sum_r v_{kr} = 1$. Für ein gegebenes \mathcal{V} sind die optimalen \mathbf{c}_r die Zentroide $\mathbf{c}_r = \sum_k v_{kr} \mathbf{x}_k / \sum_k v_{kr}$, so daß U nur von \mathcal{V} abhängt, d.h. $U \equiv U(\mathcal{V})$. Allerdings ist die Berechnung desjenigen Satzes \mathcal{V} von Zuordnungsvariablen v_{kr} , der $U(\mathcal{V})$ minimiert, ein höchst kompliziertes, NP-vollständiges Optimierungsproblem [GJ79].

Um dieses Problem zu lösen, kann man nach K. Rose et al. das Fehlerfunktional $U(\mathcal{V})$ als die *Energie* eines physikalischen Systems mit den dynamischen Variablen v_{kr} interpretieren, ein kanonisches Ensemble solcher Systeme betrachten und versuchen, zunächst das Minimum der zugehörigen freien Energie bei endlicher Temperatur zu bestimmen (vgl. auch [Küh94]). Ein Abkühlen des Ensembles führt dann zum Grundzustand \mathcal{V}_0 des Systems mit minimaler Energie $U(\mathcal{V}_0)$. Da die freie Energie dieses kanonischen Ensembles nicht einfach berechnet werden kann, schlugen K. Rose et al. vor, das Problem im Rahmen der *mean-field* Näherung zu behandeln. In dieser Näherung ist die freie Energie bei der Temperatur $T \equiv 2\sigma^2$ eine Funktion des Kodebuchs \mathcal{C} und ist gegeben durch

$$F(\mathcal{C}, \sigma) = -2\sigma^2 \sum_{k=1}^N \ln \left(\sum_{r=1}^M \exp[-(\mathbf{x}_k - \mathbf{c}_r)^2 / 2\sigma^2] \right). \quad (2.15)$$

Sie nimmt ihr Minimum bei den Erwartungswerten $P(r|\mathbf{x}, \bar{\mathcal{C}}, \sigma) \equiv \bar{v}_{kr}$ der Zuordnungsvariablen v_{kr} an, die durch

$$P(r|\mathbf{x}, \bar{\mathcal{C}}, \sigma) = \frac{\exp[-(\mathbf{x} - \bar{\mathbf{c}}_r)^2 / 2\sigma^2]}{\sum_{r'=1}^M \exp[-(\mathbf{x} - \bar{\mathbf{c}}_{r'})^2 / 2\sigma^2]} \quad (2.16)$$

gegeben sind und Zuordnungswahrscheinlichkeiten repräsentieren. Die Erwartungswerte $\bar{\mathbf{c}}_r$ der optimalen Kodebuchvektoren ergeben sich auch hier durch die Zentroidbedingung, die sich nun als

$$\bar{\mathbf{c}}_r = \frac{\langle P(r|\mathbf{x}, \bar{\mathcal{C}}, \sigma) \mathbf{x} \rangle_{\mathcal{X}}}{\langle P(r|\mathbf{x}, \bar{\mathcal{C}}, \sigma) \rangle_{\mathcal{X}}} \quad (2.17)$$

schreiben läßt. Durch Differenzieren von (2.15) kann man leicht verifizieren, daß die nicht-linear gekoppelten Gleichungssysteme (2.16) und (2.17) die Sattelpunktgleichungen für $F(\mathcal{C}, \sigma)$ und damit notwendige Bedingungen für ein Minimum von $F(\mathcal{C}, \sigma)$ darstellen.

Es sei angemerkt, daß die Nebenbedingung $\sum_r v_{kr} = 1$ auch noch für die Erwartungswerte $P(r|\mathbf{x}, \bar{\mathcal{C}}, \sigma) \equiv \bar{v}_{kr}$ gilt [vgl. (2.16)] und sich nun als Normierungsbedingung der Zuordnungswahrscheinlichkeiten

$$\sum_{r=1}^M P(r|\mathbf{x}, \bar{\mathcal{C}}, \sigma) = 1 \quad (2.18)$$

erweist. Faßt man die M Zuordnungswahrscheinlichkeiten $P(r|\mathbf{x}, \bar{\mathcal{C}}, \sigma)$ als Funktionen von \mathbf{x} auf, so definieren sie eine *unscharfe* Partition des Merkmalsraums [Gra95], durch die

jeder Punkt \mathbf{x} mit der Wahrscheinlichkeit $P(r|\mathbf{x}, \bar{\mathcal{C}}, \sigma)$ dem Kodebuchvektor \mathbf{c}_r zugeordnet wird.

Zur Bestimmung des optimalen Kodebuchs schlagen K. Rose et al. nun vor, durch eine Fixpunktiteration von (2.16) und (2.17) das Minimum von $F(\mathcal{C}, \sigma)$ zunächst bei hoher Temperatur $2\sigma^2$ zu berechnen, weil $F(\mathcal{C}, \sigma)$ dort einfach konvex ist und das Minimum dort entsprechend sicher gefunden wird. Anschließend sollte durch Verringerung des Temperaturparameters σ ein Abkühlvorgang modelliert und so ein Kodebuch bestimmt werden, welches das oben aufgeführte Fehlermaß (2.14) näherungsweise minimiert. Die Eigenschaften dieses Abkühlprozesses, der zu einer hierarchisch geordneten Serie von Kodebüchern führt, welche die Strukturen des Datensatzes mit zunehmender Auflösung widerspiegeln, sind von verschiedenen Autoren detailliert untersucht worden [DT94b, DT94a, Der95, RGF90].

Im Besonderen kann dieses RGF-*clustering* auch als sequentieller, stochastischer Algorithmus formuliert werden [Der95]. Dabei wird aus dem Datensatz \mathcal{X} zufällig ein Datenpunkt \mathbf{x}_k gezogen und entsprechend der Lernregel

$$\mathbf{c}_r^{\text{neu}} = \mathbf{c}_r^{\text{alt}} + \epsilon_c \Delta \mathbf{c}_r(\mathbf{x}_k), \quad r = 1 \dots M \quad (2.19)$$

mit

$$\Delta \mathbf{c}_r(\mathbf{x}_k) = P(r|\mathbf{x}_k, \mathcal{C}, \sigma)(\mathbf{x}_k - \mathbf{c}_r^{\text{alt}}) + \boldsymbol{\nu} \quad (2.20)$$

eine Anpassung des Kodebuchs \mathcal{C} vorgenommen. Hierbei ist $\boldsymbol{\nu}$ ein normalverteilter Zufallsvektor mit einer sehr kleinen Varianz und einem verschwindenden Mittelwert [Der95]. Nach jedem Lernschritt (2.20) wird der Temperaturparameter σ geringfügig verkleinert bis – ausgehend von einem Startwert σ_i – ein Endwert σ_f erreicht ist. Im Verlauf dieser iterativen Kodebuchberechnung wird auch der *Lernparameter* ϵ_c von einem großen Wert (≈ 0.9) auf einen kleinen Wert ($\approx M/N$) verringert.

Es sei angemerkt, daß die Lernregeln (2.19) und (2.20) tiefgreifende formale Ähnlichkeiten zu *clustering*-Algorithmen wie dem Kohonen-Algorithmus [Koh82] aufweisen, die aus der Neuroinformatik stammen. Diese in der Dissertation von D. Dersch [Der95] genauer erklärten Zusammenhänge lassen es auch zu, das RGF-*clustering* als Modell für selbstorganisiertes Lernen in biologischen neuronalen Netzwerken zu interpretieren (vgl. [ABK⁺99]).

D. Dersch und P. Tavan haben gezeigt [DT94b, DT94a], daß ein optimales Kodebuch im Grenzfall sehr vieler Kodebuchvektoren ($M \rightarrow \infty$) die Eigenschaft der *load balance*, d.h.

$$\langle P(r|\mathbf{x}, \mathcal{C}, \sigma) \rangle_{\mathcal{X}} = \frac{1}{M}, \quad r = 1 \dots M \quad (2.21)$$

und der Schwerpunkterhaltung, d.h.

$$\frac{1}{M} \sum_{r=1}^M \mathbf{c}_r = \langle \mathbf{x} \rangle_{\mathcal{X}} \quad (2.22)$$

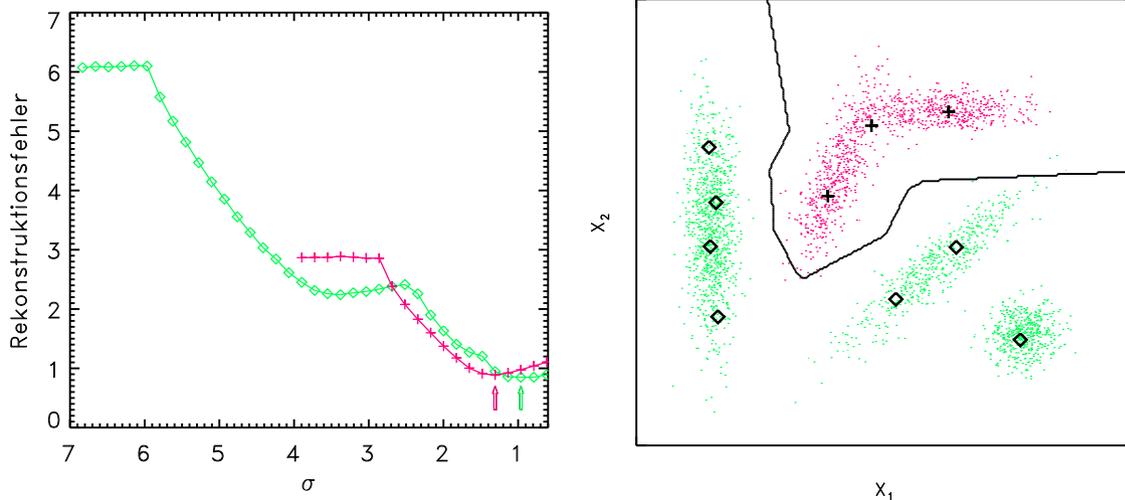


Abbildung 2.9.: **RGF-clustering und nächster Nachbar Erkenner.** Für die Konstruktion eines nächsten Nachbar Erkenners wurden die Datenpunkte der Klassen $\lambda(\mathbf{x}) = 1$ und $\lambda(\mathbf{x}) = 2$ getrennt einem RGF-clustering mit jeweils $M_{\lambda=1} = 7$ und $M_{\lambda=2} = 3$ Kodebuchvektoren unterzogen. Links wird für jede der beiden Klassen der Verlauf des Rekonstruktionsfehlers $R(\mathcal{C}, \mathcal{X})$ während des Abkühlprozesses als Funktion des Temperaturparameters σ dargestellt. Der Temperaturparameter σ_f bei minimalem $R(\mathcal{C}, \mathcal{X})$ ist jeweils als Pfeil eingetragen. Rechts werden die beiden bei minimalem $R(\mathcal{C}, \mathcal{X})$ erhaltenen Kodebücher \mathcal{C}_1 und \mathcal{C}_2 (\diamond und $+$) im Datensatz \mathcal{X} gezeigt. Die aus (2.13) resultierende Klassengrenze ist als schwarze Linie eingetragen.

hat. Die Eigenschaft der *load balance* besagt dabei, daß jeder Kodebuchvektor \mathbf{c}_r die gleiche Anzahl von Datenpunkten repräsentiert. D. Dersch entwickelte auf Grundlage der beiden Eigenschaften (2.21) und (2.22) einen Algorithmus, der diese Eigenschaften als Nebenbedingung einführt und so zu besseren Kodebüchern führt [Der95].

Das Ergebnis des RGF-clustering ist unabhängig vom gewählten Startkodebuch, wenn $\sigma_i > \sigma_{\mathcal{X}}$ gewählt wird, wobei $\sigma_{\mathcal{X}}$ die Standardabweichung des Datensatzes \mathcal{X} ist [RGF90]. Für die Wahl des Endwerts σ_f gibt es aber keine ähnlich verbindliche Vorschrift. D. Dersch führte als Abbruchkriterium das Minimum des *Rekonstruktionsfehlers* $R(\mathcal{C}, \mathcal{X}) = \langle |\mathbf{x} - \sum_{r=1}^M P(r|\mathbf{x}, \mathcal{C}, \sigma) \mathbf{c}_r| \rangle_{\mathcal{X}}$ ein. Allerdings konnte dieses Kriterium nur heuristisch begründet werden [Der95]. In Abschnitt 2.3.3 wird dagegen ein theoretisch fundiertes Abbruchkriterium eingeführt werden, das auf der noch zu erklärenden Äquivalenz des RGF-clustering mit Verfahren der Dichteschätzung durch Mischungen univariater Normalverteilungen beruht.

Zur Illustration der Ergebnisse des RGF-clustering zeigt Abbildung 2.9 einen nächsten Nachbar Erkenner, der mittels des RGF-clustering berechnet wurde. Auch wenn der so resultierende Klassifikator zu guten Ergebnissen führt (siehe auch [Alb95, Gro95]), so ist

ein solcher nächster Nachbar Erkennen aus wahrscheinlichkeitstheoretischer Sicht dennoch wenig befriedigend.

Bayesscher Erkennen

Bei der Behandlung der Klassifikation im Rahmen der Wahrscheinlichkeitstheorie [Gar97] wird das Ziel verfolgt, einen unbekanntem Merkmalsvektor \mathbf{x} so zu klassifizieren, daß das Risiko einer Fehlklassifikation minimiert wird [Bis95]. Für einen Merkmalsvektor \mathbf{x} bezeichne $P(\lambda|\mathbf{x})$ die Wahrscheinlichkeit, daß die Klasse λ vorliegt. Ist diese *a posteriori*-Wahrscheinlichkeit $P(\lambda|\mathbf{x})$ bekannt, dann kann durch die Klassifikationsvorschrift

$$\lambda(\mathbf{x}) = \arg \max_{\lambda'=1\dots K} P(\lambda'|\mathbf{x}) \quad (2.23)$$

eine Minimierung der Fehlklassifikationen erreicht werden [Rip96].

Für die Konstruktion eines solchen risikominimierenden Erkenners müssen die *a posteriori*-Wahrscheinlichkeiten $P(\lambda|\mathbf{x})$ aus den Merkmalsvektoren \mathcal{X} und den Klassenzugehörigkeiten \mathcal{Y} eines Trainingsdatensatzes \mathcal{Z} bestimmt werden. Mit Hilfe des aus der Wahrscheinlichkeitstheorie bekannten Bayesschen Satzes [Rip96]

$$P(\lambda|\mathbf{x}) = \frac{P_\lambda p(\mathbf{x}|\lambda)}{p(\mathbf{x})} = \frac{P_\lambda p(\mathbf{x}|\lambda)}{\sum_{\lambda'}^K P_{\lambda'} p(\mathbf{x}|\lambda')} \quad (2.24)$$

läßt sich $P(\lambda|\mathbf{x})$ auf Größen zurückführen, die durch die Analyse von \mathcal{X} und \mathcal{Y} geschätzt werden können. Dabei sind P_λ die *a priori*-Wahrscheinlichkeiten der K Klassen, die einfach aus den Mächtigkeiten der verschiedenen Klassen in \mathcal{Z} bestimmt werden können. Die Dichte $p(\mathbf{x})$ beschreibt die Verteilung aller Merkmalsvektoren in \mathcal{X} und kann durch die Summe über alle *a priori*-Wahrscheinlichkeiten, multipliziert mit den *klassenlokalen* Dichten $p(\mathbf{x}|\lambda)$, berechnet werden.

Letztendlich ist also für eine Klassifikation nach (2.23) und (2.24) die näherungsweise Bestimmung von P_λ und $p(\mathbf{x}|\lambda)$ notwendig. Ein solcher Klassifikator wird dann auch als Bayesscher Erkennen bezeichnet. Abbildung 2.10 illustriert das Vorgehen an Hand eines eindimensionalen Datensatzes. In diesem Beispiel wird auch deutlich, daß die *a posteriori*-Wahrscheinlichkeiten $P(\lambda|\mathbf{x})$ eine *unscharfe* Partitionierung des Merkmalsraums darstellen, da für jedes \mathbf{x} die Normierungsbedingung

$$\sum_{\lambda=1}^K P(\lambda|\mathbf{x}) = 1 \quad (2.25)$$

gilt.

Im Bayesschen Satz (2.24) beschreibt $p(\mathbf{x}|\lambda)$ die Verteilung der Merkmalsvektoren der Klasse λ . Da diese Dichtefunktionen üblicherweise unbekannt sind, müssen sie für die

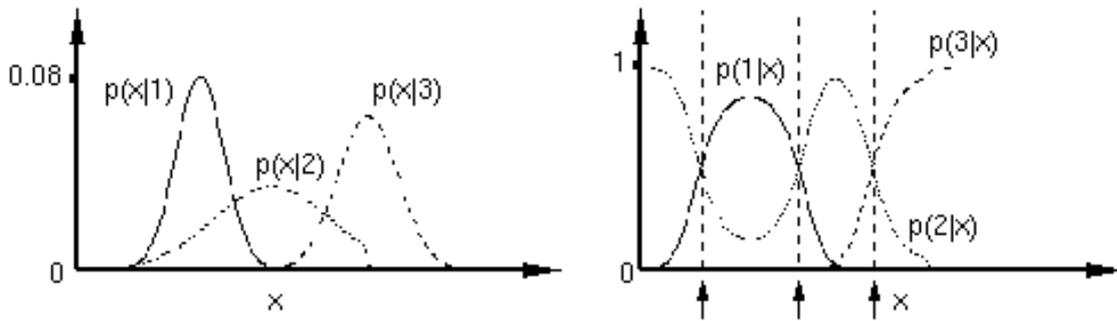


Abbildung 2.10.: **Bayesscher Satz.** Links sind schematisch klassenlokale Dichten $p(x|1)$, $p(x|2)$ und $p(x|3)$ eines eindimensionalen Datensatzes zu erkennen, der aus drei Klassen besteht. Die *a priori*-Wahrscheinlichkeiten der Klassen sind $P_1 = P_2 = 1$ und $P_3 = 3$. Rechts sind die *a posteriori*-Wahrscheinlichkeiten $P(1|x)$, $P(2|x)$ und $P(3|x)$ nach Anwendung des Bayesschen Satzes (2.24) dargestellt. Die Pfeile kennzeichnen die Klassifikationsgrenzen, die sich aus der Gleichheit (z.B. $P(2|x) = P(3|x)$) der *a posteriori*-Wahrscheinlichkeiten ergeben.

Konstruktion eines Bayesschen Mustererkenners zunächst aus dem Datensatz \mathcal{Z} geschätzt werden. Dieses als Dichteschätzung bekannte Problem wird im 3. Kapitel eine herausragende Rolle spielen. Im nächsten Abschnitt werden daher zwei verschiedene Verfahren zur Dichteschätzung diskutiert, um die Schwierigkeiten und mögliche Herangehensweisen an diese Aufgabenstellung zu illustrieren.

2.3.3. Dichteschätzung

In der klassischen Statistik ist die *Dichteschätzung* eines gegebenen Datensatzes $\mathcal{X} = \{\mathbf{x}_k | k = 1 \dots N\}$ ein bekanntes Problem [DH73]. Die Aufgabe besteht darin, die Wahrscheinlichkeitsdichte $p(\mathbf{x})$, mit der Daten durch einen bestimmten Prozeß erzeugt wurden, durch eine Funktion $\hat{p}(\mathbf{x}|\Theta)$ zu schätzen. Alle dazu notwendigen Parameter werden dabei mit Θ bezeichnet. Die Dichteschätzung eines Datensatzes ist in der Statistik häufig von großem Nutzen. So ist, wie oben verdeutlicht wurde, eine gute Dichteschätzung der klassenlokalen Datenverteilungen unabdingbar für die Konstruktion eines guten Bayesschen Mustererkenners.

Die einfachste Methode einer Dichteschätzung ist die Histogrammnäherung [DH73], welche in Abbildung 2.11 veranschaulicht wird. Dabei wird zunächst für jede Achse $i = 1 \dots D$ des Eingaberaums \mathbb{R}^D ein Intervall $\mathcal{J}_i = [x_i^a, x_i^b]$ festgelegt und dieses anschließend in N_i^z Teilintervalle $\mathcal{J}_i^{z_i}$, $z_i = 1 \dots N_i^z$, $i = 1 \dots D$, aufgeteilt, so daß $N^z = \prod_i^D N_i^z$ Zellen entstehen. Für die Histogrammnäherung wird nun abgezählt, wie viele Datenpunkte aus \mathcal{X} innerhalb der so definierten Volumina $\mathcal{J}^{z_1 \dots z_D} \equiv \mathcal{J}_1^{z_1} \times \mathcal{J}_2^{z_2} \times \dots \times \mathcal{J}_D^{z_D}$ liegen. Diese Kardinalzahlen $M^{z_1 \dots z_D}$ sind ein Maß für die Wahrscheinlichkeit einen Datenpunkt inner-

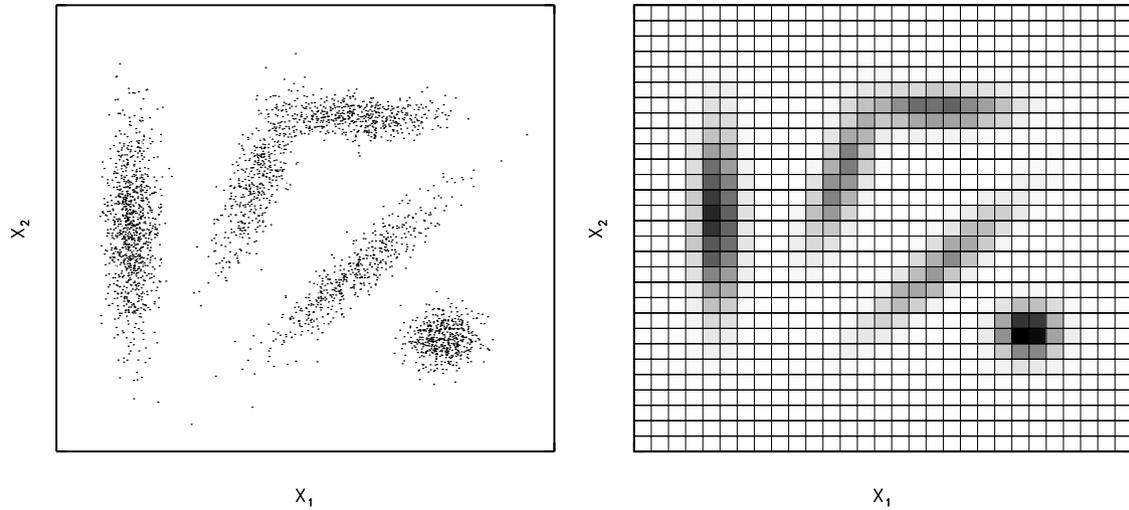


Abbildung 2.11.: **Datenverteilung und Histogrammnäherung.** Links ist nochmals die zweidimensionale Datenverteilung \mathcal{X} dargestellt. Rechts ist das Ergebnis einer Dichteschätzung mit der Histogrammnäherung zu sehen. Der zweidimensionale Eingaberaum wurde dafür in 30×30 Zellen zerlegt. Die Grautönung ist ein Maß für die Anzahl von Datenpunkten in einer Zelle.

halb von $\mathcal{J}^{z_1 \dots z_D}$ zu finden. Die Dichteschätzung erfolgt schließlich über die Normierung der $M^{z_1 \dots z_D}$ mit der Gesamtanzahl N der Datenpunkte:

$$\hat{p}(\mathbf{x}|\Theta) = \frac{M^{z_1 \dots z_D}}{N} \quad \text{mit } \mathbf{x} \in \mathcal{J}^{z_1 \dots z_D}. \quad (2.26)$$

Die Parameter Θ einer solchen Histogrammnäherung umfassen die Kardinalzahlen $M^{z_1 \dots z_D}$ und die Definition der Zellen $\mathcal{J}^{z_1 \dots z_D}$. Eine solche Dichteschätzung ist in Abbildung 2.11 durch den Grauton der 30×30 Zellen dargestellt.

Dieses Verfahren zur Dichteschätzung ist zwar einfach in der algorithmischen Ausführung, aber nur für niedrigdimensionale Datensätze \mathcal{X} geeignet. Da die Zellenzahl N^z exponentiell mit der Dimension D anwächst, ist sehr schnell die Grenze des Machbaren erreicht⁵. Eine große Zellenzahl N^z erfordert einen sehr umfangreichen Datensatz \mathcal{X} , um eine einigermaßen aussagekräftige Dichteschätzung zu erhalten. Ist der Datensatz \mathcal{X} kleiner als die Zellenzahl N^z , so wird in viele Zellen $\mathcal{J}^{z_1 \dots z_D}$ kein Datenpunkt eingezählt. Damit werden die Schätzungen sehr unzuverlässig und die statistischen Fehler entsprechend groß. Die hier angesprochenen Probleme werden in der englischsprachigen Literatur häufig als *curse of dimensionality* bezeichnet [Bis95].

Von größerer Bedeutung für die Dichteschätzung hochdimensionaler Datensätze sind andere parametrische Verfahren. Eine Möglichkeit ist die Verwendung von Mischungen mul-

⁵Wird ein 20-dimensionaler Eingaberaum in jeweils 3 Zellen pro Raumachse aufgeteilt, so sind schon $3^{10} \approx 3.5 \times 10^9$ Zellen notwendig.

tivariater Normalverteilungen. Dichteschätzungen vermittels solcher Mischungen werden entscheidend in das von mir weiterentwickelte RBF-Netzwerk zur Funktionenapproximation eingehen (siehe Kapitel 3). Die beiden folgenden Abschnitte stellen daher zwei etablierte Verfahren zur Dichteschätzung mit Hilfe *einer* einzigen und mit Hilfe einer Mischung von multivariaten Normalverteilungen dar.

Dichteschätzung mit einer multivariaten Normalverteilung

Zunächst wird die Dichteschätzung mit einer multivariaten Normalverteilung

$$\hat{p}(\mathbf{x}|\Theta^{PCA}) = \frac{\exp[-(\mathbf{x} - \mathbf{c})^T \Sigma^{-1}(\mathbf{x} - \mathbf{c})]}{(2\pi)^{D/2}(\det \Sigma)^{1/2}} \quad (2.27)$$

betrachtet [DH73]. Die positive Funktion (2.27) hat am Punkt $\mathbf{c} \in \mathbb{R}^D$ ihr Maximum. Die Ausdehnung von $\hat{p}(\mathbf{x}|\Theta^{PCA})$ in die verschiedenen Raumrichtungen wird über die symmetrische Matrix $\Sigma \in \mathbb{R}^{D \times D}$ skaliert. Auf Grund der Normierung ist $\hat{p}(\mathbf{x}|\Theta^{PCA})$ eine Wahrscheinlichkeitsdichte.

Wie können nun die Parameter $\Theta^{PCA} = \{\Sigma, \mathbf{c}\}$ so aus den Trainingsdaten \mathcal{X} bestimmt werden, daß $\hat{p}(\mathbf{x}|\Theta^{PCA})$ möglichst gut die Verteilung von \mathcal{X} abschätzt? Eine mögliche Antwort auf diese Frage liefert der *maximum likelihood*-(ML)-Ansatz. Nach diesem Ansatz gibt der Wert $\hat{p}(\mathbf{x}_k|\Theta^{PCA})$ gerade die *likelihood* an, daß \mathbf{x}_k durch die Modelldichte $\hat{p}(\mathbf{x}|\Theta^{PCA})$ erzeugt wurde. Entsprechend ist das Produkt

$$p(\mathcal{X}|\Theta^{PCA}) = \prod_{k=1}^N \hat{p}(\mathbf{x}_k|\Theta^{PCA}) \quad (2.28)$$

die *likelihood*, daß der gesamte Datensatz \mathcal{X} von der Modelldichte $p(\mathcal{X}|\Theta^{PCA})$ erzeugt wurde. Jener Parametersatz $\bar{\Theta}^{PCA}$, für den $p(\mathcal{X}|\Theta^{PCA})$ maximal ist, kennzeichnet also die optimale zu \mathcal{X} passende Modelldichte. Aus analytischen Gründen wird häufig auch die *log-likelihood*

$$l(\mathcal{X}|\Theta^{PCA}) = \ln p(\mathcal{X}|\Theta^{PCA}) = \sum_{k=1}^N \ln \hat{p}(\mathbf{x}_k|\Theta^{PCA}) \quad (2.29)$$

verwendet. Da der Logarithmus eine monoton steigende Funktion ist, maximiert $\bar{\Theta}^{PCA}$ auch $l(\mathcal{X}|\Theta^{PCA})$. Am Maximum der *log-likelihood* müssen die Sattelpunktbedingungen

$$\frac{\partial}{\partial \theta} l(\mathcal{X}|\Theta^{PCA}) = 0 \quad \forall \theta \in \Theta^{PCA} \quad (2.30)$$

erfüllt sein. Werden diese Bedingungen auf die multivariate Schätzung (2.27) angewendet, erhält man mit (2.6) für die Parameter Θ^{PCA} die Bestimmungsgleichungen

$$\mathbf{c} = \langle \mathbf{x} \rangle_{\mathcal{X}} \quad (2.31)$$

$$\Sigma = \langle (\mathbf{x} - \mathbf{c})(\mathbf{x} - \mathbf{c})^T \rangle_{\mathcal{X}}. \quad (2.32)$$

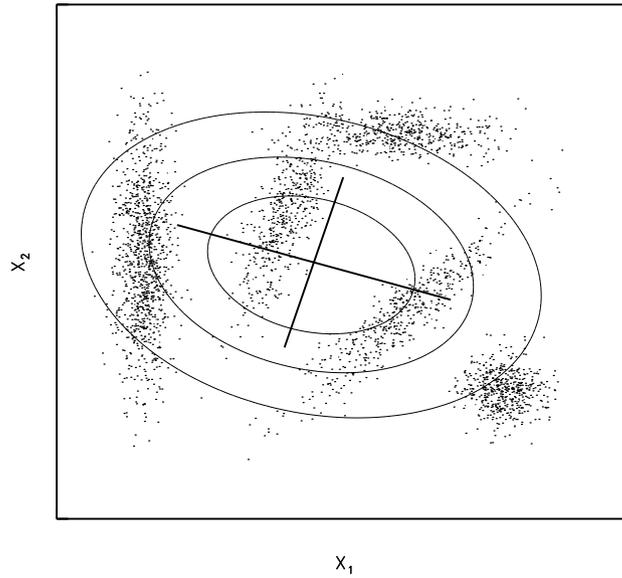


Abbildung 2.12.: *principal component analysis*. In der Abbildung ist das Ergebnis einer PCA des exemplarischen Datensatzes aus Abbildung 2.11 zu sehen. Die beiden Hauptachsen w_i sind als Koordinatenkreuz eingezeichnet, welches auf dem Mittelwert \mathbf{c} der Verteilung zentriert ist. Die Standardabweichungen σ_i sind durch die Längen der Achsen angegeben. Die Form der zweidimensionalen Normalverteilung $\hat{p}(\mathbf{x}|\Theta^{PCA})$ wird durch die Höhenlinien angedeutet.

Der optimale Punkt \mathbf{c} ist gerade der Mittelwert von \mathcal{X} und die optimale metrische Matrix Σ ist die Kovarianzmatrix der Daten \mathcal{X} [DH73]. Die ML-Schätzung der Daten \mathcal{X} mit der Dichte (2.27) führt also zu den Parametern einer Statistik 2. Ordnung des Datensatzes \mathcal{X} [Gar97].

Die Kovarianzmatrix Σ kann in ihre Eigenvektoren \mathbf{w}_i und in die Eigenwerte σ_i^2 , $i = 1 \dots D$, zerlegt werden. Mit der orthogonalen Matrix $\mathbf{W} = (\mathbf{w}_1 \dots \mathbf{w}_D)$ und der diagonalen Matrix $\hat{\Sigma}$ ($\hat{\Sigma}_{ii} = \sigma_i^2$) wird die Spektralzerlegung

$$\Sigma = \mathbf{W} \hat{\Sigma} \mathbf{W}^T \quad (2.33)$$

ermöglicht [MKB94]. Ebenso kann die Inverse Σ^{-1} durch die inverse Diagonalmatrix $\hat{\Sigma}^{-1}$ dargestellt werden,

$$\Sigma^{-1} = \mathbf{W} \hat{\Sigma}^{-1} \mathbf{W}^T, \quad (2.34)$$

und es gilt $\det \Sigma = \det \hat{\Sigma}$. Die Parameter der Schätzung (2.27) können also auch als $\Theta^{PCA} = \{\hat{\Sigma}, \mathbf{W}, \mathbf{c}\}$ geschrieben werden. Diese Beschreibung eines Datensatzes \mathcal{X} durch eine multivariate Normalverteilung wird üblicherweise als *Hauptachsenanalyse* oder *principal component analysis* (PCA) bezeichnet [MKB94].

In Abbildung 2.12 ist die PCA der Beispielverteilung aus Abbildung 2.11 dargestellt. Erwartungsgemäß ist die ML-Dichteschätzung auf Grund des eingeschränkten Modellansatzes von schlechter Qualität. Abbildung 2.13 zeigt einen Bayesschen Erkennen, der auf

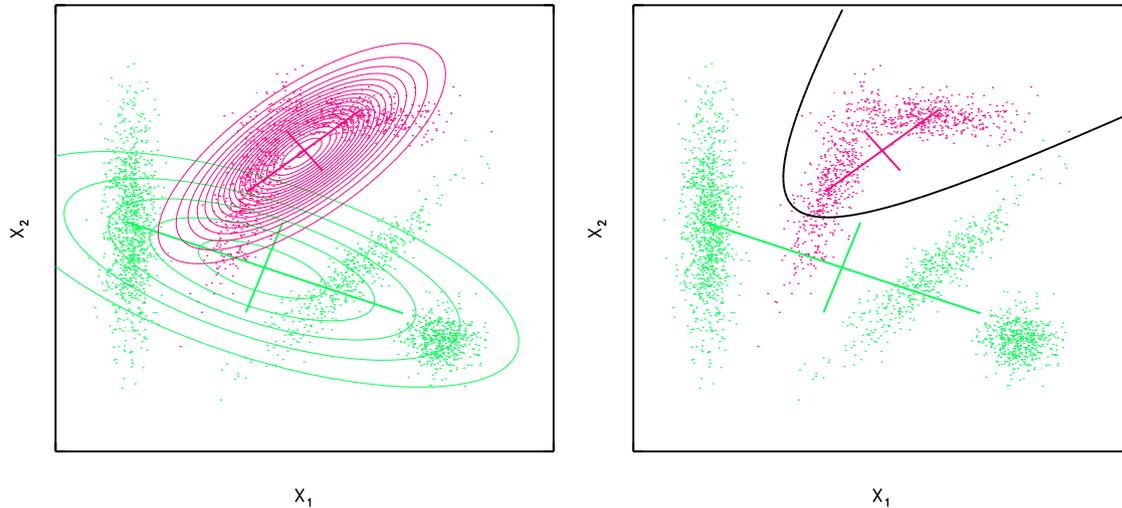


Abbildung 2.13.: **Klassenlokale ML-Dichteschätzung und Bayesscher Erkener.** Links sind die klassenlokalen ML-Dichteschätzungen $\hat{p}(\mathbf{x}|\lambda = 1)$ und $\hat{p}(\mathbf{x}|\lambda = 2)$ gezeigt, die sich mit je einer bivariaten Normalverteilung für die beiden Klassen des Datensatzes aus Abbildung 2.8 ergeben. Aus diesen kann zusammen mit den *a priori*-Wahrscheinlichkeiten $P_{\lambda=1} = 2/3$ und $P_{\lambda=2} = 2/3$ nach dem Bayesschen Satz (2.24) die *a posteriori*-Wahrscheinlichkeit $\hat{P}(1|\mathbf{x})$ bzw. $\hat{P}(2|\mathbf{x})$ bestimmt werden. Die durch $\hat{P}(1|\mathbf{x}) = \hat{P}(2|\mathbf{x})$ resultierende Klassengrenze ist im rechten Teilbild eingetragen.

klassenlokalen ML-Dichteschätzungen $\hat{p}(\mathbf{x}|\lambda = 1)$ und $\hat{p}(\mathbf{x}|\lambda = 2)$ vermittelt jeweils einer bivariaten Normalverteilung durch eine PCA beruht. Trotz der sehr geringen Parameterzahl, erhält man durch diesen einfachen Erkener schon eine recht gute Abgrenzung zwischen den beiden Klassen. Eine deutliche Verbesserung ist zu erwarten, wenn für die ML-Dichteschätzung mit einer Mischung multivariater Normalverteilungen gearbeitet wird. Dies wird im nächsten Abschnitt diskutiert.

Dichteschätzung mit einer Mischung multivariater Normalverteilungen

Eine Verallgemeinerung der Modelldichte (2.27) besteht darin, eine unbekannte Dichte $p(\mathbf{x})$ mit einer Mischung

$$\hat{p}(\mathbf{x}|\Theta^M) = \sum_{r=1}^M P_r \hat{p}(\mathbf{x}|r, \Theta_r^{PCA}) \quad (2.35)$$

von M multivariaten Normalverteilungen mit

$$\hat{p}(\mathbf{x}|r, \Theta_r^{PCA}) = \frac{\exp[-(\mathbf{x} - \mathbf{c}_r)^T \boldsymbol{\Sigma}_r^{-1} (\mathbf{x} - \mathbf{c}_r)/2]}{(2\pi)^{D/2} (\det \boldsymbol{\Sigma}_r)^{1/2}} \quad (2.36)$$

zu schätzen [DH73]. Die P_r sind positive Zahlen und stellen die statistischen Gewichte der Komponenten $\hat{p}(\mathbf{x}|\Theta_r^{PCA})$ dar, d.h. $\sum_r P_r = 1$. Die P_r und $\Theta_r^{PCA} = \{\hat{\Sigma}_r, \mathbf{W}_r, \mathbf{c}_r\}$ werden zu einem Gesamtparametersatz $\Theta^M = \{P_r, \Theta_r^{PCA} | r = 1 \dots M\}$ zusammengefaßt.

Bildet man die *log-likelihood* für die Schätzung (2.35) und führt die Ableitungen nach den verschiedenen Parametern aus, so erhält man für den Parametersatz Θ^M die Sattelpunktbedingungen [Klo96]

$$\mathbf{c}_r = \langle \mathbf{x} \rangle_{r, \Theta^M} \quad (2.37)$$

$$P_r = \langle \hat{p}(r|\mathbf{x}, \Theta^M) \rangle_{\mathcal{X}} \quad (2.38)$$

$$\Sigma_r = \mathbf{C}_{r, \Theta^M} \quad (2.39)$$

mit

$$\mathbf{C}_{r, \Theta^M} \equiv \langle (\mathbf{x} - \mathbf{c}_r)(\mathbf{x} - \mathbf{c}_r)^T \rangle_{r, \Theta^M}. \quad (2.40)$$

Bei der Bildung der Erwartungswerte $\langle \cdot \rangle_{r, \Theta^M}$ in (2.37) und (2.40) haben nicht alle Datenpunkte $\mathbf{x}_k \in \mathcal{X}$ dasselbe statistische Gewicht. Vielmehr wird die Mittelungsvorschrift

$$\langle g(\mathbf{x}) \rangle_{r, \Theta^M} \equiv \frac{\langle P(r|\mathbf{x}, \Theta^M)g(\mathbf{x}) \rangle_{\mathcal{X}}}{\langle P(r|\mathbf{x}, \Theta^M) \rangle_{\mathcal{X}}} \quad (2.41)$$

verwendet. Dies sind lokale Erwartungswerte, die bezüglich der *a posteriori*-Wahrscheinlichkeiten

$$P(r|\mathbf{x}, \Theta^M) = \frac{P_r \hat{p}(\mathbf{x}|r, \Theta_r^{PCA})}{\sum_{r'} P_{r'} \hat{p}(\mathbf{x}|r', \Theta_{r'}^{PCA})} \quad (2.42)$$

der Zuordnungen von Punkten \mathbf{x} zu lokalen Verteilungen r gebildet werden. Die *a posteriori*-Wahrscheinlichkeiten $P(r|\mathbf{x}, \Theta^M)$ definieren eine unscharfe Partitionierung des Merkmalsraums, da für jedes \mathbf{x}

$$\sum_{r=1}^M P(r|\mathbf{x}, \Theta^M) = 1 \quad (2.43)$$

gilt. Jede Partitionierungsfunktion $P(r|\mathbf{x}, \Theta^M)$ gibt die Wahrscheinlichkeit an, mit der ein Datenpunkt \mathbf{x} der r -ten Komponente der Mischungsdichte (2.38) zuzuordnen ist. In die r -lokalen Mittelwerte $\langle \cdot \rangle_{r, \Theta^M}$ gehen daher nur jene Datenpunkte mit hohem statistischen Gewicht ein, die mit großer Wahrscheinlichkeit der r -ten Komponente zugeordnet werden.

Für $M = 1$, also die Schätzung mit *einer* multivariaten Normalverteilung, reduzieren sich (2.37) – (2.39) auf die Gleichungen (2.31) und (2.32) der Statistik 2. Ordnung. In diesem Fall werden die Mittelungen über den gesamten Datensatz \mathcal{X} ausgeführt und die Parameter lassen sich mit Hilfe einer PCA direkt berechnen. Im Unterschied dazu handelt es sich für $M > 1$ bei (2.37), (2.38) und (2.39) um implizite, nichtlinear gekoppelte Bestimmungsgleichungen für den Parametersatz Θ^M . Zusammenfassend kann man also feststellen: Im

Rahmen des ML-Ansatzes ist das Problem der Dichteschätzung mit einer Mischung multivariater Normalverteilungen, jene Lösung von (2.37) – (2.39) zu finden, für die die *log-likelihood*

$$l(\mathcal{X}|\Theta^M) = \sum_{k=1}^N \ln \hat{p}(\mathbf{x}_k|\Theta^M) \quad (2.44)$$

maximal ist.

Die einfachste Möglichkeit wäre die Anwendung eines iterativen Gradientenaufstiegs zur Maximierung von $l(\mathcal{X}|\Theta^M)$. Allerdings ist $l(\mathcal{X}|\Theta^M)$ im allgemeinen keine konvexe Funktion, so daß ein solcher Gradientenaufstieg sehr leicht zu einem lokalen Maximum konvergiert und damit nicht zur optimalen Lösung von (2.37) – (2.39) führt [DH73].

Eine weitere Möglichkeit zur Bestimmung der Parameter Θ^M bietet der *expectation-maximization*-(EM)-Algorithmus [DLR77]. Dabei ist zunächst eine initiale Schätzung der Parameter notwendig. Mit dieser werden nun zunächst die Partitionierungsfunktionen (2.42) (E-Schritt) und damit anschließend ein neuer Parametersatz nach (2.37) – (2.39) (M-Schritt) bestimmt. Dieser Vorgang wird iterativ solange fortgesetzt, bis sich eine selbstkonsistente Lösung einstellt.

Allerdings sind die Ergebnisse des EM-Algorithmus im starken Maße von der Anfangsschätzung der Parameter Θ^M abhängig. Ebenso stellen sich häufig singuläre Lösungen ein, indem die Ausdehnung $\det \Sigma_r$ für einige Normalverteilungen sehr klein und für andere sehr groß wird. Der Grund für diese Instabilitäten sind zufällige Fluktuationen der metrischen Matrix Σ_r während des iterativen Ablaufs des EM-Algorithmus [YSU94, KT97, YC98]. Diese Schwankungen können im nächsten M-Schritt zu einer Verringerung des Gewichts P_r führen und damit letztendlich zum Verschwinden einer Komponente $P_r \hat{p}(\mathbf{x}|r, \Theta_r^{PCA})$. Gerade bei großen Dimensionen des Merkmalsraums ist der EM-Algorithmus besonders anfällig für diese Instabilitäten, da dann eine kleine Fluktuation von Σ_r eine große Änderung des Volumens $\det \Sigma_r$ und damit nach (2.38) auch des statistischen Gewichts P_r verursacht. Diese Eigenschaften zeigen die komplexe Struktur der *log-likelihood* im Parameterraum und verweisen auf die Schwierigkeiten eines naiven Gradientenaufstiegs mittels des EM-Algorithmus.

Entscheidende Impulse zur Lösung dieser Schwierigkeiten kamen von M. Kloppenburg und P. Tavan [Klo96, KT97], denen die Äquivalenz des RGF-*clusterings* mit einem Spezialfall der ML-Dichteschätzung durch Mischungen von Normalverteilungen aufgefallen war. Für den Fall univariater Normalverteilungen mit identischen statistischen Gewichten, d.h. $\Sigma_r^{-1} = \mathbf{I}\sigma^{-2}$ und $P_r = 1/M$, stimmt nämlich die ML-Stationaritätsbedingung (2.37) für die \mathbf{c}_r exakt mit der RGF-Sattelpunktbedingung (2.17) überein. Ebenso stimmt der Ausdruck (2.42) für die *a posteriori*-Wahrscheinlichkeit $P(r|\mathbf{x}, \Theta^M)$ mit der Gleichung (2.16) für die Zuordnungswahrscheinlichkeit $P(r|\mathbf{x}, \mathcal{C}, \sigma)$ beim RGF-*clustering* überein. Da auch die *log-likelihood* (2.44) bis auf den Faktor $-2\sigma^2$ mit der freien Energie (2.15) übereinstimmt, entspricht die Maximierung von $l(\mathcal{X}|\Theta^M)$ einer Minimierung von $F(\mathcal{C}, \sigma)$. Daher

ist das RGF-*clustering* tatsächlich ein Spezialfall der ML-Dichteschätzung. Im Gegensatz zur freien Energie $F(\mathcal{C}, \sigma)$ besitzt die *log-likelihood* $l(\mathcal{X}|\Theta)$ jedoch bei einem nichtverschwindenden Wert des Parameters σ ein Extremum und erlaubt so die Bestimmung einer minimalen Temperatur $2\sigma_f^2$ während des Abkühlprozesses beim RGF-*clustering*. Dieser entsprechend modifizierte RGF-Algorithmus zur ML-Dichteschätzung mit Mischungen univariater Normalverteilungen wird künftig auch als *univar*-Verfahren bezeichnet.

Nun scheint die ML-Dichteschätzung mit einer Mischung univariater Normalverteilungen mit identischen Varianzen und statistischen Gewichten zunächst keinen Vorteil zu haben. Aber gerade die Beschränkung des Parametersatzes auf die \mathbf{c}_r und die Verwendung von σ als Abkühlparameter ermöglichen beim *univar*-Algorithmus eine hohe Stabilität bei der Parameterbestimmung. Diese algorithmische Stabilität des *univar*-Verfahrens hat M. Kloppenburg und P. Tavan dazu motiviert, einen verwandten Algorithmus auch für den multivariaten Fall zu entwickeln [KT97]. Dabei werden die Sattelpunktbedingungen (2.38) und (2.39) für die Parameter P_r und Σ_r so erweitert, daß ein Abkühlverfahren möglich wird.

Zunächst werden dazu die Σ_r durch ihre Eigenvektoren \mathbf{w}_{ir} und Eigenwerte σ_{ir}^2 , $i = 1 \dots D$, ersetzt. Die inverse Matrix Σ_r^{-1} kann daher nach einer Spektralzerlegung $\Sigma_r^{-1} = \mathbf{W}_r \hat{\Sigma}_r^{-1} \mathbf{W}_r^T$ in die orthogonale Matrix $\mathbf{W}_r = (\mathbf{w}_{1r} \dots \mathbf{w}_{Dr})$ der Eigenvektoren und die diagonale Matrix $\hat{\Sigma}_r$ der Eigenwerte aufgeteilt werden. Mit dieser Repräsentation von Σ_r^{-1} gliedert sich die Sattelpunktbedingung (2.39) in zwei Gruppen von Gleichungen. Die erste Gruppe

$$\sigma_{ir}^2 = \mathbf{w}_{ir}^T \mathbf{C}_{r, \Theta^M} \mathbf{w}_{ir} \quad (2.45)$$

beschreibt die Varianzen σ_{ir}^2 in Richtung der \mathbf{w}_{ir} und die zweite Gruppe

$$\sigma_{ir}^2 \mathbf{w}_{ir} = \mathbf{C}_{r, \Theta^M} \mathbf{w}_{ir} \quad (2.46)$$

die Eigenvektoren \mathbf{w}_{ir} der Kovarianzmatrizen \mathbf{C}_{r, Θ^M} . M. Kloppenburg und P. Tavan erweitern nun die Sattelpunktbedingungen (2.45) zu

$$\sigma_{ir}^2 = \mathbf{w}_{ir}^T \mathbf{C}_{r, \Theta^M} \mathbf{w}_{ir} + \mu \frac{\sigma^2 - \sigma_{ir}^2}{\langle P(r|\mathbf{x}, \Theta^M) \rangle_{\mathcal{X}}} \quad (2.47)$$

Nun sind die Varianzen σ_{ir}^2 über einen Kopplungsparameter μ an den Temperaturparameter σ gekoppelt. Im Fall einer starken Kopplung ($\mu \rightarrow \infty$) erhält man daher den *univar*-Algorithmus. Bei der Anwendung des EM-Algorithmus werden nun im M-Schritt die Gleichungen (2.46) und (2.47) an Stelle von (2.39) ausgewertet. Zusätzlich ist es nach jedem M-Schritt notwendig, die Vektoren \mathbf{w}_{ir} einer hierarchischen Orthonormierung, z.B. mit dem Schmidtschen Verfahren [Fis86], zu unterziehen. Als Folge wird \mathbf{w}_{1r} zum Eigenvektor von \mathbf{C}_{r, Θ^M} mit dem größten Eigenwert σ_{1r}^2 , \mathbf{w}_{2r} zum Eigenvektor mit dem zweitgrößten σ_{2r}^2 usw., konvergieren [RT89].

Schließlich kann auch die EM-Gleichung (2.38) durch

$$P_r = \langle P(r|\mathbf{x}, \Theta^M) \rangle_{\mathcal{X}} + \lambda \left(\frac{1}{M} - P_r \right) \quad (2.48)$$

ersetzt werden. Durch den Zusatzterm werden für einen großen Kopplungsparameter λ die statistischen Gewichte annähernd auf denselben Wert $1/M$ gebracht und folglich auch die Eigenschaft der *load balance* erzwungen [vgl. (2.21)].

Für den Abkühlprozeß während der EM-Schritte mit (2.37) und (2.46) – (2.48) wird der Temperaturparameter σ langsam von einem großen zu einem kleinen Wert verringert. Währenddessen oder anschließend werden auch die Kopplungsparameter μ und λ verkleinert [Klo96]. Der Verlauf der Optimierung kann durch die *log-likelihood* $l(\mathcal{X}|\Theta^M, \sigma, \mu, \lambda)$ beurteilt und bei Erreichen eines Maximums von $l(\mathcal{X}|\Theta^M, \sigma, \mu, \lambda)$ beendet werden.

M. Kloppenburg und P. Tavan haben somit einen EM-Algorithmus entwickelt, der unter Verwendung eines Abkühlverfahrens sicher konvergiert, ein klar definiertes Abbruchkriterium hat und zuverlässig zu einer guten Dichteschätzung führt. Dieser Algorithmus ist inzwischen erfolgreich für die Klassifikation von Sprachdaten verwendet worden [Met98].

2.4. Künstliche neuronale Netze

Die im vorangehenden Abschnitt diskutierten Verfahren zur Funktionenapproximation, Mustererkennung und Dichteschätzung basierten hauptsächlich auf mathematischen Vorstellungen. Beim Entwurf einiger dieser Methoden, wie etwa bei den zum RGF-*clustering* verwendeten Abkühlungsmethoden [RGF93, DT94a], standen allerdings auch Konzepte aus der statistischen Physik Pate. Ferner wurden auch schon neurobiologisch motivierte Methoden erwähnt. Der letztere Zugang kennzeichnet das Forschungsgebiet der Neuroinformatik, das systematisch versucht, aus der Funktionsweise biologischer Informationsverarbeitung algorithmische Vorgehensweisen zur Lösung schwieriger mathematischer Probleme zu entwickeln. In diesem Abschnitt sollen einige Konzepte der Neuroinformatik vorgestellt werden, die für das Verständnis von RBF-Netzwerken notwendig sind. Am Ende dieses Kapitels werden dann schließlich alle Begriffe zur Verfügung stehen, die zur Diskussion solcher RBF-Netzwerke nötig sind.

In Abschnitt 2.2 wurden eine Reihe wichtiger Strukturen des Gehirns und der Kleinhirnrinde skizziert. Es wurde dort deutlich, daß die Nervenzelle eine zentrale Rolle bei der Informationsverarbeitung im Gehirn einnimmt. Bei der Konstruktion künstlicher neuronaler Netzwerke (kNN) spiegelt sich dieses Konzept in der Einführung des *formalen Neurons* wieder. Abbildung 2.14 zeigt ein einzelnes formales Neuron innerhalb eines Netzwerks von M formalen Neuronen. Das formale Neuron ist die kleinste informationsverarbeitende Einheit eines kNN. Es erhält über die Verbindungen mit anderen Neuronen⁶ Eingaben, die innerhalb des Neurons verarbeitet, vom ihm ausgegeben und an andere Neuronen weitergeleitet werden.

⁶Zur genaueren Unterscheidung werden künftig biologische Neuronen als *Nervenzellen* und formale Neuronen als *Neuronen* bezeichnet.

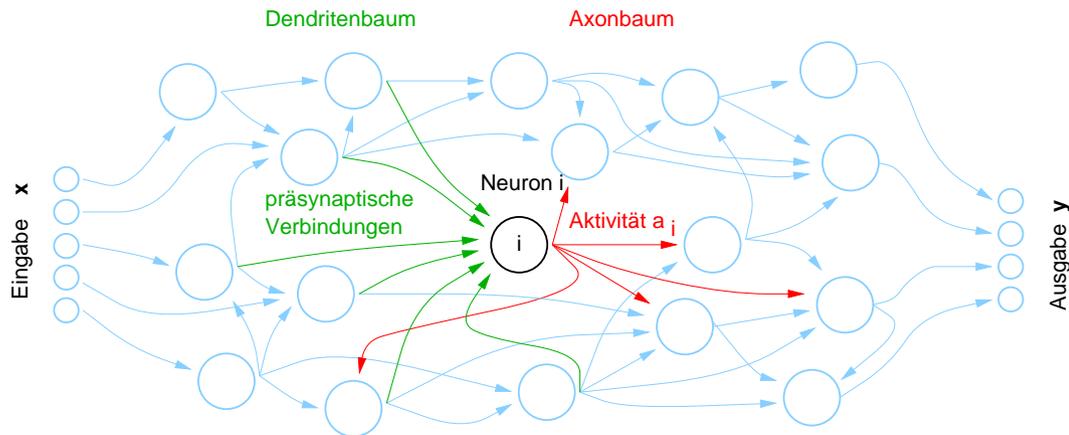


Abbildung 2.14.: **Künstliches neuronales Netzwerk.** Elementarer Baustein eines künstlichen neuronalen Netzwerks ist das formale Neuron. Dieses erhält über die präsynaptischen Verbindungen Informationen von anderen formalen Neuronen. Diese Eingaben werden verrechnet und als Aktivität weitergegeben.

Die meisten Modelle der Neuroinformatik beschreiben dabei die elektrische Aktivität einer Nervenzelle i durch eine reellwertige *Aktivitätsvariable* a_i , $i = 1 \dots M$. Die Aktivitäten aller Neuronen eines kNN können zu einem Vektor $\mathbf{a} = (a_1 \dots a_M)$ zusammengefaßt werden, der damit den Aktivitätszustand dieses Netzes erfährt. Analog zu einem natürlichen Netzwerk von Nervenzellen wird jede Aktivität a_i über den *Axonbaum* zu nachgeschalteten Neuronen weitergeleitet (vgl. die in Abbildung 2.14 rot eingezeichneten Verbindungen). In biologischem Nervengewebe bildet eine Nervenzelle über ihren Axonbaum sehr viele synaptische Kontakte mit anderen Zellen aus. In einem kNN wird die Übertragungsqualität eines solchen Kontakts von einem Neuron i zu einem Neuron j durch eine reellwertige Variable w_{ji} modelliert. Diese sog. *Verbindungsstärken* w_{ji} geben an, wie stark eine neuronale Aktivität a_i die Aktivität eines Neurons j beeinflusst. Alle zu einem Neuron i konvergierenden Verbindungsstärken können in einem Vektor $\mathbf{w}_i = (w_{1i} \dots w_{Mi})^T$ gebündelt werden. Dieser sog. *Dendritenbaum* ist in Abbildung 2.14 grün eingezeichnet. Ebenso können die Stärken der von einem Neuron i auslaufenden Verbindungen zu einem Vektor $\mathbf{f}_j = (w_{j1} \dots w_{jM})$ zusammengefaßt werden, der die Übertragungseigenschaften des Axonbaums modelliert. Verbindungsstärken w_{ji} mit einem Wert von Null kennzeichnen dabei eine nicht vorhandene Verbindung von Neuron i zu einem Neuron j .

Die Zellaktivitäten innerhalb biologischen Nervengewebes sind durch dynamische Musterbildungsprozesse geprägt [Hin94, ST85]. Die Modellierung dieser dynamischen Arbeitsweise in einem Netzwerk formaler Neuronen kann beispielsweise durch eine zeitliche Diskretisierung der folgenden Art vorgenommen werden: Ein Neuron erhält zum Zeitpunkt⁷ t die über den Dendritenbaum \mathbf{w}_i modulierte Aktivität $\mathbf{a}[t - 1]$ als Eingabe. Die Aktivität $a_i[t]$ des Neurons i wird dabei durch eine, im allgemeinen nichtlineare, Übertragungsfunk-

⁷Diskretisierte Zeitangaben werden im folgenden durch eckige Klammern $[\cdot]$ gekennzeichnet.

tion $g_i : \mathbb{R}^M \rightarrow \mathbb{R}$ bestimmt. Die Dynamik eines kNN ist im Rahmen dieses Modells vollständig durch

$$a_i[t] = g_i(\mathbf{a}[t-1]; \Theta_i), \quad i = 1 \dots M, \quad (2.49)$$

beschrieben. Die Parametermenge Θ_i umfaßt den Dendritenbaum \mathbf{w}_i und sonstige für die Beschreibung der Übertragungsfunktion notwendigen Parameter.

Für die Klasse der rekurrenten kNN sind die dynamischen Eigenschaften, die aus (2.49) folgen, von entscheidender Bedeutung [Hop82, TKG90]. Netzwerke dieses Typs sind so strukturiert, daß auch Rekursionen innerhalb des Netzwerks auftreten können. Das in Abbildung 2.14 skizzierte Netzwerk weist einige solche Verbindungen auf. Im folgenden werden aber hauptsächlich *vorwärtsgerichtete* Netzwerke betrachten, bei denen keine rücklaufenden Verbindungen auftreten. Wird einem solchen *feedforward* Netzwerk über ausgezeichnete Neuronen eine Eingabeaktivität $\mathbf{x} \in \mathbb{R}^D$ zugeführt, so wird sich an den speziellen Ausgabeneuronen nach einer endlichen Zeit eine Ausgabeaktivität $\mathbf{y} \in \mathbb{R}^K$ einstellen. Ein solches Netzwerk führt daher eine Abbildung

$$\mathbf{y} = \hat{\mathbf{f}}(\mathbf{x}; \Theta) \quad (2.50)$$

aus. $\Theta = \{\Theta_i | i = 1 \dots M\}$ beinhaltet alle zur Parametrisierung des Netzwerks notwendigen Größen.

Ein Vergleich von (2.50) mit (2.3) zeigt unmittelbar, daß ein solches *feedforward* Netzwerk einen Funktionenapproximator repräsentiert. Ein kNN der Form (2.50) kann daher zur Funktionenapproximation und damit auch zur Mustererkennung verwendet werden. Es drängt sich hier sofort die Frage auf, wie der Parametersatz Θ so bestimmt werden kann, daß das kNN einen in Form eines Trainingsdatensatzes $\mathcal{Z} = \{(\mathbf{x}_k, \mathbf{y}_k) | k = 1 \dots N\}$ vorgegebenen funktionalen Zusammenhang möglichst gut approximiert. Analog zur Lernfähigkeit von biologischen Netzwerken, die durch wiederholte Präsentation von Beispielen neue Zusammenhänge lernen können, werden hierfür Lernalgorithmen benötigt.

Ein Ziel der Neuroinformatik ist damit – unter Berücksichtigung des Lernverhaltens biologischer Netzwerke – die Entwicklung von *Lernregeln*, mit deren Hilfe die Parameter Θ sequentiell an präsentierte Beispiele angepaßt werden können. Gesucht sind Lernregeln, die für ein zufällig aus \mathcal{Z} gezogenes Datentupel $(\mathbf{x}_k, \mathbf{y}_k)$ eine Anpassung

$$\Theta_i^{\text{alt}} = \Theta_i^{\text{neu}} + \epsilon \Delta \Theta_i(\mathbf{x}_k, \mathbf{y}_k) \quad i = 1 \dots M \quad (2.51)$$

vornehmen [And97]. Der *Lernparameter* ϵ ist eine positive Zahl kleiner Eins, die im Zuge des Lernverfahrens langsam verringert wird. Der *Lernschritt* $\Delta \Theta_i(\mathbf{x}_k, \mathbf{y}_k)$ muß derart gestaltet sein, daß sein Betrag bei wiederholter Anwendung der Lernregel (2.51) immer kleiner wird, und so schließlich die Konvergenz von (2.51) eintritt.

Die Neuroinformatik hat im Laufe der letzten Jahrzehnte eine große Zahl verschiedener Typen von kNN entwickelt, die sich wesentlich in der Wahl ihrer Verbindungstopologie und der Formulierung der Lernregeln unterscheiden (siehe z.B. [And97, Roj93, RMS92]). Der nächste Abschnitt diskutiert das für diese Entwicklung wichtige zweischichtige Perzeptron, bevor im Abschnitt 2.4.2 das RBF-Netzwerk eingeführt wird.

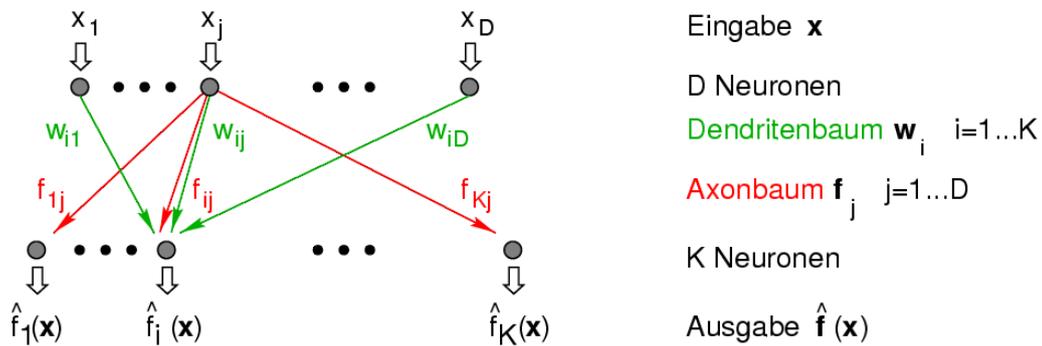


Abbildung 2.15.: Schema eines zweischichtigen Perzeptrons.

2.4.1. Perzeptron

Den ersten Durchbruch bei der Entwicklung lernfähiger neuronaler Netzwerke erzielte F. Rosenblatt 1958 mit dem *Perzeptron* [Ros58], das seitdem für viele Weiterentwicklungen Pate gestanden ist. Abbildung 2.15 zeigt die Verbindungstopologie eines solchen Perzeptrons und weist es als ein zweischichtiges Netzwerk aus. Die Aufgabe dieses Netzwerks ist die *Klassifikation* von Daten. An die Eingabeschicht wird ein Eingabevektor $\mathbf{x} \in \mathbb{R}^D$ angelegt, dessen Komponenten x_j die Aktivitäten der D Eingabeneuronen angeben. Diese Aktivitäten werden vermittels der Dendritenbäume \mathbf{w}_j zu den K Neuronen der Ausgabeschicht transportiert und erzeugen dort eine Eingabe, die durch das Skalarprodukt $\mathbf{w}_i \cdot \mathbf{x}$ gegeben ist. Jedes dieser Neuronen gehört zu einer der K Klassen, die im Datensatz \mathcal{X} enthalten sind. Die Abbildung, die das Perzeptron ausführt, soll von der Form sein, daß bei der Klassifikation eines Datenpunkts \mathbf{x} aus einer Klasse $\lambda(\mathbf{x})$ die Aktivität y_λ des Ausgabeneurons λ Eins wird und die Aktivitäten der verbleibenden Ausgabeneuronen Null werden.

Die Übertragungsfunktion eines Ausgabeneurons i wird bei einem Perzeptron durch eine Stufenfunktion

$$\hat{f}_i(\mathbf{x}; \Theta) = \begin{cases} 1 & : \sum_{j=1}^D w_{ij} x_j > \Phi \\ 0 & : \text{sonst} \end{cases} \quad i = 1 \dots K \quad (2.52)$$

beschrieben. Um eine einfachere Formulierung von (2.52) zu erhalten, kann ein Perzeptron mit dem Schwellenwert Φ in ein äquivalentes Perzeptron mit einem Schwellenwert 0 transformiert werden. Dafür ist es lediglich notwendig, den negativen Schwellenwert $-\Phi$ als Verbindungsstärke $w_0 = -\Phi$ einer fiktiven zusätzlichen Eingabeleitung zu verwenden, die an die konstante Eingabe $x_0 = 1$ angeschlossen wird. Man kann dann auch

$$\hat{f}_i(\mathbf{x}; \Theta) = \begin{cases} 1 & : \mathbf{w}_i \cdot \mathbf{x} > 0 \\ 0 & : \text{sonst} \end{cases} \quad i = 1 \dots K \quad (2.53)$$

schreiben, wobei \mathbf{w}_i den entsprechend erweiterten Dendritenbaum des Ausgabeneurons i repräsentiert. Da $\mathbf{w}_i \cdot \mathbf{x} = 0$ im Merkmalsraum der Eingabedaten \mathcal{X} eine Hyperebene mit

dem Normalenvektor \mathbf{w}_i beschreibt, wird der Merkmalsraum durch jedes Ausgabeneuron in zwei Teilräume aufgeteilt. Die Klassifikation durch ein Perzeptron erfolgt dann entlang dieser Teilraumzerlegungen.

Wie können die Verbindungsstärken $\Theta = \{\mathbf{w}_i | i = 1 \dots K\} = \{\mathbf{f}_j | j = 1 \dots D\}$ nun so berechnet werden, daß ein durch einen Beispieldatensatz $\mathcal{Z} = \{(\mathbf{x}_k, \mathbf{y}_k) | k = 1 \dots N; (\mathbf{y}_k)_i = \delta_{i\lambda_k}\}$ vorgegebenes Klassifikationsproblem möglichst gut gelöst wird? Da die ursprüngliche Lernregel von F. Rosenblatt inzwischen nur geringe Bedeutung hat, wird im folgenden ein Vorschlag von B. Widrow und M. Hoff aus dem Jahre 1960 diskutiert. Für die Herleitung dieser Lernregel wird mit der mittleren quadratischen Abweichung

$$E(\Theta, \mathcal{Z}) = \langle |\hat{\mathbf{f}}(\mathbf{x}_k; \Theta) - \mathbf{y}_k|^2 \rangle_{\mathcal{Z}}. \quad (2.54)$$

ein Fehlermaß festgelegt, mit welchem die Klassifikationsleistung eines Perzeptrons beurteilt werden kann [WH60, WS85]. Im Raum aller Verbindungsstärken definiert dieses Maß ein Fehlergebirge, in dem die Lage des globalen Minimums E_{\min} die Parameter des optimalen Perzeptrons festlegt. Die einfachste Möglichkeit ein Minimum von $E(\Theta, \mathcal{Z})$ zu berechnen ist ein Gradientenabstieg auf $E(\Theta, \mathcal{Z})$ [And97]. Dies führt zu einer sequentiellen Lernregel der Form (2.51) für die Axonbäume \mathbf{f}_j der Eingabeneuronen j mit einem Lernschritt

$$\Delta \mathbf{f}_j(\mathbf{x}_k, \mathbf{y}_k) = (x_j)_k (\mathbf{y}_k - \hat{\mathbf{f}}(\mathbf{x}_k; \Theta)), \quad j = 0 \dots D, \quad (2.55)$$

wobei die j -te Komponente des Eingabevektors \mathbf{x}_k mit $(x_j)_k$ bezeichnet wird. Der Term $(x_j)_k \cdot \mathbf{y}_k$ dieser *Delta*-Regel führt zu einer Verstärkung der Komponenten des Axonbaums \mathbf{f}_j , wenn korrelierte und positive Ein- und Ausgabeaktivitäten vorliegen. Er modelliert damit die von D. Hebb formulierte Hypothese über den Lernmechanismus von Nervenzellen. Der zweite Term $-(x_j)_k \hat{\mathbf{f}}(\mathbf{x}_k; \Theta)$ ist notwendig, um ein unbegrenztes Wachstum der Verbindungsstärken zu verhindern, und entspricht damit einem Zerfallsterm. Daher spricht man bei (2.55) auch von einer kompetitiven Hebbschen Lernregel. Die spezielle Regel (2.55) besagt, daß bei einer richtigen Ausgabe des Neurons die entsprechenden Verbindungsstärken unverändert bleiben. Liegt eine Abweichung vom Sollwert \mathbf{y}_k vor, so werden die Verbindungsstärken so adaptiert, daß der Betrag $|\mathbf{y}_k - \hat{\mathbf{f}}(\mathbf{x}_k; \Theta)|$ beim nächsten Auftreten des Merkmalsvektors \mathbf{x}_k verringert wird.

Allerdings handelt es sich bei der Delta-Regel (2.55) um eine biologisch nicht plausible Lernvorschrift, da in die Anpassung *einer* Verbindungsstärke im Axonbaum die gesamte Netzwerkausgabe $\hat{\mathbf{f}}(\mathbf{x}_k; \Theta)$ eingeht. In einem biologischen Netzwerk wäre dies nur möglich, wenn die Ausgabe $\hat{\mathbf{f}}(\mathbf{x}_k; \Theta)$ für alle Nervenzellen der Eingabeschicht verfügbar wäre, was bei Fehlen rückwärtiger Verbindungen nicht der Fall sein kann. Man spricht daher bei (2.55) auch von einer nichtlokalen Lernregel.

Eine ebenfalls Hebbsche Lernregel für den Axonbaum \mathbf{f}_j ist die *outstar*-Regel

$$\Delta \mathbf{f}_j(\mathbf{x}_k, \mathbf{y}_k) = (x_j)_k (\mathbf{y}_k - \mathbf{f}_j), \quad j = 0 \dots D, \quad (2.56)$$

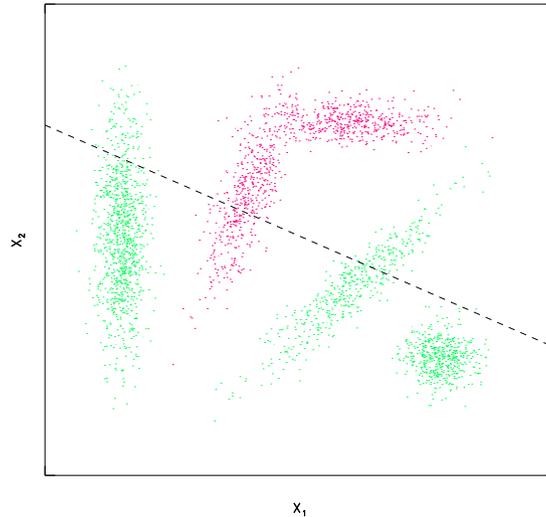


Abbildung 2.16.: **Klassifikation mit einem zweisechichtigen Perzeptron.** Die Datenverteilung \mathcal{X} aus Abbildung 2.8 soll mit einem einfachen zweisechichtigen Perzeptron klassifiziert werden. Das Perzeptron (2.53) ist aus zwei Eingabeneuronen ($D = 2$) und einem Ausgabeneuron ($K = 1$) aufgebaut und wurde mit (2.55) trainiert. In der Abbildung ist der Trainingsdatensatz zusammen mit der Geraden $w_0 + w_1x_1 + w_2x_2 = 0$ eingezeichnet, die durch die Parameter Θ definiert wird.

die erstmals von S. Grossberg vorgeschlagen wurde [Gro74]. In (2.56) wurde die Netzwerkausgabe $\hat{f}(\mathbf{x}_k; \Theta)$ aus (2.55) durch den Axonbaum \mathbf{f}_j ersetzt. Für die Anpassung einer Verbindungsstärke sind nun nur noch lokal am Eingabeneuron j verfügbare Größe erforderlich. Die *outstar*-Regel ist also eine lokale und daher biologisch plausible Lernregel.

Abbildung 2.16 zeigt das Lernergebnis für ein einfaches Perzeptron mit einem einzigen Ausgabeneuron, welches mit der *outstar*-Regel für den bekannten Datensatz aus Abbildung 2.8 trainiert wurde. Man erkennt, wie das Perzeptron versucht, mit einer Geraden die Klassifikation der beiden Klassen zu bewerkstelligen. Allerdings kann dies für den gegebenen Datensatz zu keinem optimalen Ergebnis führen, da die beiden Klassen nicht linear separierbar sind, d.h. nicht durch eine Gerade zu trennen sind.

1969 wiesen M. Minsky und S. Papert [MP69] eine wesentliche Eigenschaft eines zweisechichtigen Perzeptrons mit beliebig vielen Ausgabeneuronen nach, das sog. Perzeptron-Konvergenzkriterium. Dieses sagt aus, daß die Lernregel (2.55) immer konvergiert, falls eine Lösung vorhanden ist. Falls die Daten linear separierbar sind, also durch Hyperebenen zu trennen sind, wird die Lernregel gerade zu dieser optimalen Lösung konvergieren und so zu einer fehlerfreien Klassifikation führen. Ist die Verteilung der verschiedenen Klassen – wie in Abbildung 2.16 – hingegen komplexer (Klassen sind ineinander verzahnt oder eine Klasse ist in eine andere eingebettet), dann macht das zweisechichtige Perzeptron große Fehler.

Die Beschränkung der Problemklasse auf linear separierbare Merkmalsräume, wie sie beim zweischichtigen Perzeptron gegeben ist, wurde durch die Entwicklung des mehrschichtigen Perzeptrons (*multi layer perceptron* (MLP)) aufgehoben [And97]. Beim MLP werden viele Schichten von einfachen Perzeptronen hintereinandergeschaltet, wobei jedes einzelne Neuron noch immer seine Ausgabeaktivität nach (2.53) berechnet. Da jede Schicht eines solchen Perzeptrons zu einer Umkodierung der Merkmalsvektoren führt, ist die Flexibilität eines solchen Netzwerks deutlich größer als diejenige eines zweischichtigen Perzeptrons. 1989 wurde sogar gezeigt [Cyb89, HSW89, Hor91, Fun89], daß schon ein dreischichtiges Perzeptron mit sigmoiden Antwortcharakteristiken der Zwischenschichtneuronen jede Funktion beliebig genau approximieren kann, wenn die Zahl der Neuronen dort ausreichend groß ist. Allerdings war lange nicht klar, mit welchen Lernregeln die Verbindungsstärken eines solchen dreischichtigen MLP bestimmt werden sollen. Erst Ende der achtziger Jahre setzte sich immer mehr der *back propagation*-Algorithmus durch [Wer74, RHW86]. Das Ziel dieses Verfahrens ist die Minimierung der quadratischen Differenz zwischen dem Sollwert y_k und der Netzwerkausgabe $\hat{f}(\mathbf{x}_k; \Theta)$ durch einen Gradientenabstieg auf der Fehlerfunktion $E(\Theta, \mathcal{Z})$ [siehe (2.54)]. Dies führt zu einer Lernregel, bei der der vom Netzwerk gemachte Fehler "rückwärts" durch das Netzwerk propagiert werden muß, um die Verbindungsstärken anpassen zu können. Dieser Algorithmus hat in den letzten Jahren eine breite Anwendung bei praktischen Problemen gefunden. Da durch den Gradientenabstieg im gesamten Parameterraum gleichzeitig nach einer optimalen Lösung für die Gewichte gesucht wird, konvergiert das Verfahren aber nur sehr langsam und findet häufig nur suboptimale Lösungen [And97]. Zudem konnte in biologischen Netzwerken bisher noch nicht beobachtet werden, daß eine Rückwärtspropagation von Fehlerinformationen stattfindet [Cri89].

2.4.2. GRBF-Netzwerk

Seit Anfang der neunziger Jahre ist ein dreischichtiger Netzwerktyp immer populärer geworden, der sich von einem dreischichtigen MLP dadurch unterscheidet, daß die Übertragungsfunktion der Neuronen auf der Zwischenschicht nicht durch eine sigmoide Funktion oder Stufenfunktion, sondern durch eine nicht normierte, univariate Gaußfunktion $g(|\mathbf{c}_r - \mathbf{x}|, \sigma)$ mit der Varianz σ^2 beschrieben wird (siehe z.B. [CCG91, MAC+95, BFG95, BF95, KL98]). Die Aktivitäten der M Zwischenschichtneuronen sind bei diesem Netzwerk, dessen Struktur zusammen mit den Bezeichnungen in Abbildung 2.17 dargestellt ist, durch

$$a_r(\mathbf{x}; \Theta_r) = g(|\mathbf{c}_r - \mathbf{x}|, \sigma) \equiv \exp\left(-\frac{|\mathbf{c}_r - \mathbf{x}|^2}{2\sigma^2}\right) \quad (2.57)$$

gegeben. $\mathbf{c}_r \in \mathbb{R}^D$ umfaßt die Verbindungsstärken des Dendritenbaums eines Zwischenschichtneurons r und $\mathbf{x} \in \mathbb{R}^D$ die Aktivität der Eingabeneuronen. In neurobiologischer Interpretation spezifiziert \mathbf{c}_r das Zentrum eines isotropen, gaußförmigen rezeptiven Felds der Breite σ im Merkmalsraum, der durch die Eingangsreize aufgespannt wird [PG89]. Der

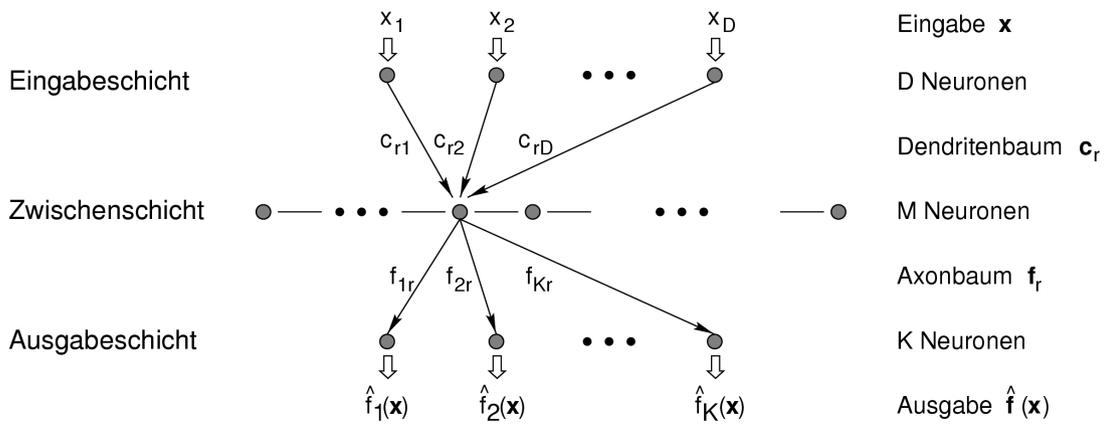


Abbildung 2.17.: **Schema von RBF- und GRBF-Netzwerken.** Solche Netzwerke stellen nach T. Poggio und F. Girosi [PG89] Modelle des Cerebellums dar (vgl. Abschnitt 2.2.2). Dabei werden die Eingabeneuronen mit den Moosfasern, die Neuronen in der Zwischenschicht mit den Körnerzellen und die Ausgabeneuronen mit den Purkinjezellen identifiziert (vgl. Abbildung 2.4 und 2.5). Die lateralen Verbindungen in der Zwischenschicht repräsentieren die *feedback*-Wechselwirkung der Körnerzellen.

Parametersatz Θ_r , der zur Spezifikation der Übertragungsfunktion (2.57) benötigt wird, enthält also lediglich \mathbf{c}_r und σ . Die Netzwerkausgabe $\hat{\mathbf{f}}(\mathbf{x}; \Theta)$ ergibt sich durch eine Linearkombination der Aktivitäten $a_r(\mathbf{x}; \Theta_r)$, die mit den Verbindungsstärken $\mathbf{f}_r \in \mathbb{R}^K$ im Axonbaum der Zwischenschichtneuronen gewichtet sind, gemäß

$$\hat{\mathbf{f}}(\mathbf{x}; \Theta) = \sum_{r=1}^M \mathbf{f}_r a_r(\mathbf{x}; \Theta_r). \quad (2.58)$$

Da die Übertragungsfunktionen $a_r(\mathbf{x}; \Theta_r)$ der Zwischenschichtneuronen nach (2.57) um \mathbf{c}_r zentrierte, radialsymmetrische Gaußfunktionen mit der Varianz σ^2 sind und die Ausgabe (2.58) eine Linearkombination dieser Funktionen ist, wird ein Netzwerk, welches durch (2.57) und (2.58) beschrieben wird, auch als radiales Basisfunktionen-Netzwerk oder kurz RBF-Netzwerk bezeichnet.

Das RBF-Netzwerk hat seinen Ursprung in den oben schon erwähnten Untersuchungen von M. Powell zur Funktioneninterpolation mit radialen Basisfunktionen [Pow87]. T. Poggio und F. Girosi stellten erstmals die Verbindung zur Neurobiologie her [PG89, PG90b]. 1988 schlugen J. Moody und C. Darken eine Modifikation des RBF-Netzwerks vor [MD88], die im folgenden als generalisiertes RBF-Netzwerk oder GRBF-Netzwerk bezeichnet wird.

Beim GRBF-Netzwerk werden die radialsymmetrischen Übertragungsfunktionen (2.57) durch die normierten Zwischenschichtaktivitäten

$$a_r(\mathbf{x}; \Theta^c) = \frac{g(|\mathbf{c}_r - \mathbf{x}|, \sigma)}{A(\mathbf{x}; \Theta^c)} \quad (2.59)$$

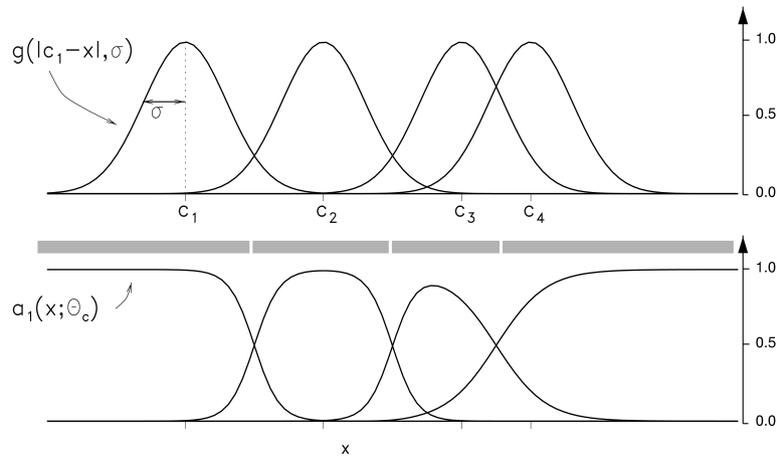


Abbildung 2.18.: **Unnormierte und normierte Basisfunktionen.** Dargestellt ist die Form von vier Basisfunktionen ($M = 4$) für einen eindimensionalen Merkmalsraum ($D = 1$). Oben sind die univariaten Gaußfunktionen $g(|\mathbf{c}_r - \mathbf{x}|, \sigma)$ und unten die normierten Basisfunktionen $a_r(\mathbf{x}; \Theta^c)$ dargestellt. Bei den $a_r(\mathbf{x}; \Theta^c)$ handelt es sich nicht mehr um radialsymmetrische Funktionen. Trotzdem sind die $a_r(\mathbf{x}; \Theta^c)$ in der Regel um den Punkt \mathbf{c}_r stark lokalisierte Funktionen. Wegen $\sum_r a_r(\mathbf{x}; \Theta^c) = 1$ wird der Merkmalsraum durch die $a_r(\mathbf{x}; \Theta^c)$ partitioniert. Dies wird durch die vier grauen Balken angedeutet.

ersetzt, wobei die normierende Funktion

$$A(\mathbf{x}; \Theta^c) = \sum_{r=1}^M g(|\mathbf{c}_r - \mathbf{x}|, \sigma) \quad (2.60)$$

die Summe aller univariaten Gaußfunktionen ist, die auch als *Gesamtaktivität* der Zwischenschicht bezeichnet wird. Faßt man, wie in der Erläuterung zu Abbildung 2.17 skizziert, GRBF-Netzwerke als Modelle für das Cerebellum auf, dann modelliert die Normierung in (2.59) die laterale *feedback*-Wechselwirkung der Körnerzellen. Nach (2.59) enthält der zur Spezifikation der GRBF-Übertragungsfunktion nötige Parametersatz Θ^c also die Dendritenbäume \mathbf{c}_r aller M Zwischenschichtneuronen, d.h. $\Theta^c = \{\sigma, \mathbf{c}_r | r = 1 \dots M\}$.

Abbildung 2.18 illustriert an einem eindimensionalen Beispiel die Gemeinsamkeiten und Unterschiede der Übertragungsfunktionen (2.57) und (2.59). Man erkennt dort, daß sich die durch die Aktivitätsfunktionen definierten rezeptiven Felder der Zwischenschichtneuronen bei der gegebenen Wahl der Gaußbreite σ in beiden Fällen überlappen. Die Neuronen *kooperieren* daher bei der Kodierung von Eingangsreizen \mathbf{x} aus den Überlappungsbereichen. Neben dieser Gemeinsamkeit sind jedoch vor allem die folgenden beiden Unterschiede von Bedeutung: (i) die Übertragungsfunktion (2.59) des GRBF-Netzwerks ist keine um \mathbf{c}_r radialsymmetrische Funktion und (ii) die entsprechenden $a_r(\mathbf{x}; \Theta^c)$ haben auf Grund der

Normierung mit (2.60) die Eigenschaft

$$\sum_{r=1}^M a_r(\mathbf{x}; \Theta^c) = 1, \quad (2.61)$$

die besagt, daß jeder Eingangsreiz \mathbf{x} dieselbe *effektive* Gesamtaktivität auf der Zwischenschicht induziert. Nach C. v. Malsburg [Mal73] und D. Dersch [Der95] kann eine derartige Eigenschaft auch als *Konkurrenz* der Zwischenschichtneuronen um die Kodierung der Eingangsreize \mathbf{x} aufgefaßt werden. Ein solches Element der Konkurrenz fehlt im einfachen Fall des RBF-Modells. Hier ist die reizinduzierte Aktivität eines Neurons lediglich durch die Form des rezeptiven Felds, nicht jedoch, wie beim GRBF-Netzwerk, durch den Aktivitätszustand der übrigen Neuronen gegeben. Mathematisch sind Übertragungsfunktionen (2.59) wegen (2.61) also Partitionierungsfunktionen, die den Merkmalsraum in M unscharf abgegrenzte Gebiete aufteilen.

Die Ausgabe eines GRBF-Netzwerks wird analog zum RBF-Netzwerk wiederum durch die Überlagerung

$$\hat{\mathbf{f}}(\mathbf{x}; \Theta^{GRBF}) = \sum_{r=1}^M \mathbf{f}_r a_r(\mathbf{x}; \Theta^c) \quad (2.62)$$

der mit den Verbindungsstärken \mathbf{f}_r in den Axonbäumen gewichteten Zwischenschichtaktivitäten gebildet. Der Parametersatz Θ^{GRBF} , welcher ein GRBF-Netzwerk vollständig beschreibt, setzt sich also aus den Parametern $\Theta^c = \{\sigma, \mathbf{c}_r | r = 1 \dots M\}$ der Zwischenschicht, d.h. den Dendritenbäumen \mathbf{c}_r und der gemeinsamen Gaußbreite σ , und den Axonbäumen $\Theta^f = \{\mathbf{f}_r | r = 1 \dots M\}$ zusammen.

RBF- vs. GRBF-Netzwerk

Worin unterscheidet sich die Arbeitsweise eines GRBF-Netzwerks von einem RBF-Netzwerk? An Hand von Abbildung 2.19 lassen sich die beiden Ansätze (2.58) und (2.62) zur Funktionenapproximation gut vergleichen. In beiden Fällen wurden identische Dendritenbäume \mathbf{c}_r und Axonbäume \mathbf{f}_r gewählt, die nicht durch einen Lernalgorithmus sondern aus der Kenntnis der zu approximierenden Funktion $f(x)$ bestimmt wurden. In der Abbildung ist die Funktionenapproximation für eine Gaußbreite σ von jeweils 0.02 (blau) und 0.1 (rot) dargestellt.

Aus dem Beispiel wird deutlich, daß beide Modelle bei hinreichend starker Kooperativität der Zwischenschichtneuronen (große Gaußbreite σ) glatte Funktionenapproximationen (rote Linien) liefern können. Es zeigt ferner, daß die Approximation einer Funktion durch ein GRBF-Netzwerk bei abnehmender Kooperation und zunehmender Konkurrenz (abnehmendes σ) auf Grund der zunehmend schärferen Partitionierung des Eingaberaums immer mehr zu einer Stufenfunktion übergeht. Bei verschwindendem σ entspricht das GRBF-Netzwerk daher einer Diskretisierung der Funktion durch eine Tabelle. Die Qualität einer

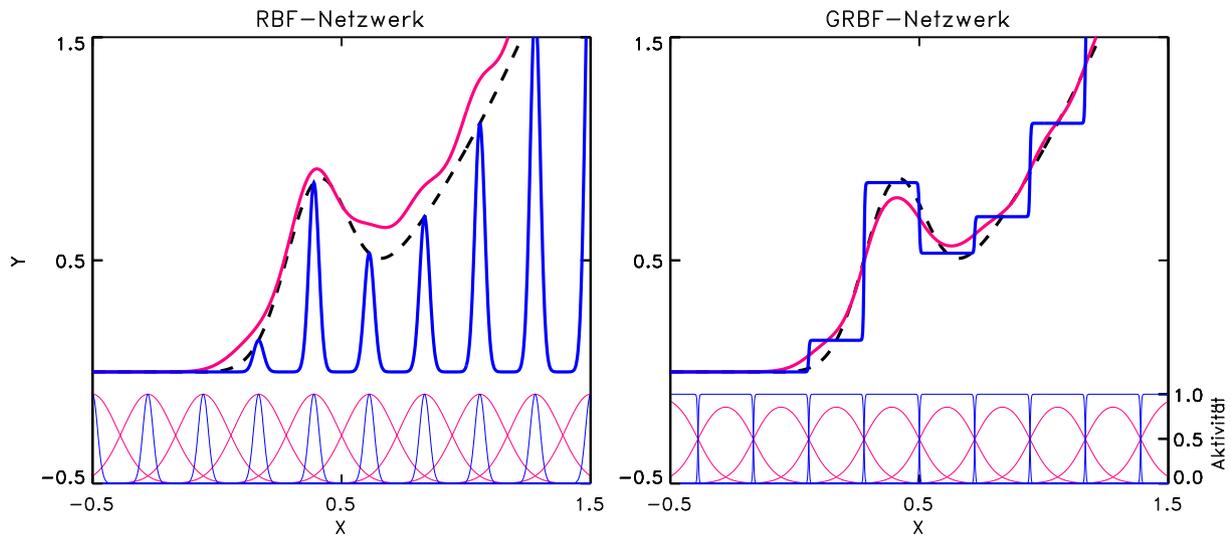


Abbildung 2.19.: **Funktionenapproximation durch ein RBF- und GRBF-Netzwerk.** Für den Vergleich der beiden Netzwerke wurden die Parameter aus der analytischen Kenntnis der zu approximierenden Funktion $f(x)$ (gestrichelte Linie) bestimmt. Die Gaußzentren c_r der $M = 10$ Zwischenschichtneuronen sind äquidistant im Intervall $[-0.5, 1.5]$ verteilt. Die Verbindungsstärken f_r ergeben sich durch den Wert $f_r = f(c_r)$ der zu approximierenden Funktion $f(x)$ am Ort $x = c_r$. Die roten Kurven zeigen die Approximationen für eine Gaußbreite von $\sigma = 0.1$, und die blauen Kurven für $\sigma = 0.02$. Zusätzlich sind noch die Aktivitäten (2.57) bzw. (2.59) der Zwischenschichtneuronen eingetragen.

solchen Approximation kann bei einer ausreichend großen Zahl M von Zwischenschichtneuronen beliebig gesteigert werden. Bei $\sigma > 0$ werden die Stufen zunehmend geglättet. Anders verhält sich das RBF-Netzwerk bei verschwindendem σ . In diesem rein kooperativen Netzwerk werden die Übertragungsfunktionen (2.57) bei verschiedener Kooperativität zu stark um c_r lokalisierten Funktionen, so daß das RBF-Netzwerk nur noch für $x \equiv c_r$ eine korrekte Ausgabe liefert. Im Gegensatz zum rein kooperativen RBF-Netzwerk bleibt also das kooperativ-kompetitive GRBF-Netzwerk auch im Grenzfall (kleine σ) generalisierungsfähig. Der hier angesprochene Vorteil von Modellen, die Kooperation und Konkurrenz in sich vereinen, gegenüber rein kooperativen Modellen, wird bei der von mir vorgestellten Entwicklung einer allgemeinen Architektur von Systemen lokaler Experten in Kapitel 4 noch von entscheidender Bedeutung sein.

Die Zwischenschichtneuronen als lokale Merkmalsdetektoren

Ein großer Vorteil des GRBF-Netzwerks ist – im Gegensatz zu den *multi layer*-Perzeptronen – die gute Interpretierbarkeit der Parametersätze Θ^{GRBF} , die die speziellen Eigenschaften dieser Netzwerke festlegen. Diese Eigenschaften sind zum einen durch die Ausgabeak-

tivitäten der Zwischenschichtneuronen, die von den Verbindungsstärken c_r und der Ausdehnung σ bestimmt werden, und zum anderen durch die Verbindungsstärken f_r in den Axonbäumen zur Ausgabeschicht charakterisiert. Die Verbindungsstärken c_r legen prototypische Punkte im Merkmalsraum \mathbb{R}^D fest, an welchen die Verbindungsstärken f_r “angeheftet” sind. Durch die – abhängig von σ – mehr oder minder glatten Partitionierungsfunktionen (2.59) wird für jedes Zwischenschichtneuron im Merkmalsraum ein Bereich definiert, auf den das Neuron besonders empfindlich reagiert, d.h. eine Eingabe \mathbf{x} aus einem solchen Bereich führt zu einer Ausgabeaktivität $a_r(\mathbf{x}; \Theta^c)$ des Neurons r nahe Eins und daher für kleines σ zu einer Netzwerkausgabe $\hat{\mathbf{f}}(\mathbf{x}; \Theta^{GRBF}) \approx \mathbf{f}_r$. Für größere Werte des Kooperationsparameters σ findet dagegen eine Interpolation der Funktionswerte \mathbf{f}_r gemäß (2.62) statt. Man spricht daher bei den Zwischenschichtneuronen auch von *lokalen Merkmalsdetektoren*, da diese nur auf Merkmalskombinationen \mathbf{x} ansprechen, die aus einer unscharf begrenzten Umgebung des prototypischen Punkts c_r stammen.

Diese Eigenschaft modelliert die Beobachtung, daß viele Nervenzellen der sensorischen Cortices nur für bestimmte Merkmalskombinationen empfindlich sind. Zum Beispiel sind die Nervenzellen im somato-sensorischen Cortex auf bestimmte Bereiche der Körperoberfläche spezialisiert und reagieren nur auf dort örtlich lokalisierte Reize [EKS94]. Eine ähnliche Beobachtung macht man im visuellen Cortex. Hier finden sich Nervenzellen, die selektiv auf Signale von der Retina reagieren, die von einer Objektorientierung mit einem bestimmten Winkel herrühren [Hub90]. In beiden Fällen wird die Selektivität der Reizantwortcharakteristik der Nervenzellen ergänzt durch eine Kooperation vieler Zellen bei der Kodierung eines gegebenen Eingangsreizes, die aus der Überlagerung der zugehörigen rezeptiven Felder herrühren (für eine eingehende Diskussion der resultierenden Populationskodierung siehe [HKRT94]). Diese beiden Beispiele verdeutlichen ferner, daß man einer Nervenzelle häufig sowohl eine physikalische Position, also die geometrische Lage im Gehirn, wie auch eine *virtuelle* Position im Eingaberaum zuordnen kann [Tgk90].

In der obigen Diskussion wurde die Frage der Parameterbestimmung nicht berücksichtigt. T. Poggio und F. Girosi untersuchten für das RBF-Netzwerk ein Regressionsverfahren, welches durch einen Gradientenabstieg auf der Fehlerfunktion (2.5) alle Netzwerkparameter gleichzeitig optimiert [PG89]. Auf Grund der langsamen Konvergenz und der suboptimalen Resultate dieses Verfahrens (siehe [Ado92] für eine eingehende Diskussion) schlugen J. Moody und C. Darken für die Parameterschätzung des GRBF-Netzwerks ein heuristisches dreistufiges Vorgehen vor [MD89], für welches kein globales Optimierungskriterium angegeben werden kann. Die (i) Zentren c_r werden durch Anwendung des *k-means clusterings* [Llo82] und (ii) die Gaußbreite σ durch eine Heuristik bestimmt. Anschließend werden (iii) die Axonbäume f_r mit Hilfe der Delta-Lernregel (2.55) berechnet. Wie im folgenden Abschnitt gezeigt wird, können (i) und (ii) durch ein anderes, besser begründetes Verfahren ersetzt und in einem gemeinsamen Optimierungsschritt zusammengefaßt werden.

Das GRBF-Netzwerk und ML-Dichteschätzung

Vergleicht man nämlich die Zwischenschichtaktivität (2.59) des GRBF-Netzwerks mit der *a posteriori*-Wahrscheinlichkeit (2.42) bei der Dichteschätzung mit einer Mischung multivariater Normalverteilungen, so fällt auf, daß diese beiden Ausdrücke im Fall gleichgewichteter univariater Normalverteilungen identischer Varianz ($P_r = 1/M$ und $\Sigma_r^{-1} = \mathbf{I}\sigma^{-2}$) übereinstimmen, d.h.

$$P(r|\mathbf{x}, \Theta^c) \equiv a_r(\mathbf{x}; \Theta^c). \quad (2.63)$$

Bis auf den Normierungsfaktor entspricht auch die unnormierte Aktivität $g(|\mathbf{c}_r - \mathbf{x}|, \sigma)$ in Gleichung (2.59) der Komponente (2.36) der Mischungsdichte für den univariaten Fall. Ebenso gleicht – bis auf einen Faktor – die Gesamtaktivität $A(\mathbf{x}; \Theta^c)$ der Mischungsdichte (2.35). Die Zwischenschicht eines GRBF-Netzwerks kann daher einer ML-Dichteschätzung der Eingabedaten \mathcal{X} mit einer Mischung univariater Normalverteilungen zugeordnet werden. In Abschnitt 2.3.3 wurde festgestellt, daß alle Parameter Θ^c einer solchen Schätzung mit dem *univar*-Algorithmus (modifiziertes RGF-*clustering*, vgl. Abschnitt 2.3.2) zuverlässig bestimmt werden können. Es bietet sich daher an die Dendritenbäume \mathbf{c}_r der Zwischenschichtneuronen mit Hilfe des in (2.19) und (2.20) definierten sequentiellen Algorithmus und die Gaußbreiten σ_f durch die Maximierung der *log-likelihood* zu berechnen. Dann werden die Aktivitäten $a_r(\mathbf{x}; \Theta^c)$ zu *a posteriori*-Wahrscheinlichkeiten $P(r|\mathbf{x}, \Theta^c)$ einer ML-Dichteschätzung mit einer Mischung univariater Normalverteilungen.

Training der Verbindungsstärken \mathbf{f}_r

In Abschnitt 2.4.1 haben wir festgestellt, daß die Delta-Regel (2.55) zwar zu einer Minimierung des Fehlermaßes (2.54) führt, aber wegen ihrer nichtlokalen Struktur biologisch nicht plausibel ist. U. Adolphs schlug daher erstmals vor, die Verbindungsstärken \mathbf{f}_r des Axonbaums in einem GRBF-Netzwerk mit einer entsprechend modifizierten *outstar*-Regel (2.56) zu trainieren [Ado92]. Diese Lernregel, die bei Klassifikationsaufgaben aus dem Bereich der Spracherkennung zu guten Ergebnissen geführt hat [Alb95, Gro95], wird auch in dieser Arbeit für das Training der \mathbf{f}_r verwendet. Sie hat nun die Form

$$\Delta \mathbf{f}_r = a_r(\mathbf{x}_k; \Theta^c)(\mathbf{y}_k - \mathbf{f}_r), \quad r = 1 \dots M. \quad (2.64)$$

Abbildung 2.20 zeigt die Approximation der Daten aus Abbildung 2.6 durch das oben beschriebene GRBF-Netzwerk. Dabei wurden für die Bestimmung von Θ^c die Eingabedaten \mathcal{X} zunächst einem *clustering* mit dem *univar*-Algorithmus unterzogen. Dieser wurde am Maximum der *log-likelihood* beendet. Anschließend wurden die Verbindungsstärken \mathbf{f}_r mit der sequentiellen *outstar*-Regel (2.56) berechnet. Es ist zu erkennen, wie die Approximation für eine hohe Dichte an Datenpunkten relativ gut ist, aber für Bereiche mit wenigen Merkmalsdetektoren ($x \gtrsim 0.9$) versagt. Andererseits wird der konstante Funktionsbereich ($x \lesssim 0$) sehr gut angenähert.

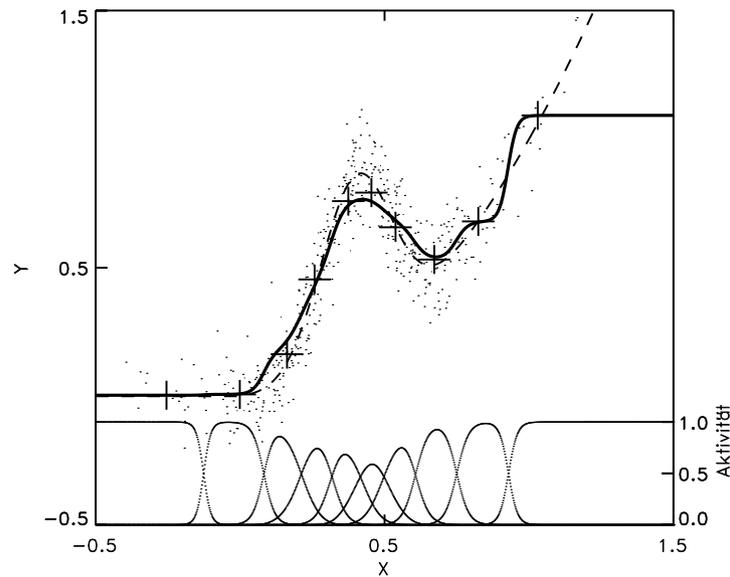


Abbildung 2.20.: **Funktionenapproximation mit einem GRBF-Netzwerk.** Ein GRBF-Netzwerk mit $M = 10$ Zwischenschichtneuronen wurde mit dem Datensatz \mathcal{Z} aus Abbildung 2.6 trainiert. Die durchgezogene Linie ist die Approximation $\hat{f}(x; \Theta^{GRBF})$ durch das GRBF-Netzwerk und die gestrichelte Linie die dem Datensatz zu Grunde liegende Funktion. Die Kreuze geben die Lage der Merkmalsdetektoren c_r (horizontale Position) und der Verbindungsstärken f_r (vertikale Position) wieder. Die Achsenlänge der Kreuze entspricht σ_f . Unten sind zusätzlich die Zwischenschichtaktivitäten $a_r(x; \Theta^c)$ eingezeichnet.

Fazit

Zusammenfassend sei festgehalten, daß die in unserer Arbeitsgruppe entwickelte Kombination [Ado92, Der95, KT97, ABK⁺99] des *univar*-Verfahrens mit der *outstar*-Regel (2.64) ein Verfahren zur Parameterschätzung von GRBF-Netzwerken darstellt, das sicher konvergiert und qualitativ akzeptable Funktionenapproximationen liefert, für das jedoch ein Fehlerfunktional als globales Optimierungskriterium fehlt. Entsprechend gab es bis zu meiner Arbeit kein stabiles Lernverfahren zur gleichzeitigen Bestimmung aller Parameter.

3. GRBF-Netzwerke, Dichteschätzung und Funktionenapproximation

Im letzten Kapitel wurden verschiedene Verfahren zur Funktionenapproximation vorgestellt. Wie wir gesehen haben, beruhen klassische Verfahren, wie die Regression, auf der Minimierung eines vorgegebenen Fehlermaßes. Dagegen werden die Parameter von künstlichen neuronalen Netzen, die ebenfalls der Funktionenapproximation dienen, gelegentlich, wie im Falle des diskutierten GRBF-Netzwerks, mittels sequentieller Lernregeln bestimmt, deren Optimierungsziel nicht immer vollständig bekannt ist. Neben diesen beiden Möglichkeiten ist aber auch eine Funktionenapproximation

$$\hat{\mathbf{f}}(\mathbf{x}; \Theta) \equiv \int_{\mathbb{R}^K} \mathbf{y} \hat{p}(\mathbf{y}|\mathbf{x}, \Theta) d\mathbf{y} \quad (3.1)$$

auf Grundlage einer Schätzung der bedingten Wahrscheinlichkeitsdichte $\hat{p}(\mathbf{y}|\mathbf{x}, \Theta)$ denkbar. Bei diesem Vorgehen wird für ein vorgegebenes \mathbf{x} der Erwartungswert von \mathbf{y} berechnet. Mit dem Bayesschen Satz $\hat{p}(\mathbf{y}|\mathbf{x}, \Theta) = \hat{p}(\mathbf{x}, \mathbf{y}|\Theta)/\hat{p}(\mathbf{x}|\Theta)$ und mit $\hat{p}(\mathbf{x}|\Theta) = \int \hat{p}(\mathbf{x}, \mathbf{y}|\Theta) d\mathbf{y}$ kann für (3.1) auch

$$\hat{\mathbf{f}}(\mathbf{x}; \Theta) = \frac{\int_{\mathbb{R}^K} \mathbf{y} \hat{p}(\mathbf{x}, \mathbf{y}|\Theta) d\mathbf{y}}{\int_{\mathbb{R}^K} \hat{p}(\mathbf{x}, \mathbf{y}|\Theta) d\mathbf{y}} \quad (3.2)$$

geschrieben werden. Für eine Funktionenapproximation nach (3.2) ist also eine Schätzung $\hat{p}(\mathbf{x}, \mathbf{y}|\Theta)$ der zum Trainingsdatensatz $\mathcal{Z} = \{\mathcal{X}, \mathcal{Y}\} = \{(\mathbf{x}_k, \mathbf{y}_k) | k = 1 \dots N\}$ gehörigen Verbundwahrscheinlichkeitsdichte $p(\mathbf{x}, \mathbf{y})$ notwendig.

Nach Abschnitt 2.3.3 bieten dazu Mischungen *multivariater* Normalverteilungen einen möglichen Modellansatz, für den zudem mit dem von M. Kloppenburg und P. Tavan vorgeschlagenen EM-Abkühlverfahren ein sicher konvergierender Weg zur Parameteroptimierung im Sinne der Maximierung der *likelihood* vorliegt.

Es ist das Ziel dieses Abschnitts,

- diesen Ansatz näher zu erkunden und
- seine Beziehung zu GRBF-Netzwerken aufzuklären.

Die zweite Fragestellung liegt nach den Ergebnissen von Abschnitt 2.4.2 nahe, da dort gezeigt wurde, daß die Parameter Θ^c der Zwischenschicht eines GRBF-Netzwerks vermittels einer ML-Schätzung der Dichte $p(\mathbf{x})$ der Eingabedaten bestimmt werden können und daß die Funktion (2.60) zur Berechnung der Gesamtaktivität bis auf konstante Faktoren als ML-Mischungsdichte $\hat{p}(\mathbf{x}; \Theta^c)$ univariater Normalverteilungen aufgefaßt werden kann. Allerdings fordert der Ansatz (3.2) statt einer Schätzung von $p(\mathbf{x})$ eine Schätzung der Verbunddichte $p(\mathbf{x}, \mathbf{y})$. Es stellt sich damit die Frage, ob und gegebenenfalls wie das GRBF-Netzwerk aus Abbildung 2.17 erweitert werden kann, damit allen freien Parametern Θ einer zugehörigen ML-Mischungsdichte aus multivariaten Normalverteilungen $\hat{p}(\mathbf{x}, \mathbf{y}|\Theta)$ ein Netzwerkkorrelat zugeordnet werden kann.

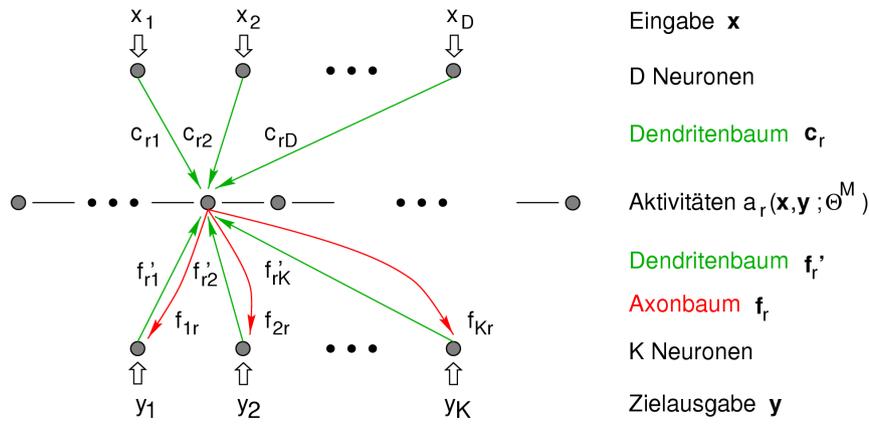
Eine solche Zuordnung ist dabei vor allem deshalb von Interesse, weil eine Abbildung zunächst rein mathematisch motivierter Konzepte auf neurobiologische Modellvorstellungen erfahrungsgemäß wichtige Anregungen zur Lösung schwieriger algorithmischer Fragestellungen liefern kann (siehe z.B. [Der95, ABK⁺99]). Darüberhinaus kann eine derartige neurobiologische Interpretation möglicherweise auch zum Verständnis der Hirnfunktion beitragen.

Im vorliegenden Fall erhebt sich beispielsweise die weitere Frage, ob und wie das erwähnte EM-Abkühlverfahren zur ML-Dichteschätzung als *sequentieller* stochastischer Lernalgorithmus, d.h. als neurobiologisch plausibles ‐Lernen aus Beispielen‐ formuliert werden kann. Solche Algorithmen haben den Vorteil sich auch für adaptives *on-line* Lernen zu eignen. Es sei dabei daran erinnert, daß das EM-Abkühlverfahren nach (2.37), (2.47) und (2.48) zunächst vermittels statistischer Erwartungswerte formuliert worden ist.

Bevor im folgenden der in Gleichung (3.2) formulierte Ansatz zur Funktionenapproximation näher untersucht wird, soll zunächst die Frage untersucht werden, in welcher Beziehung (3.2) zu einem GRBF-Netzwerk steht.

3.1. Erweiterung des GRBF-Netzwerks

In Abschnitt 2.4.2 wurde festgestellt, daß ein Teil der Parameter des in Abbildung 2.17 schematisch dargestellten GRBF-Netzwerks zur Funktionenapproximation vermittels des zum *univar*-Verfahren erweiterten RGF-*clusterings* festgelegt werden kann. Es waren dies die Zentren \mathbf{c}_r und die Breiten σ_f der rezeptiven Felder der Zwischenschichtneuronen r , während für die Axonbäume \mathbf{f}_r dieser Neuronen, welche die übrigen Parameter aus Θ^{GRBF} umfassen, eine separate Lernregel (2.64) verwendet wurde. Vergleicht man nun die lokale


 Abbildung 3.1.: Schema eines GRBF-Netzwerks mit Rückwärtsverbindungen \mathbf{f}'_r .

Lernregel (2.64) für die \mathbf{f}_r mit der RGF-*clustering* Lernregel (2.20) für die \mathbf{c}_r , so fällt die formale Ähnlichkeit dieser beiden kompetitiven Hebbschen Regeln sofort auf. In beiden Fällen ist der Hebbsche Wachstumsterm das Korrelationsprodukt der Zwischenschichtaktivität a_r mit den Aktivitäten \mathbf{x} bzw. \mathbf{y} der prä- bzw. postsynaptischen Neuronen. Auch die wachstumsbegrenzenden Terme $-a_r \mathbf{c}_r$ und $-a_r \mathbf{f}_r$ haben die gleiche Form und umfassen lediglich lokale, am Neuron r verfügbare Größen. Der wesentliche Unterschied besteht darin, daß beim Training des *feedforward* Netzes aus Abbildung 2.17 in die Aktivierung a_r des Zwischenschichtneurons r lediglich der Eingangsreiz \mathbf{x}_k nicht jedoch die zugehörige Sollausgabe \mathbf{y}_k aus dem Beispieldatenpaar

$$\mathbf{z}_k \equiv (\mathbf{x}_k, \mathbf{y}_k) \in \mathcal{Z} \quad (3.3)$$

eingeht, so daß die Sollausgabe nach (2.64) beim Axonwachstum keinen Einfluß auf den präsynaptischen Teil a_r des Hebbschen Korrelationsprodukts hat.

Im Gegensatz zu dieser Modellarchitektur können im Falle des Cerebellums die Übertragungseigenschaften der Axone der Körnerzellen zu den Pukinjezellen mittels der Kletterfasern präsynaptisch (bzgl. der die Ausgabe kodierenden Purkinjezellen) durch Vorgabe einer Sollausgabe moduliert werden (vgl. Abschnitt 2.2). Zur Erfassung dieses Befunds muß daher die *feedforward* Architektur des GRBF-Modells erweitert werden. Dies kann, wie in Abbildung 3.1 dargestellt ist, durch Einführung rückwärtiger Verbindungen \mathbf{f}'_r von den Ausgabeneuronen zur Zwischenschicht, d.h. durch Erweiterung der Dendritenbäume $\mathbf{c}_r \in \mathbb{R}^D$ zu Dendritenbäumen

$$\mathbf{d}_r \equiv (\mathbf{c}_r, \mathbf{f}'_r) \in \mathbb{R}^D \times \mathbb{R}^K \quad (3.4)$$

geschehen. Während der Trainingsphase fungieren die Ausgabeneuronen damit als zusätzliche Eingabeneuronen und die Zwischenschichtaktivitäten a_r werden zu Funktionen des Eingangsreizes \mathbf{x}_k und der Sollausgabe \mathbf{y}_k , d.h.

$$a_r \equiv a_r(\mathbf{z}_k; \Theta) = a_r(\mathbf{x}_k, \mathbf{y}_k; \Theta), \quad (3.5)$$

wobei Θ die Menge der Netzwerkparameter ist. Verwendet man nun für den erweiterten Dendritenbaum die bisherige *clustering* Lernregel, so hat sie die Form

$$\Delta \mathbf{c}_r = a_r(\mathbf{x}_k, \mathbf{y}_k; \Theta) (\mathbf{x}_k - \mathbf{c}_r) + \boldsymbol{\nu} \quad (3.6)$$

und

$$\Delta \mathbf{f}'_r = a_r(\mathbf{x}_k, \mathbf{y}_k; \Theta) (\mathbf{y}_k - \mathbf{f}'_r) + \boldsymbol{\nu}' \quad (3.7)$$

und kann unter Verwendung der Definitionen (3.3) und (3.4) sowie mit $\tilde{\boldsymbol{\nu}} \equiv (\boldsymbol{\nu}, \boldsymbol{\nu}')$ auch zu

$$\Delta \mathbf{d}_r = a_r(\mathbf{z}_k; \Theta) (\mathbf{z}_k - \mathbf{d}_r) + \tilde{\boldsymbol{\nu}} \quad (3.8)$$

zusammengefaßt werden. Die Lernregel (2.64) für die Axonbäume hat nun die Form

$$\Delta \mathbf{f}_r = a_r(\mathbf{x}_k, \mathbf{y}_k; \Theta) (\mathbf{y}_k - \mathbf{f}_r) \quad (3.9)$$

und ist mit der Lernregel (3.7) für die Rückwärtsverbindungen bis auf den Rauschterm identisch. Falls für diesen Term $\langle \boldsymbol{\nu}' \rangle = 0$ gilt, dann konvergieren die synaptischen Gewichte f_{jr} zu denselben Werten wie die f'_{rj} , d.h. es gilt am Ende des Lernprozesses

$$f_{jr} = f'_{rj} \quad \text{bzw.} \quad \mathbf{f}_r = \mathbf{f}'_r. \quad (3.10)$$

Die durch (3.10) ausgedrückte Symmetrie der reziproken Verschaltung zwischen Ausgabe- und Zwischenschicht ist dabei unabhängig von der speziellen Wahl der Übertragungsfunktion $a_r(\mathbf{z}; \Theta) \equiv a_r(\mathbf{x}, \mathbf{y}; \Theta)$ der Zwischenschichtneuronen r .

Wählt man jedoch für diese Funktionen analog zu (2.59) und (2.60) die Partitionsfunktionen

$$a_r(\mathbf{z}; \tilde{\Theta}^c) = \frac{g(|\mathbf{z} - \mathbf{d}_r|, \sigma)}{A(\mathbf{z}; \tilde{\Theta}^c)} \quad (3.11)$$

mit der normierenden Gesamtaktivität

$$A(\mathbf{z}; \tilde{\Theta}^c) = \sum_{r=1}^M g(|\mathbf{z} - \mathbf{d}_r|, \sigma), \quad (3.12)$$

so resultiert die stochastische Iteration der Lernregel (3.8) in einem RGF-*clustering* des Trainingsdatensatzes $\mathcal{Z} = \{\mathbf{z}_k \equiv (\mathbf{x}_k, \mathbf{y}_k) | k = 1 \dots N\}$ durch das Kodebuch $\tilde{\mathcal{C}}_\sigma = \{\mathbf{d}_r | r = 1 \dots M\}$. Wie in Abschnitt 2.4.2 erklärt wurde, maximiert dieser Lernprozeß für jedes σ die *likelihood*, daß der Datensatz \mathcal{Z} von der Modelldichte

$$\hat{p}(\mathbf{z} | \tilde{\Theta}^c) = \frac{1}{M(\sqrt{2\pi}\sigma)^{D+K}} A(\mathbf{z}; \tilde{\Theta}^c) \quad (3.13)$$

erzeugt wurde, wobei der Parametersatz $\tilde{\Theta}^c$ das stationäre Kodebuch \tilde{C}_σ und die Gaußbreite σ umfaßt. Wie dort ebenfalls erläutert wurde, erlaubt das Abkühlen des Temperaturparameters σ im *univar*-Verfahren die Bestimmung jenes Parametersatzes $\bar{\Theta}^c = \{\tilde{C}_{\sigma_f}, \sigma_f\}$, welcher die *likelihood* maximiert, daß \mathcal{Z} von einer solchen Mischungsdichte aus univariaten Normalverteilungen mit identischen statistischen Gewichten $P_r = 1/M$ und Varianzen $\sigma_r = \sigma_f$ erzeugt wurde.

Beachtet man nun noch die Symmetriebeziehung (3.10) so können die folgenden Resultate festgehalten werden:

- Auf Grund der Einführung reziproker Verbindungen können *alle* Netzwerkparameter $\Theta^{GRBF} = \{\bar{\Theta}^c, \mathbf{f}_r | r = 1 \dots M\}$ in einem *gemeinsamen* Lernprozeß aus \mathcal{Z} extrahiert werden. Separate Trainingsprozesse für die \mathbf{c}_r und \mathbf{f}_r sowie heuristische Prozeduren zur Bestimmung der Gaußbreiten, wie sie im ursprünglichen GRBF-Modell von J. Moody und C. Darken vorgesehen waren [MD88, MD89], entfallen damit.
- Die Funktion $A(\mathbf{z}; \bar{\Theta}^c)$, die zur Erfassung der normierenden Gesamtaktivität auf der Zwischenschicht des GRBF-Netzwerks während des Lernprozesses eingeführt wurde, hat nun eine neue statistische Bedeutung. Nach (3.13) stellt sie bis auf konstante Faktoren eine ML-Schätzung der *Verbundwahrscheinlichkeitsdichte* $\hat{p}(\mathbf{x}, \mathbf{y} | \bar{\Theta}^c) = \hat{p}(\mathbf{z} | \bar{\Theta}^c)$ der Trainingsdaten \mathcal{Z} dar. Wie schon mehrfach dargelegt wurde, hatten wir dagegen im ursprünglichen Modell aus Abbildung 2.17 eine Aktivierungsfunktion $A(\mathbf{x}; \Theta^c)$ verwendet, welche bis auf einen konstanten Faktor lediglich eine ML-Schätzung der Verteilung der Eingabedaten \mathcal{X} darstellte.
- Damit ist nun auch ein einheitliches Optimierungskriterium für alle GRBF-Parameter formuliert.

Mit diesen Resultaten ist eine erste Antwort auf die zweite Eingangsfragestellung nach der Beziehung zwischen GRBF-Netzwerken und Funktionenapproximationen (3.2) aus Schätzungen $\hat{p}(\mathbf{x}, \mathbf{y} | \bar{\Theta}^c)$ der Verbunddichte $p(\mathbf{x}, \mathbf{y})$ gegeben. Denn es hat sich gezeigt, daß alle Parameter einer solchen Funktionenapproximation aus einem Lernprozeß für ein GRBF-Netzwerk, das um rückwärtige Verbindungen erweitert wurde, abgeleitet werden können. Allerdings handelt es sich dabei bislang um ein Modell, bei dem $p(\mathbf{x}, \mathbf{y})$ durch eine Mischung *univariater* Normalverteilungen mit identischen Gewichten und Varianzen approximiert wird. Wegen der größeren Flexibilität von Mischungsdichten aus *multivariaten* Komponenten wären jedoch derartige Modelle vorzuziehen, insbesondere wenn sie auch noch durch adaptives *on-line* Lernen erzeugt werden können.

Wenden wir uns daher der Anschlußfrage zu, die auf eine Formulierung des EM-Abkühlverfahrens für eine Mischung multivariater Normalverteilungen als sequentiellem Algorithmus zielt. Für diese Diskussion kann unter Verwendung der Definitionen (3.3) und (3.4) die Notation vereinfacht und statt der ML-Schätzung einer gemeinsamen Dichte $p(\mathbf{x}, \mathbf{y})$ mit $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^D \times \mathbb{R}^K$ die Schätzung einer Dichte $p(\mathbf{z})$ mit $\mathbf{z} \in \mathbb{R}^{D+K}$ diskutiert werden.

Damit reduziert sich das Problem formal auf die in Abschnitt 2.3.3 im Detail erläuterte Frage der ML-Schätzung einer unbekanntes Dichte $p(\mathbf{x})$ mit einer durch (2.35) und (2.36) gegebenen Mischung aus M multivariaten Normalverteilungen.

Es sei angemerkt, daß ein solcher multivariater Ansatz auch für die Wahl der Übertragungsfunktion der Zwischenschichtneuronen im ursprünglichen Netzwerkmodell aus Abbildung 2.17 gewählt werden kann. Das resultierende *multivariate GRBF-Netzwerk* ist dann eine Verallgemeinerung des univariaten GRBF-Netzwerks (2.62), die sich von diesem lediglich in der Wahl der Übertragungsfunktion der Zwischenschichtneuronen unterscheidet. Diese Funktion $a_r(\mathbf{x}; \Theta^M)$ des multivariaten GRBF-Netzwerks ist analog zum Ausdruck (2.59) für die Übertragungsfunktion im univariaten Fall aufgebaut und ist durch

$$a_r(\mathbf{x}; \Theta^M) = \frac{P_r \hat{p}(\mathbf{x}|r, \Theta_r^{PCA})}{\sum_{r'} P_{r'} \hat{p}(\mathbf{x}|r', \Theta_{r'}^{PCA})} \quad (3.14)$$

gegeben. Der Unterschied besteht in der Wahl von multivariaten Verteilungen $\hat{p}(\mathbf{x}|r, \Theta_r^{PCA})$ nach (2.36), deren Kovarianzmatrizen Σ_r und statistischen Gewichte P_r für jedes Zwischenschichtneuron verschieden gewählt werden können. Ein Vergleich mit (2.42) zeigt, daß die Aktivierungsfunktionen des multivariaten GRBF-Netzwerks gerade die *a posteriori*-Wahrscheinlichkeiten $P(r|\mathbf{x}, \Theta^M)$ sind, d.h. es gilt

$$a_r(\mathbf{x}; \Theta^M) = P(r|\mathbf{x}, \Theta^M). \quad (3.15)$$

Ferner ist in dieser Formulierung die normierende Gesamtaktivität $A(\mathbf{x}; \Theta^M)$ mit der ML-Modellldichte (2.35) identisch, d.h.

$$A(\mathbf{x}; \Theta^M) = \hat{p}(\mathbf{x}|\Theta^M). \quad (3.16)$$

Zur Formulierung eines sequentiellen Algorithmus für das EM-Abkühlverfahren werden wir uns zunächst an diesem Konzept eines multivariaten GRBF-Netzwerks orientieren und es dazu verwenden, den Parametern Θ^M der ML-Dichteschätzung $\hat{p}(\mathbf{x}|\Theta^M)$ Netzwerk-korrelate zuzuordnen. Diese Zuordnung wird aus der Hoffnung gespeist, daß eine solche neuronale Interpretation Hinweise auf die zu wählenden Lernalgorithmen liefern kann.

3.2. Lernregeln für das multivariate GRBF-Netzwerk

Am Ende des Abschnitts 2.3.3 wurde dargestellt, wie M. Kloppenburg und P. Tavan mit einem *maximum likelihood*-Ansatz zur Bestimmung der Parameter $\Theta^M = \{\mathbf{c}_r, \sigma_{ir}, \mathbf{w}_{ir}, P_r | i = 1 \dots D, r = 1 \dots M\}$ zu einem EM-Algorithmus kommen, der durch die Anwendung eines deterministischen Abkühlverfahrens auch sicher konvergiert [KT97]. Das Ziel dieses Abschnitts ist die Entwicklung biologisch plausibler, sequentieller Lernregeln für ein multivariates GRBF-Netzwerk mit den Aktivitäten (3.14).

Lernregeln für die \mathbf{c}_r

Für die Dendritenbäume \mathbf{c}_r ist die Antwort auf diese Frage offensichtlich. Ersetzt man in der Lernregel (2.20) für das RGF-clustering die Zuordnungsfunktion $P(r|\mathbf{x}, \mathcal{C}, \sigma)$ durch die Aktivierungsfunktion aus (3.14), so ist eine geeignete Lernregel definiert. Denn man kann sich leicht davon überzeugen, daß die stationären Punkte dieser sequentiellen Lernregel und der Fixpunktiteration (2.37) identisch sind. Daher fehlen lediglich noch Lernregeln der Form

$$\mathbf{w}_{ir}^{\text{neu}} = \mathbf{w}_{ir}^{\text{alt}} + \epsilon_w \Delta \mathbf{w}_{ir}(\mathbf{x}_k) \quad (3.17)$$

$$\sigma_{ir}^{2,\text{neu}} = \sigma_{ir}^{2,\text{alt}} + \epsilon_\sigma \Delta \sigma_{ir}^2(\mathbf{x}_k) \quad (3.18)$$

$$P_r^{\text{neu}} = P_r^{\text{alt}} + \epsilon_P \Delta P_r(\mathbf{x}_k) \quad (3.19)$$

für die Eigenvektoren \mathbf{w}_{ir} , die Eigenwerte σ_{ir}^2 und die statistischen Gewichte P_r , welche die Gleichungen (2.44)-(2.46) der EM-Fixpunktiteration ersetzen können.

Zur Ableitung solcher Regeln wollen wir zunächst versuchen, die Parameter des multivariaten GRBF-Netzwerks neurobiologisch zu interpretieren. Analog zur Interpretation der univariaten Gaußfunktion in (2.59) als Formfunktion eines rezeptiven Feldes eines Neurons auf der Zwischenschicht, kann auch der Ausdruck (2.36) für die multivariate Normalverteilung $\hat{p}(\mathbf{x}|r, \Theta_r^{PCA})$ als derartige Formfunktion gedeutet werden. Die Ausrichtung eines solchen rezeptiven Feldes ist durch den Satz orthonormaler Vektoren \mathbf{w}_{ir} , $i = 1 \dots D$, definiert, der die lokale Kovarianzmatrix Σ_r diagonalisiert. Werden die Parameter $\{\mathbf{c}_r, \mathbf{w}_{ir}\}$ zu θ_{ir} zusammengefaßt, so kann die multivariate Normalverteilung (2.36) auch als Produkt der eindimensionalen Normalverteilungen

$$\hat{p}(s_{ir}(\mathbf{x}; \theta_{ir})) = \frac{\exp[-s_{ir}^2((\mathbf{x}; \theta_{ir})/2)]}{(2\pi\sigma_{ir}^2)^{1/2}} \quad (3.20)$$

geschrieben werden, wobei die Argumente $s_{ir}(\mathbf{x}; \theta_{ir})$ durch

$$s_{ir}(\mathbf{x}; \theta_{ir}) \equiv \sigma_{ir}^{-1} \mathbf{w}_{ir}^T (\mathbf{x} - \mathbf{c}_r) \quad (3.21)$$

gegeben sind. Dies ist die Projektion der Differenz zwischen dem Datenpunkt \mathbf{x} und dem Zentrum \mathbf{c}_r eines rezeptiven Feldes auf den Vektor $\sigma_{ir}^{-1} \mathbf{w}_{ir} \in \mathbb{R}^D$. Derartige Projektionen von Eingabevektoren auf feste Richtungen \mathbf{w}_{ir} werden in der Neuroinformatik üblicherweise als Merkmalsextraktionen bezeichnet. Entsprechend beschreibt jeder dieser Vektoren einen Merkmalsdetektor, dessen Richtung im Merkmalsraum durch \mathbf{w}_{ir} und dessen Gewichtung durch σ_{ir}^{-1} gegeben sind.

Lernregel für die Orientierungen \mathbf{w}_{ir}

In den letzten Jahren war die Erforschung von Lernregeln für die Selbstorganisation von Merkmalsdetektoren ein wichtiges Thema in der Neuroinformatik [HSW89]. 1988 wurde

beispielsweise von J. Rubner und P. Tavan dazu ein zweischichtiges Netzwerk vorgeschlagen [RT89]. Die Gewichte \mathbf{w}_i , $i = 1 \dots D$, zwischen Ein- und Ausgabeschicht dieses Netzwerks werden mit einer Hebbischen Lernregel trainiert, wohingegen zusätzliche laterale Verbindungen zwischen den Neuronen der Ausgabeschicht mit einer anti-Hebbischen Lernregel gelernt werden. Dieses Netzwerk führt eine PCA der Verteilung der Eingangsdaten \mathcal{X} aus. Die Aktivitäten der Ausgabeneuronen werden dabei statistisch dekorreliert, die Gewichte $\mathbf{w}_i \in \mathbb{R}^D$ konvergieren zu den Eigenvektoren der Kovarianzmatrix der Eingangsdaten und die Gewichte der lateralen Verbindungen verschwinden. Der Eigenvektor \mathbf{w}_1 , der zum größten Eigenwert σ_1^2 gehört, wird als erstes berechnet, gefolgt von der orthogonalen Menge der restlichen Eigenvektoren in der Reihenfolge der entsprechenden Eigenwerte. Mit diesem Modell ist es möglich, psycho-physiologische Befunde der Farbwahrnehmung wie auch Strukturen der rezeptiven Felder von Merkmalsdetektoren im visuellen Kortex zu erklären [Rub89, RS90].

Es erscheint daher sinnvoll, die Lernregeln dieses Modells auf die Selbstorganisation der Orientierungen \mathbf{w}_{ir} der Merkmalsdetektoren (3.20) zu übertragen. Allerdings sind Anpassungen notwendig, da es sich nun nicht um eine globale PCA handelt, sondern um die Analyse der Richtungen $\mathbf{x} - \mathbf{c}_r$ durch einen lokalen Detektor. Zusätzlich werden die lateralen Verbindungen, welche zu einer Orthogonalisierung der \mathbf{w}_{ir} führen, vernachlässigt und durch das Schmidtsche Orthonormierungsverfahren ersetzt [Fis86], da sich dadurch die technische Umsetzung deutlich einfacher gestaltet. Eine geeignete sequentielle stochastische Lernregel ist dann

$$\Delta \mathbf{w}_{ir}(\mathbf{x}_k) = a_r(\mathbf{x}_k; \Theta^M) s_{ir}(\mathbf{x}_k; \theta_{ir}) (\mathbf{x}_k - \mathbf{c}_r) / \sigma_{ir}, \quad (3.22)$$

wobei jedem Lernschritt eine Orthonormierung der \mathbf{w}_{ir} folgt. (3.22) ist eine Hebbische Lernregel, wenn das Produkt $a_r(\mathbf{x}_k; \Theta^M) s_{ir}(\mathbf{x}_k; \theta_{ir}) / \sigma_{ir}$ mit der postsynaptischen Aktivität eines Neurons (r, i) assoziiert wird und die Komponenten von $\mathbf{x}_k - \mathbf{c}_r$ als die Aktivitäten von D präsynaptischen Neuronen verstanden werden. Durch die Renormierung der \mathbf{w}_{ir} nach jedem Lernschritt wird die obige Lernregel zu einer *kompetitiven* Hebbischen Lernregel [RT89].

Es bleibt zu bemerken, daß (3.22) in nichtlinearer Weise von allen Parametern Θ^M abhängt. Im Besonderen sind die Gewichte σ_{ir}^{-1} der lokalen Merkmalsdetektoren enthalten, für die noch Lernregeln angegeben werden müssen. Um eine solche Lernregel ableiten zu können, wird zunächst die Stationaritätsbedingung für die \mathbf{w}_{ir} betrachtet, die der ständigen Renormierung dieser Vektoren während des stochastischen Lernprozesses (3.22) Rechnung trägt. Sie lautet

$$\langle \Delta \mathbf{w}_{ir} \rangle_{\mathcal{X}} = \lambda_{ir} \mathbf{w}_{ir} \quad (3.23)$$

mit skalaren Parametern λ_{ir} . Setzt man nun die Definition (3.21) in (3.22) und das obige Resultat in die obige Stationaritätsbedingung ein, so erhält man unter Verwendung der Definition (2.40) der r -lokalen Kovarianzmatrizen \mathbf{C}_{r, Θ^M} sowie der Identität (3.15) nach

einer kurzen Rechnung die äquivalente Bedingung

$$\mathbf{C}_{r, \Theta^M} \mathbf{w}_{ir} = \lambda'_{ir} \mathbf{w}_{ir} \quad (3.24)$$

mit den neuen Parametern $\lambda'_{ir} = \lambda_{ir} \sigma_{ir}^2 / \langle a_r(\mathbf{x}; \Theta^M) \rangle_{\mathcal{X}}$. Damit wird die Lernregel (3.22) genau dann stationär, sobald die Orientierungen \mathbf{w}_{ir} zu Eigenvektoren der *lokalen* Kovarianzmatrizen (2.40) werden. Die sequentielle stochastische Lernregel (3.17) hat damit denselben stationären Punkt wie die Fixpunktiterationsgleichung (2.46). Hier wird auch der wesentliche Unterschied zum ursprünglichen Modell von J. Rubner und P. Tavan deutlich, welches nur die globale Kovarianzmatrix des gesamten Datensatzes diagonalisiert. Da die beiden Algorithmen ansonsten identisch sind, gilt die schnelle Konvergenz des ursprünglichen Verfahrens [HK92, RT89] auch für den modifizierten Algorithmus. Daher kann man annehmen, daß nach einem kurzen Relaxationsprozeß – unter der Voraussetzung kleiner Lernparameter und langsamer Abkühlung – die \mathbf{w}_{ir} während des gesamten Lernprozesses tatsächlich die Eigenvektoren der lokalen Kovarianzmatrizen sind.

Als Folge der lokalen Kovarianzanalyse durch die Merkmalsdetektoren werden die Projektionen (3.21) lokal dekorreliert. Dies kann durch

$$\langle s_{ir}(\mathbf{x}; \theta_{ir}) s_{jr}(\mathbf{x}; \theta_{jr}) \rangle_{r, \Theta^M} = \frac{\lambda'_{ir}}{\sigma_{ir} \sigma_{jr}} \delta_{ij} \quad (3.25)$$

ausdrückt werden und folgt aus den Definitionen (3.21), (2.41) sowie der soeben begründeten Annahme, daß die \mathbf{w}_{ir} das Eigenwertproblem (3.24) lösen.

Lernregel für die Gewichte σ_{ir}^{-1}

Es wird nun verlangt, daß die Projektionen (3.21) auf die lokalen Merkmalsvektoren die Eigenschaft

$$\langle s_{ir}^2(\mathbf{x}; \theta_{ir}) \rangle_{r, \Theta^M} = 1 \quad (3.26)$$

haben sollen, welche auch als *functional balance* bezeichnet werden soll. Mit (3.26) wird für jeden Merkmalsdetektor eine lokale Mahalanobi-Metrik eingeführt. Dies entspricht der Forderung, daß die verschiedenen Merkmalsdetektoren eine ausgewogene Rolle bei der Repräsentation der Daten spielen sollen. Aus (3.25) folgt nun, daß (3.26) gilt, wenn der Parameter σ_{ir}^2 mit dem Eigenwert λ'_{ir} übereinstimmt. Um diese Gleichheit und damit die *functional balance* zu garantieren, wird die sequentielle stochastische Lernregel

$$\Delta \sigma_{ir}^2(\mathbf{x}_k) = a_r(\mathbf{x}_k; \Theta^M) \sigma_{ir}^2 [s_{ir}^2(\mathbf{x}_k; \theta_{ir}) - 1] \quad (3.27)$$

eingeführt, deren stationärer Punkt durch $\sigma_{ir}^2 = \lambda'_{ir}$ gegeben ist. Am stationären Punkt ist der Erwartungswert der Lernregel (3.27) gerade gleich der Sattelpunktsgleichung (2.45).

Letzere wurde im ML-Algorithmus von M. Kloppenburg und P. Tavan durch (2.47) ersetzt, um ein Abkühlungsverfahren zu ermöglichen. Die Erweiterung der obigen Lernregel gemäß

$$\Delta\sigma_{ir}^2(\mathbf{x}_k) = a_r(\mathbf{x}_k; \Theta^M) \left(\sigma_{ir}^2 [s_{ir}^2(\mathbf{x}_k; \theta_{ir}) - 1] + \mu \frac{\sigma_f^2 - \sigma_{ir}^2}{\langle a_r \rangle} \right), \quad (3.28)$$

wobei $\langle a_r \rangle$ einen gleitenden Mittelwert der Aktivität $a_r(\mathbf{x}; \Theta^M)$ darstellt, hat denselben Effekt. Wie im genannten EM-Abkühlungsverfahren sind auch hier die σ_{ir}^2 bei großen Werten von μ stark an den Temperaturparameter σ_f gekoppelt. Bei Verringerung von μ nähern sich die σ_{ir}^2 immer mehr den Eigenwerten der lokalen Kovarianzmatrizen \mathbf{C}_{r, Θ^M} an.

Lernregel für die P_r

Nachdem nun Lernregeln für die \mathbf{c}_r , \mathbf{w}_{ir} und σ_{ir} entwickelt wurden, fehlen noch entsprechende Lernregeln für die statistischen Gewichte P_r in (3.14). Aus der Sicht der Dichteschätzung drückt die Eigenschaft der *load balance* beim *univar*-Algorithmus die Tatsache aus, daß die Gewichte P_r nicht als freie Parameter optimiert werden, sondern auf $1/M$ fixiert sind. Dies führt dazu, daß die verbleibenden Parameter dazu gezwungen werden, sich während des Abkühlvorgangs so anzupassen, daß die *load balance* so gut wie möglich erhalten wird. Führt man statt dieser Kompetition zwischen den einzelnen Komponenten der Mischungsdichte eine Lernregel für die P_r ein, so besteht die Gefahr, singuläre Lösungen ($P_r = 0$) zu erhalten. Eine stochastische Lernregel, die die EM-Sattelpunktgleichung (2.38) ersetzt, ist

$$\Delta P_r = a_r(\mathbf{x}_k; \Theta^M) - P_r. \quad (3.29)$$

Diese Regel beschreibt eine stochastisch ausgeführte Mittelung, die für $P_r = \langle a_r(\mathbf{x}; \Theta^M) \rangle_{\mathcal{X}}$ stationär wird [vgl. (2.38)]. Wenn während des Lernvorgangs ein Neuron r zwischenzeitlich nur schwach aktiviert ist, so wird das zugehörige statistische Gewicht P_r auf Grund von (3.29) abnehmen und wegen der Summennormierung der Aktivität (3.14) in der Folge noch weiter verkleinert. Um einen stabilen Verlauf des Lernprozesses zu ermöglichen, wird in dieser Arbeit daher diese Lernregel nicht verwendet und P_r bei $1/M$ fixiert.

Fazit

Wir haben mit (2.20), (3.22) und (3.28) sowie mit der Festsetzung $P_r = 1/M$ einen Satz sequentieller stochastischer Lernregeln für die Parameter $\Theta_r^{PCA} = \{\mathbf{c}_r, \mathbf{W}_r, \hat{\Sigma}_r\}$ formuliert, welche die Gleichungen (2.37), (2.48), (2.47) des EM-Abkühlverfahrens ersetzen. Im Verlauf des so definierten stochastischen Abkühlverfahrens spiegeln die Parameter der Komponenten $\hat{p}(\mathbf{x}|r, \Theta_r^{PCA})$ der Mischungsdichte in zunehmender Weise die lokale Statistik des Datensatzes \mathcal{X} in den durch $a_r(\mathbf{x}; \Theta^M)$ definierten Umgebungen der Zentren \mathbf{c}_r

wider. Auf Grund des Ansatzes bezeichnen wir dieses Verfahren, das eine lokale PCA des Datensatzes \mathcal{X} ausführt, auch als *multivar*-Algorithmus.

Der nächste Abschnitt geht näher auf algorithmische Details und auf die Wahl der Lern- und Abkühlparameter ein. Abschließend wird noch der Verlauf des *multivar*-Verfahrens an Hand der aus Abbildung 2.8 bekannten Daten demonstriert.

3.2.1. Durchführung des Lernverfahrens

Bei der Anwendung des *multivar*-Algorithmus zur Konstruktion eines Spracherkenners [Met98, Wil98] hat sich herausgestellt, daß es günstig ist, die Bestimmung der Parameter Θ^M in einen *univar*- und einen *multivar*-Schritt zu gliedern. Diese beiden Schritte werden im folgenden zusammengefaßt und in Abbildung 3.2 und 3.3 an einem Beispiel veranschaulicht.

- Der *univar*-Schritt basiert auf dem *univar*-Algorithmus (siehe Abschnitt 2.3.2) mit den Aktivitäten (2.16). Für sehr große σ sind alle Aktivitäten unabhängig von den Daten und gleich $1/M$. In diesem Fall gilt $\mathbf{c}_r = \langle \mathbf{x} \rangle_{\mathcal{X}}$. Die Initialisierung der Merkmalsdetektoren \mathbf{c}_r erfolgt also sinnvollerweise auf den Schwerpunkt des Datensatzes \mathcal{X} . In [Der95] wird gezeigt, daß die \mathbf{c}_r im Schwerpunkt verbleiben, bis σ unter einen kritischen Wert abgekühlt wird, der der Streuung $\sigma_{\mathcal{X}}$ des Datensatzes \mathcal{X} entspricht. Daher wird der Anfangswert σ_i auf knapp über $\sigma_{\mathcal{X}}$ gesetzt. Mit dieser Initialisierung wird die stochastische Lernregel (2.19) mit dem Lernschritt (2.20) wiederholt ausgeführt. Während des Abkühlens von σ auf einen kleinen Wert wird der Lernparameter ϵ_c üblicherweise linear von 0.9 auf etwa M/N verringert. Das Abkühlen erfolgt solange, bis die *log-likelihood* $l(\mathcal{X}; \Theta^c)$ maximal wird. Dieses Maximum bestimmt den Endwert σ_f und damit einen Parametersatz $\Theta^c = \{\sigma_f, \mathbf{c}_r | r = 1 \dots M\}$, welcher im Rahmen des gewählten Modells die optimale Schätzung des Datensatzes \mathcal{X} ist.
- Der anschließende *multivar*-Schritt wird mit Θ^c und σ_f initialisiert, d.h. die \mathbf{c}_r werden übernommen, die lokalen Kovarianzmatrizen auf $\hat{\Sigma}_r = \sigma_f \mathbf{I}$ und die statistischen Gewichte P_r auf $1/M$ gesetzt¹. Anschließend werden die stochastischen, sequentiellen Lernregeln (2.19), (3.17) und (3.18) mit den Lernschritten (2.20), (3.22), (3.28) und den multivariaten Aktivitäten $a_r(\mathbf{x}_k; \Theta^M)$ ausgeführt. Um einen langsamen Übergang vom *univar*-Schritt zu erreichen, wird der Kopplungsparameter μ zunächst mit einem sehr großen Wert belegt. Im Laufe des *multivar*-Verfahrens wird μ auf den Wert Null verringert, so daß die Ausdehnungen σ_{i_r} der einzelnen multivariaten Schätzungen nicht mehr an σ_f gekoppelt sind. Die Lernparameter ϵ_c , ϵ_w und

¹ Die Parameter entsprechen nun genau dem *univar*-Fall. Man könnte also den *univar*-Schritt auch mit den Lernschritten (2.20), (3.22), (3.28) und einer großer Kopplung μ berechnen. Dagegen sprechen aber die aufwendige Auswertung der multivariaten Normalverteilung (2.36) und die beiden wegen der starken Kopplung an σ überflüssigen Lernschritte (3.22) und (3.28).

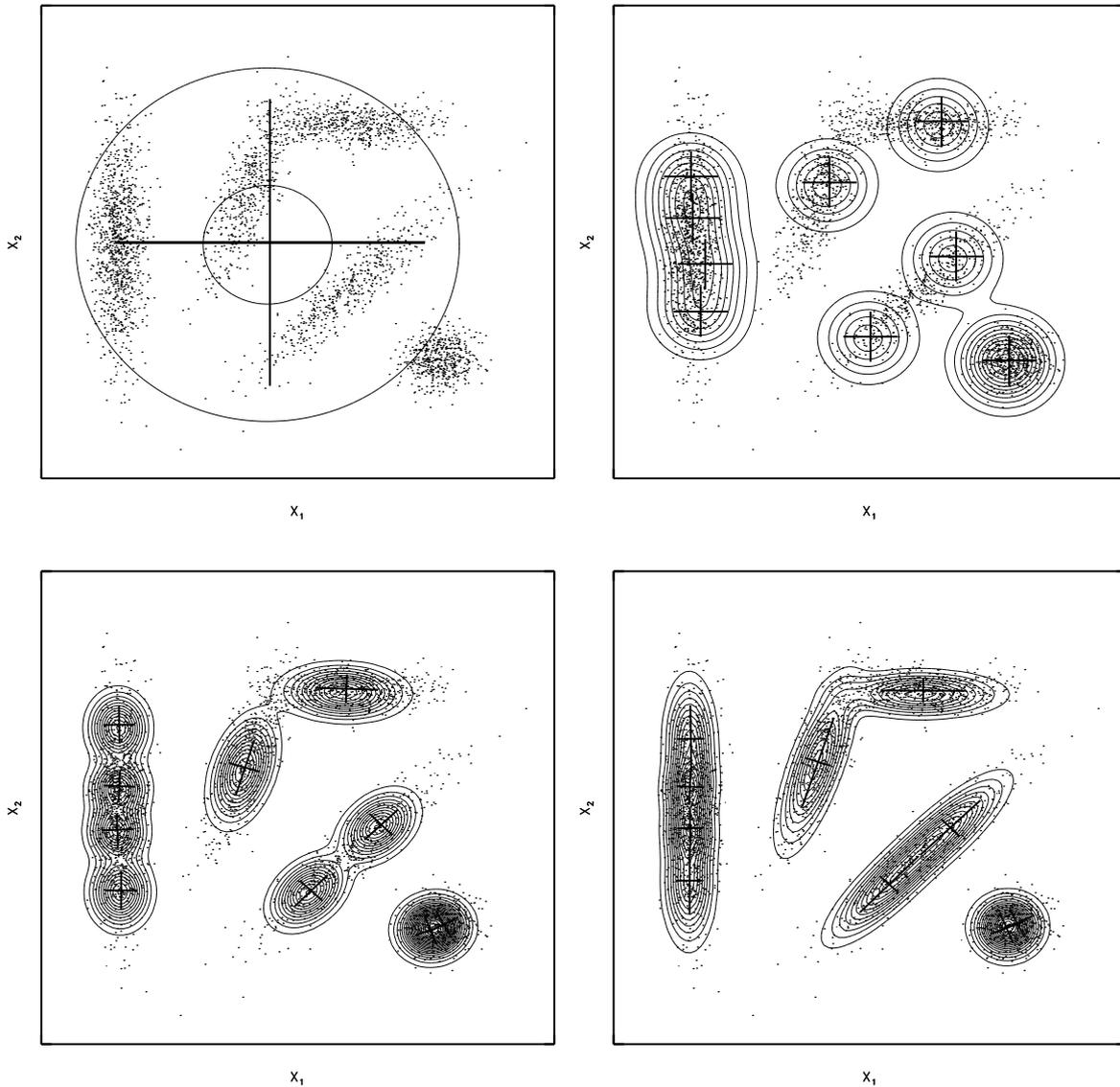


Abbildung 3.2.: **Dichteschätzung mit dem *multivar*-Algorithmus.** Die Datenverteilung aus Abbildung 2.8 wird hier für 10 Neuronen mit dem *multivar*-Algorithmus geschätzt. Die Dichten $\hat{p}(\mathbf{x}|\Theta^c)$ bzw. $\hat{p}(\mathbf{x}|\Theta^M)$ werden durch Höhenlinien dargestellt. Das Koordinatenkreuz zeigt die Ausrichtung der Eigenvektoren \mathbf{w}_{i_r} und die Länge der Achsen entspricht der Standardabweichung σ_{i_r} . Links oben ist der Initialisierungszustand für den *univar*-Schritt gezeigt. Alle lokalen Merkmalsvektoren \mathbf{c}_r sind auf den Mittelwert der Daten zentriert. σ_i ist auf einen Wert knapp über $\sigma_{\mathcal{X}}$ gesetzt. Rechts oben ist das Ergebnis des *univar*-Schritts gezeigt und rechts unten das Ergebnis eines darauf aufbauenden *multivar*-Schritts. Links unten ist ein beliebig herausgegriffener Zustand während des *multivar*-Schritts dargestellt.

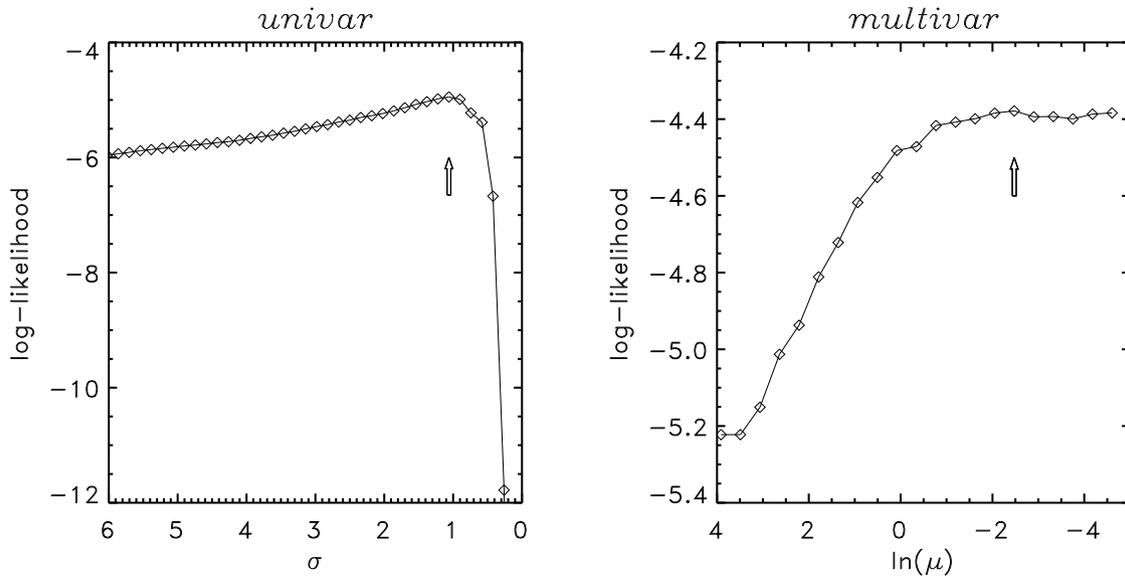


Abbildung 3.3.: **Die *log-likelihood* als Abbruchkriterium.** Links ist die *log-likelihood* für den Datensatz aus Abbildung 3.2 im Verlauf der Abkühlung von σ gezeigt. Das Maximum der *log-likelihood* (Pfeil) definiert den Endwert σ_f und einen Parametersatz Θ^c , der in Abbildung 3.2, rechts oben dargestellt ist. Das *multivar*-Verfahren führt zu einer weiteren Vergrößerung der *log-likelihood*. Der Pfeil kennzeichnet denjenigen Moment des Lernvorgangs, an dem der Parametersatzes Θ^M optimal ist. Θ^M ist auch in Abbildung 3.2, rechts unten, dargestellt.

ϵ_σ werden dabei wiederum von 0.9 auf etwa M/N verringert. Im *multivar*-Schritt können sich daher die Positionen \mathbf{c}_r weiter verändern und es findet gleichzeitig eine Anpassung der einzelnen Komponenten r an die lokale Form der Verteilung statt. Der Algorithmus wird beendet, sobald die *log-likelihood* (2.44) bei einem möglichst großen Wert stationär geworden ist.

Abbildung 3.2 zeigt den Ablauf einer Dichteschätzung der bekannten zweidimensionalen Daten aus Abbildung 2.8 mit dem *multivar*-Algorithmus. Die obere Zeile zeigt die Situation bei der Initialisierung und das Ergebnis des *univar*-Schritts. In der unteren Zeile ist ein Zwischenzustand und das Resultat des *multivar*-Schritts dargestellt. Der *univar*-Schritt liefert schon ein recht gutes Ergebnis für $\hat{p}(\mathbf{x}|\Theta^c)$. Die Struktur des Datensatzes wird erkannt, allerdings kann die Ausrichtung der lokalen Verteilungen nicht wiedergegeben werden. Dies ist erst nach dem *multivar*-Schritt möglich, welcher zu einer Schätzung $\hat{p}(\mathbf{x}|\Theta^M)$ führt, die relativ gut mit der ursprünglichen Verteilung übereinstimmt. Der Verlauf der *log-likelihood* während der *univar*- bzw. *multivar*-Schritte wird in Abbildung 3.3 dargestellt, welche die Zunahme der Qualität der Dichteschätzung im Verlauf des Lernprozesses dokumentiert.

3.3. Funktionenapproximation

Nachdem nun geklärt ist, daß mit den iterativen Lernregeln und Abkühlverfahren des *multivar*-Algorithmus ein zuverlässiges Instrument zur ML-Dichteschätzung eines Datensatzes durch eine Mischung (2.35) zur Verfügung steht, bleibt noch zu untersuchen, welche Auswirkungen multivariat erweiterte Übertragungsfunktionen (3.14) auf die Eigenschaften eines GRBF-Netzwerkes haben. Ferner ist noch unklar, ob und wie die von mir vorgeschlagene Alternative (3.2) zur Funktionenapproximation, in die als wesentliches Element eine Schätzung der Verbundwahrscheinlichkeitsdichte $p(\mathbf{x}, \mathbf{y})$ des Datensatzes \mathcal{Z} eingeht, mit der Approximation (2.62) von Funktionen durch multivariate GRBF-Netzwerke zusammenhängt. Zwar haben wir schon gesehen, daß das Training eines um rückwärtige Verbindungen erweiterten univariaten GRBF-Netzwerks mittels des *univar*-Algorithmus dazu führt, daß (i) alle Parameter eines solchen Netzes gleichzeitig optimiert werden können und (ii) die resultierende Gesamtaktivität der Zwischenschicht $A(\mathbf{x}, \mathbf{y}; \bar{\Theta}^c)$ nach (3.13) eine ML-Schätzung des Datensatzes \mathcal{Z} durch eine univariate Mischungsdichte $\hat{p}(\mathbf{x}, \mathbf{y} | \bar{\Theta}^c)$ darstellt. Aber damit sind die Konsequenzen der Verallgemeinerung auf den multivariaten Fall noch nicht gezogen und ein Zusammenhang zur Alternative (3.2) ist noch nicht sichtbar. Es scheint lediglich plausibel, daß ein solcher Zusammenhang existieren könnte.

Um diesen Mißstand zu beseitigen, soll nun der Ansatz (3.2) näher erkundet werden. Dazu wird angenommen, daß für den Datensatz \mathcal{Z} mittels des *multivar*-Algorithmus eine ML-Dichteschätzung durchgeführt worden sei. Die resultierende Modellldichte

$$\hat{p}(\mathbf{x}, \mathbf{y} | \Theta^M) = \frac{1}{M} \sum_{r=1}^M \hat{p}(\mathbf{x}, \mathbf{y} | r, \Theta_r^{PCA}) \quad (3.30)$$

ist dann mit (3.3) analog zu (2.35) eine Mischung aus multivariaten Normalverteilungen

$$\hat{p}(\mathbf{z} = (\mathbf{x}, \mathbf{y}) | r, \Theta_r^{PCA}) = \frac{\exp[-(\mathbf{z} - \mathbf{d}_r)^T \Sigma_r^{-1} (\mathbf{z} - \mathbf{d}_r) / 2]}{(2\pi)^{(D+K)/2} (\det \Sigma_r)^{1/2}} \quad (3.31)$$

mit identischen Gewichten und kann analog zu (3.13) als normierende Gesamtaktivität $A(\mathbf{x}, \mathbf{y}; \Theta^M)$ eines multivariaten, um rückwärtige Verbindungen \mathbf{f}'_r erweiterten, GRBF-Netzwerks (vgl. Abbildung 3.1) aufgefaßt werden. Entsprechend kann die *multivar*-Optimierung der Dichteschätzung als Lernprozeß für die Parameter Θ^M eines solchen GRBF-Netzwerks interpretiert werden. Während des Lernens sind dann die Aktivierungsfunktionen der Zwischenschichtneuronen r analog zu (3.14) durch

$$a_r(\mathbf{x}, \mathbf{y}; \Theta^M) = \frac{(1/M) \hat{p}(\mathbf{x}, \mathbf{y} | \Theta_r^{PCA})}{\hat{p}(\mathbf{x}, \mathbf{y} | \Theta^M)} \quad (3.32)$$

gegeben. Sie sind unscharfe Partitionsfunktionen im \mathbb{R}^{D+K} , deren Werte Zuordnungswahrscheinlichkeiten von Punkten (\mathbf{x}, \mathbf{y}) zu Zwischenschichtneuronen r angeben. Der Parametersatz $\Theta^M = \{\hat{\Sigma}_r, \mathbf{W}_r, \mathbf{c}_r, \mathbf{f}'_r | r = 1 \dots M\}$ beschreibt rezeptive Felder der Zwischenschichtneuronen im erweiterten Merkmalsraum \mathbb{R}^{D+K} , die um die Punkte $(\mathbf{c}_r, \mathbf{f}'_r)$ lokalisiert sind und deren Formen durch die in den Diagonalmatrizen $\hat{\Sigma}_r \in \mathbb{R}^{D+K} \times \mathbb{R}^{D+K}$

zusammengefaßten Varianzen und die in den orthogonalen Matrizen $\mathbf{W}_r \in \mathbb{R}^{D+K} \times \mathbb{R}^{D+K}$ zusammengefaßten Orientierungen beschrieben werden. Verwendet man für das Training der Axonbäume \mathbf{f}_r die Regel (3.9), so erhält man nach den Argumenten aus Abschnitt 3.1 die Symmetriebeziehung (3.10). Daher kann auf ein gesondertes Training der \mathbf{f}_r verzichtet werden und wir setzen von nun an

$$\mathbf{f}_r \equiv \mathbf{f}'_r. \quad (3.33)$$

Als Resultat der *multivar*-Schätzung der gemeinsamen Verteilung der Ein- und Ausgabedaten erhalten wir, wie wir bei genauerem Hinsehen sogleich feststellen werden, eine größere Parameterzahl als dies für ein multivariates GRBF-Netzwerk zur Funktionenapproximation eigentlich nötig wäre. Denn bei seinem Einsatz als Funktionenapproximator nach (2.62) wird lediglich der Eingabevektor $\mathbf{x} \in \mathbb{R}^D$ vorgegeben, während die Sollausgabe \mathbf{y} fehlt, so daß es wenig sinnvoll ist, zu diesem Zeitpunkt die im \mathbb{R}^{D+K} definierte Aktivierungsfunktion (3.32) zu verwenden. Stattdessen muß eine auf den \mathbb{R}^D eingeschränkte Aktivierungsfunktion (3.14) verwendet werden, in die entsprechend auf den \mathbb{R}^D eingeschränkte multivariate Normalverteilungen eingehen. Deren Kovarianzmatrizen haben aber eine sehr viel kleinere Dimension als jene Kovarianzmatrizen Σ_r , die aus der *multivar*-Schätzung von $p(\mathbf{x}, \mathbf{y})$ bestimmt worden sind.

Um die damit geforderte Reduktion der Parameterzahl zu bewerkstelligen, betrachten wir nun den Ansatz (3.2) zur Funktionenapproximation, der durch

$$\hat{\mathbf{f}}(\mathbf{x}; \Theta^M) = \frac{\int_{\mathbb{R}^K} \mathbf{y} \hat{p}(\mathbf{x}, \mathbf{y} | \Theta^M) d\mathbf{y}}{\int_{\mathbb{R}^K} \hat{p}(\mathbf{x}, \mathbf{y} | \Theta^M) d\mathbf{y}} \quad (3.34)$$

gegeben ist, genauer.

Zur Auswertung der Integrale (3.34) werden die Matrizen $\hat{\Sigma}_r$ und \mathbf{W}_r in Blockmatrizen zerlegt, die die Unterräume \mathbb{R}^D und \mathbb{R}^K betreffen. Wir definieren daher

$$\begin{pmatrix} \mathbf{A}_r & \mathbf{B}_r \\ \mathbf{C}_r & \mathbf{D}_r \end{pmatrix} \equiv \mathbf{W}_r \quad (3.35)$$

und

$$\begin{pmatrix} \hat{\Sigma}_{r,1}^{-1} & 0 \\ 0 & \hat{\Sigma}_{r,2}^{-1} \end{pmatrix} \equiv \hat{\Sigma}_r^{-1} \quad (3.36)$$

mit $\mathbf{A}_r \in \mathbb{R}^D \times \mathbb{R}^D$, $\mathbf{B}_r \in \mathbb{R}^K \times \mathbb{R}^K$, $\mathbf{C}_r \in \mathbb{R}^K \times \mathbb{R}^D$ und $\mathbf{D}_r \in \mathbb{R}^D \times \mathbb{R}^K$. Mit (3.35) und (3.36) kann die inverse Kovarianzmatrix Σ_r^{-1} in der Exponentialfunktion der r -ten Normalverteilung (3.31) auch durch

$$\begin{pmatrix} \boldsymbol{\chi}_r^{-1} & \boldsymbol{\Omega}_r^{-1} \\ (\boldsymbol{\Omega}_r^{-1})^T & \boldsymbol{\Phi}_r^{-1} \end{pmatrix} \equiv \Sigma_r^{-1} = \mathbf{W}_r \hat{\Sigma}_r^{-1} \mathbf{W}_r^T \quad (3.37)$$

mit

$$\boldsymbol{\chi}_r^{-1} = (\boldsymbol{\chi}_r^{-1})^T \equiv \mathbf{A}_r \hat{\boldsymbol{\Sigma}}_{r,1}^{-1} \mathbf{A}_r^T + \mathbf{B}_r \hat{\boldsymbol{\Sigma}}_{r,2}^{-1} \mathbf{B}_r^T \quad (3.38)$$

$$\boldsymbol{\Phi}_r^{-1} = (\boldsymbol{\Phi}_r^{-1})^T \equiv \mathbf{C}_r \hat{\boldsymbol{\Sigma}}_{r,1}^{-1} \mathbf{C}_r^T + \mathbf{D}_r \hat{\boldsymbol{\Sigma}}_{r,2}^{-1} \mathbf{D}_r^T \quad (3.39)$$

$$\boldsymbol{\Omega}_r^{-1} \equiv \mathbf{A}_r \hat{\boldsymbol{\Sigma}}_{r,1}^{-1} \mathbf{C}_r^T + \mathbf{B}_r \hat{\boldsymbol{\Sigma}}_{r,2}^{-1} \mathbf{D}_r^T \quad (3.40)$$

ausgedrückt werden.

Die beiden Integrale aus (3.34) werden in Anhang A.1.1 berechnet. Als Ergebnis erhält man eine Linearkombination

$$\hat{\mathbf{f}}(\mathbf{x}; \Theta^{LLM}) = \sum_{r=1}^M \tilde{\mathbf{f}}_r(\mathbf{x}; \Theta_r^{LLM}) \tilde{a}_r(\mathbf{x}; \Theta^{LLM}) \quad (3.41)$$

der affinen Funktionen

$$\tilde{\mathbf{f}}_r(\mathbf{x}; \Theta_r^{LLM}) \equiv \mathbf{f}_r + \tilde{\boldsymbol{\Phi}}_r(\mathbf{x} - \mathbf{c}_r) \quad (3.42)$$

mit den ‘‘Aktivierungsfunktionen’’

$$\tilde{a}_r(\mathbf{x}; \Theta^{LLM}) \equiv \frac{\tilde{p}(\mathbf{x}|r, \Theta_r^{LLM})}{\sum_{r'=1}^M \tilde{p}(\mathbf{x}|r', \Theta_{r'}^{LLM})}, \quad (3.43)$$

die durch die multivariaten Normalverteilungen

$$\tilde{p}(\mathbf{x}|r, \Theta_r^{LLM}) \equiv \frac{\exp[-(\mathbf{x} - \mathbf{c}_r)^T \tilde{\boldsymbol{\chi}}_r^{-1}(\mathbf{x} - \mathbf{c}_r)/2]}{(2\pi)^{D/2} (\det \tilde{\boldsymbol{\chi}}_r)^{1/2}} \quad (3.44)$$

gegeben sind. Dabei sind die Projektionsmatrizen $\tilde{\boldsymbol{\Phi}}_r$ in (3.42) durch

$$\tilde{\boldsymbol{\Phi}}_r \equiv -\boldsymbol{\Phi}_r(\boldsymbol{\Omega}_r^{-1})^T \quad (3.45)$$

und die metrischen Matrizen $\tilde{\boldsymbol{\chi}}_r^{-1}$ in (3.44) durch

$$\tilde{\boldsymbol{\chi}}_r^{-1} \equiv \boldsymbol{\chi}_r^{-1} - \boldsymbol{\Omega}_r^{-1} \boldsymbol{\Phi}_r(\boldsymbol{\Omega}_r^{-1})^T \quad (3.46)$$

definiert.

Die Funktionenapproximation $\hat{\mathbf{f}}(\mathbf{x}; \Theta^{LLM})$ ist durch einen Parametersatz Θ^{LLM} bestimmt, der sich aus den M Parametersätzen $\Theta_r^{LLM} = \{\boldsymbol{\chi}_r, \boldsymbol{\Phi}_r, (\boldsymbol{\Omega}_r^{-1})^T, \mathbf{c}_r, \mathbf{f}_r, P_r\}$ zusammensetzt. Die $\boldsymbol{\Phi}_r$ sind durch Inversion der regulären Matrizen $\boldsymbol{\Phi}_r^{-1}$ zu berechnen. Alle übrigen Parameter können direkt aus den Parametern Θ^M der multivariaten ML-Dichteschätzung (3.30) des Datensatzes \mathcal{Z} abgelesen werden.

Die Funktionenapproximation (3.41) zeigt eine Reihe von Gemeinsamkeiten mit einem GRBF-Netzwerk:

- Wie beim GRBF-Netzwerk ist die zur Funktionenapproximation verwendete Modellfunktion (3.41) eine Linearkombination, in welche die Aktivierungsfunktionen $\tilde{a}_r(\mathbf{x}; \Theta^{LLM})$ eingehen.
- Auch hier bilden die Aktivierungsfunktionen (3.43) auf Grund der Eigenschaft

$$\sum_{r=1}^M \tilde{a}_r(\mathbf{x}; \Theta^{LLM}) = 1 \quad (3.47)$$

eine unscharfe Partitionierung des Eingaberaums \mathbb{R}^D .

- Wie beim multivariaten GRBF-Netzwerk sind die Funktionen $\tilde{a}_r(\mathbf{x}; \Theta^{LLM})$ aus multivariaten Normalverteilungen $\tilde{p}(\mathbf{x}|r, \Theta_r^{LLM})$ aufgebaut. Deren Lokalisierung um die Zentren \mathbf{c}_r sorgt dafür, daß es sich auch bei den $\tilde{a}_r(\mathbf{x}; \Theta^{LLM})$ um lokalisierte Funktionen handelt.
- Der konstante Term \mathbf{f}_r in der affinen Funktion (3.42) entspricht dem Axonbaum in einem GRBF-Netzwerk.

Der einzige Unterschied zu einem GRBF-Netzwerk zeigt sich im linearen Term $\tilde{\Phi}_r(\mathbf{x} - \mathbf{c}_r)$ der Funktion (3.42). Dieser findet keine Entsprechung in einem GRBF-Netzwerk und läßt sich auch nicht zwanglos durch eine neurobiologische Beobachtung interpretieren. Für den Fall, daß die Kopplungsmatrizen Ω_r^{-1} zu Null werden und somit diese linearen Terme verschwinden [vgl. (3.45)], ist aber die Funktionenapproximation (3.41) tatsächlich ein multivariates GRBF-Netzwerk. Die Funktionenapproximation (3.41) ist also dem GRBF-Netzwerk zwar formal ähnlich, kann diesem Netzwerktyp im allgemeinen aber nicht zugeordnet werden. Da $\hat{\mathbf{f}}(\mathbf{x}; \Theta^{LLM})$ eine Funktionenapproximation durch eine Linearkombination der *lokalen linearen Modelle* $\tilde{\mathbf{f}}_r(\mathbf{x}; \Theta_r^{LLM})$ vornimmt, wird im folgenden (3.41) als LLM-Approximator bezeichnet.

Im nächsten Abschnitt werden die Eigenschaften des LLM-Approximators für verschiedene Spezialfälle diskutiert und an Hand der bereits bekannten Beispieldaten aus Abbildung 2.6 und 2.8 illustriert.

3.4. Diskussion und Beispiele

In diesem Abschnitt sollen vor allem die durch (3.45) und (3.46) definierten Projektionsmatrizen $\tilde{\Phi}_r$ und metrischen Matrizen $\tilde{\chi}_r^{-1}$ näher beleuchtet werden. Zu diesem Zweck wird der LLM-Approximator (3.41) für die drei Spezialfälle untersucht, daß (i) die ML-Schätzung der Verbunddichte $\hat{p}(\mathbf{x}, \mathbf{y}|\Theta^M)$ durch eine Mischung univariater Normalverteilungen mit gleichen Varianzen σ^2 erfolgt, daß (ii) der LLM-Approximator einen zweidimensionalen Datensatz \mathcal{Z} modelliert, und daß (iii) dem LLM-Approximator die Aufgabe der Klassifikation eines Datensatzes \mathcal{X} nach verschiedenen Klassen zugewiesen wird.

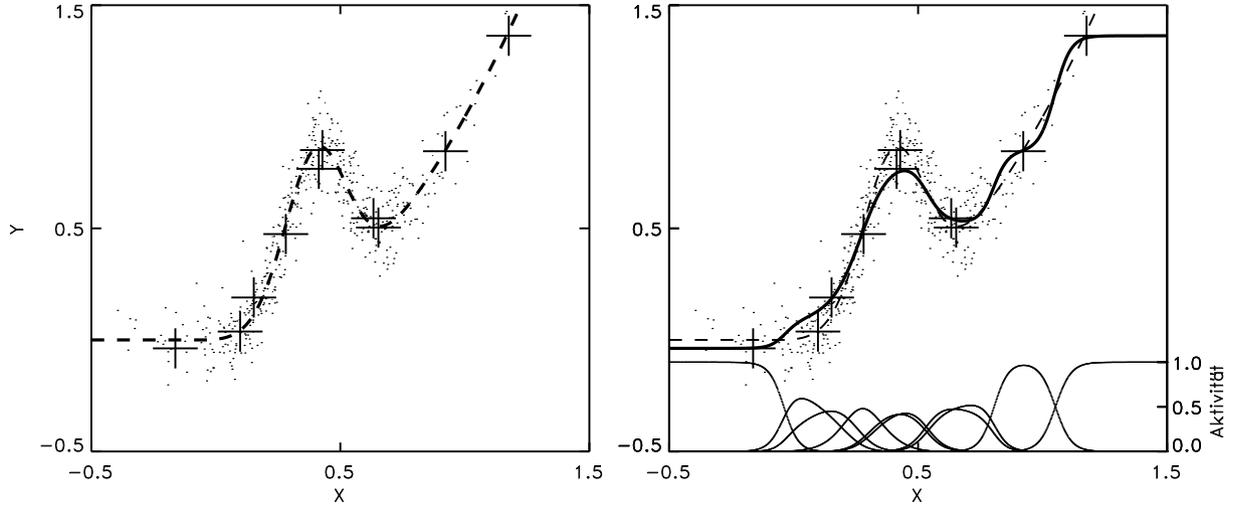


Abbildung 3.4.: **Der LLM-Approximator als univariates GRBF-Netzwerk.** Links ist das Ergebnis einer ML-Dichteschätzung der Daten $\mathcal{Z} = \{(x_k, y_k) | k = 1 \dots N\}$ durch eine Mischung univariater Normalverteilungen mit gleichen Varianzen σ^2 dargestellt. Die gestrichelte Linie ist die ursprüngliche Funktion $f(x)$, die approximiert werden soll. Rechts ist die Approximation $\hat{f}(x; \Theta^{LLM})$ mit dem aus der ML-Dichteschätzung resultierenden LLM-Approximator gezeigt (dicke durchgezogene Linie). Unten ist zusätzlich der Verlauf der Partitionierungsfunktionen $\tilde{a}_r(x; \Theta^{LLM})$ für die lokalen linearen Modelle $\tilde{f}_r(x; \Theta^{LLM}) = f_r$ eingetragen.

LLM-Approximator aus einer ML-Mischungsdichte univariater Normalverteilungen

Für den Fall, daß der LLM-Approximator aus einer ML-Mischungsdichte $\hat{p}(\mathbf{x}, \mathbf{y} | \Theta^M)$ aus univariaten Normalverteilungen mit gleichen Varianzen σ^2 abgeleitet ist, werden die inversen lokalen Kovarianzmatrizen Σ_r^{-1} in (3.37) diagonal. Da nun die Kopplungsmatrizen Ω_r^{-1} verschwinden, werden die metrischen Matrizen $\tilde{\chi}_r^{-1}$ der Normalverteilung (3.44) ebenfalls diagonal und die Projektionsmatrizen $\tilde{\Phi}_r$ zu Null [vgl. (3.45) und (3.46)], d.h.

$$\tilde{\chi}_r^{-1} = \frac{1}{\sigma_f^2} \mathbf{I} \quad \text{und} \quad \tilde{\Phi}_r^{-1} = \mathbf{0}. \quad (3.48)$$

Die Ausdehnungen der Normalverteilungen $\tilde{p}(\mathbf{x} | r, \Theta_r^{LLM})$ skalieren jetzt nur noch mit σ_f und die lokalen linearen Modelle (3.42) werden zu konstanten Modellen \mathbf{f}_r . Man erhält daher für den LLM-Approximator den Ausdruck

$$\hat{\mathbf{f}}(\mathbf{x}; \Theta^{LLM}) = \sum_{r=1}^M \mathbf{f}_r \tilde{a}_r(\mathbf{x}; \Theta^{LLM}) \quad (3.49)$$

mit den Aktivierungsfunktionen

$$\tilde{a}_r(\mathbf{x}; \Theta^{LLM}) = \frac{\exp(|\mathbf{x} - \mathbf{c}_r|^2/2\sigma^2)}{\sum_{r'=1}^M \exp(|\mathbf{x} - \mathbf{c}_{r'}|^2/2\sigma^2)}. \quad (3.50)$$

Dies ist genau das univariate GRBF-Netzwerk, das durch (2.59) und (2.62) definiert ist. Es ist aber zu beachten, daß die Parameter Θ^{LLM} des univariaten GRBF-Netzwerks (3.49) aus einer ML-Schätzung des gesamten Datensatzes \mathcal{Z} bestimmt werden, im Gegensatz zu den Parametern Θ^{GRBF} des univariaten GRBF-Netzwerks (2.62), die durch eine ML-Schätzung der Eingabedaten \mathcal{X} und die Anwendung der *outstar*-Regel (2.64) bestimmt werden.

Abbildung 3.4 zeigt die Arbeitsweise eines solchen LLM-Aproximators bzw. univariaten GRBF-Netzwerks an dem schon bekannten Beispiel aus Abbildung 2.6. Die Qualität der Approximierung ähnelt derjenigen des univariaten GRBF-Netzwerks aus Abbildung 2.20. Allerdings sind nun die Lagen der Zentren \mathbf{c}_r sowie die Werte \mathbf{f}_r der Ausgabegewichte durch eine ML-Schätzung der Verbunddichte $p(\mathbf{x}, \mathbf{y})$ festgelegt. Dies zeigt, daß der LLM-Aproximator (3.49) dem in Abbildung 3.1 skizzierten, um rückwärtige Verbindungen erweiterten GRBF-Netzwerk entspricht, dessen Lernprozeß schon in Abschnitt 3.1 diskutiert wurde.

LLM-Aproximator für einen zweidimensionalen Datensatz \mathcal{Z}

Für den Fall eines Datensatzes \mathcal{Z} , der sich aus eindimensionalen Ein- und Ausgabedaten zusammensetzt, führt der LLM-Aproximator die Abbildung $\hat{f}(x; \Theta^{LLM}) : \mathbb{R} \rightarrow \mathbb{R}$ aus. Nach den Definitionen (3.35) und (3.36) gilt dann für die Matrizen $\mathbf{W}_r \in \mathbb{R} \times \mathbb{R}$ der Eigenvektoren und für die inversen diagonalisierten Kovarianzmatrizen $\hat{\Sigma}_r^{-1} \in \mathbb{R} \times \mathbb{R}$

$$\mathbf{W}_r = \begin{pmatrix} A_r & B_r \\ C_r & D_r \end{pmatrix} \quad \text{und} \quad \hat{\Sigma}_r^{-1} = \begin{pmatrix} \sigma_{r,1}^{-2} & \\ 0 & \sigma_{r,2}^{-2} \end{pmatrix}. \quad (3.51)$$

Diese Matrizen erhält man direkt aus der ML-Schätzung der Daten \mathcal{Z} mit einer Mischung (3.30) bivariater Normalverteilungen. Mit (3.51) kann nach einer kurzen Rechnung und unter Ausnutzung der Orthonormalität der Eigenvektoren die Metrik $\tilde{\chi}_r^{-1}$ als

$$\tilde{\chi}_r^{-1} = \frac{1}{A_r^2 \sigma_{r,1}^2 + C_r^2 \sigma_{r,2}^2} \quad (3.52)$$

geschrieben werden. Mit analogen Argumenten werden die Projektionen $\tilde{\Phi}_r$, welche die Steigung $d\tilde{f}_r(x; \Theta^{LLM})/dx$ der lokalen linearen Modelle beschreiben, zu

$$\tilde{\Phi}_r = \frac{\sigma_{r,1}^2 A_r C_r + \sigma_{r,2}^2 B_r D_r}{\sigma_{r,1}^2 A_r^2 + \sigma_{r,2}^2 B_r^2}. \quad (3.53)$$

Ein einziges lokales lineares Modell

Wie wollen nun zunächst den Fall untersuchen, daß der LLM-Approximator nur auf genau einem lokalen linearen Modell beruht, d.h. $M = 1$. Für die folgenden Ausführungen wird daher der Index r weggelassen. Die Aktivierungsfunktion $\tilde{a}(x; \Theta^{LLM})$ ist nun auf Grund der Normierungseigenschaft (3.47) immer gleich Eins. Der LLM-Approximator wird daher zu der einfachen Geradengleichung

$$\tilde{f}(x; \Theta^{LLM}) = f + \frac{\sigma_1^2 AC + \sigma_2^2 BD}{\sigma_1^2 A^2 + \sigma_2^2 B^2} (x - c). \quad (3.54)$$

Diese Gleichung kann weiter analysiert werden, wenn man berücksichtigt, daß die Kovarianzmatrix einer zweidimensionalen Verteilung [DH73] einerseits durch

$$\Sigma = \begin{pmatrix} \langle x^2 \rangle - \langle x \rangle^2 & \langle xy \rangle - \langle x \rangle \langle y \rangle \\ \langle xy \rangle - \langle x \rangle \langle y \rangle & \langle y^2 \rangle - \langle y \rangle^2 \end{pmatrix} \quad (3.55)$$

und andererseits, auf Grund von (3.51), auch durch

$$\Sigma = \mathbf{W} \hat{\Sigma} \mathbf{W}^T = \begin{pmatrix} \sigma_1^2 A^2 + \sigma_2^2 B^2 & \sigma_1^2 AC + \sigma_2^2 BD \\ \sigma_1^2 AC + \sigma_2^2 BD & \sigma_1^2 C^2 + \sigma_2^2 D^2 \end{pmatrix} \quad (3.56)$$

gegeben ist. Durch Koeffizientenvergleich in (3.54)–(3.56) und wegen $c = \langle x \rangle$, $f = \langle y \rangle$, kann für (3.54) auch

$$f(x; \Theta^{LLM}) = \langle y \rangle + \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2} (x - \langle x \rangle) \quad (3.57)$$

geschrieben werden. Dies ist genau die Funktionenapproximation durch Regression mit einer Ausgleichsgeraden [vgl. (2.11)]. Abbildung 3.5 illustriert den LLM-Approximator anhand des bekannten Datensatzes aus Abbildung 2.6. Da für $M > 1$ die der Kovarianzmatrix (3.55) entsprechenden lokalen Kovarianzmatrizen Σ_r durch lokale Mittelwertbildungen [vgl. (2.39)–(2.41)] bestimmt werden, gilt die Aussage (3.57) in den durch die unscharfen Partitionierungsfunktionen $a_r(x; \Theta^{LLM})$ definierten Gebieten aus \mathbb{R}^D auch für den allgemeinen Fall eines LLM-Approximators. Daraus kann geschlossen werden, daß die lokalen linearen Modelle eines LLM-Approximators die Ausgleichshyperebenen einer lokalen Regression sind, zwischen denen die Partitionierungsfunktionen $a_r(x; \Theta^{LLM})$ eine glatte Interpolation induzieren.

M lokale lineare Modelle

Wir wollen nun zum allgemeinen Fall eines LLM-Approximators zurückkehren, der mit mehreren lokalen linearen Modellen ($M > 1$) arbeitet, und diesen genauer diskutieren. Im folgenden wird von einem eindimensionalen Eingaberaum \mathbb{R} ausgegangen, der durch die Aktivierungsfunktionen $\tilde{a}_r(x; \Theta^{LLM})$ unscharf partitioniert wird. Entscheidend für die Lage dieser Partitionen sind die Zentren c_r und die Metriken $\tilde{\chi}_r^{-1}$, die im vorliegenden Spezialfall durch (3.52) gegeben sind. Abhängig von der Steigung C_r/A_r des Eigenvektors (A_r, C_r) , der zum größten Eigenwert $\sigma_{r,1}^2$ gehört [vgl. auch (3.51)], können zwei Grenzfälle unterschieden werden:

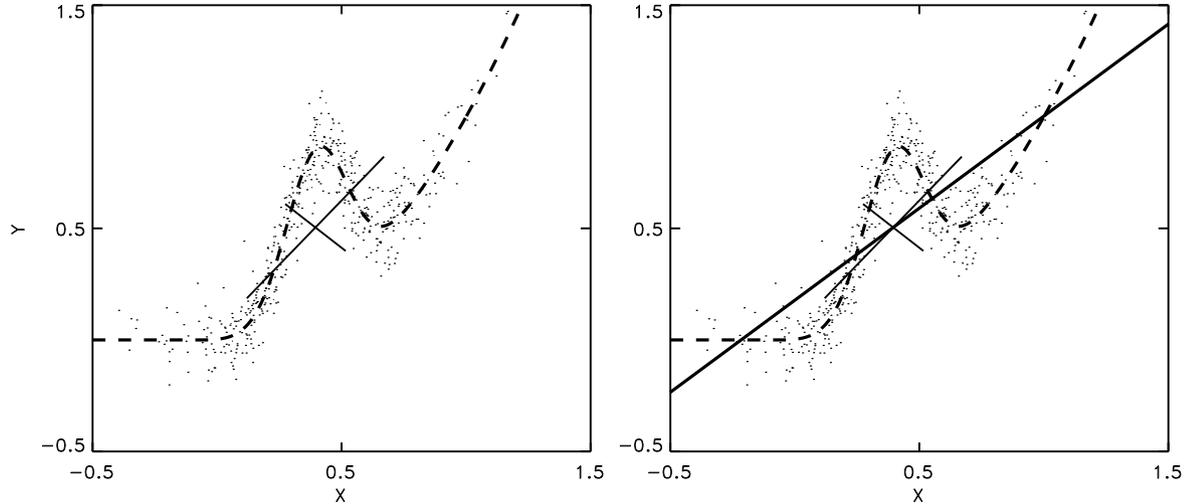


Abbildung 3.5.: **LLM-Approximator für $M = 1$.** Links sind die Eigenvektoren und Eigenwerte der globalen Kovarianzmatrix $\Sigma = \mathbf{W}\hat{\Sigma}\mathbf{W}^T$ der Daten $\mathcal{Z} = \{(x_k, y_k) | k = 1 \dots N\}$ dargestellt. Die gestrichelte Linie ist die ursprüngliche Funktion $f(x)$, die approximiert werden soll. Das rechte Bild zeigt die Funktionenapproximation (dicke Linie) durch (3.54) bzw. (3.57). Diese ist gerade die Ausgleichsgerade (vgl. Abbildung 2.7).

- Bei einer sehr kleinen Steigung ($C_r/A_r \approx 0$) wird $\tilde{\chi}_r^{-1} \approx \sigma_{r,1}^{-2}$. Die Varianz $\tilde{\chi}_r^{-1}$ der Normalverteilung $\tilde{p}(x|r, \Theta_r^{LLM})$, welche in die Partitionierungsfunktion $\tilde{a}_r(x; \Theta^{LLM})$ eingeht, ist in diesem Fall also gleich dem maximalen Eigenwert $\sigma_{r,1}^2$ der lokalen Schätzung.
- Bei einer sehr großen Steigung ($C_r/A_r \rightarrow \pm\infty$) drehen sich die Verhältnisse um. Da nun $\tilde{\chi}_r^{-1} \approx \sigma_{r,2}^{-2}$ gilt, wird die Ausdehnung der Partitionierungsfunktion $\tilde{a}_r(x; \Theta^{LLM})$ in diesem Fall von dem kleineren Eigenwert $\sigma_{r,2}^2$ bestimmt.

Die durch $\tilde{\chi}_r^{-1}$ definierte Skala der Partitionierungsfunktion paßt sich also an die lokale Struktur der Datenverteilung \mathcal{Z} an. In Regionen mit starken Steigungen der Zielfunktion wird damit die Diskretisierung der x -Achse dichter. Konstante Bereiche werden dagegen von lokalen Modellen mit großen Ausdehnungen approximiert. Abbildung 3.6 zeigt einen LLM-Approximator aus $M = 10$ lokalen linearen Modellen, an dem dieses Verhalten gut zu erkennen ist.

In analoger Weise können die lokalen linearen Modelle $\tilde{f}_r(x; \Theta^{LLM})$ untersucht werden. Für deren Steigungen $\tilde{\Phi}_r$, welche durch (3.53) gegeben sind, können wieder zwei Grenzfälle unterschieden werden:

- Stimmen die beiden lokalen Eigenwerte eines linearen Modells überein ($\sigma_{r,1} = \sigma_{r,2}$), so wird $\tilde{\Phi}_r$ zu Null, da wegen der Orthogonalität der Eigenvektoren $(B_r, D_r) =$

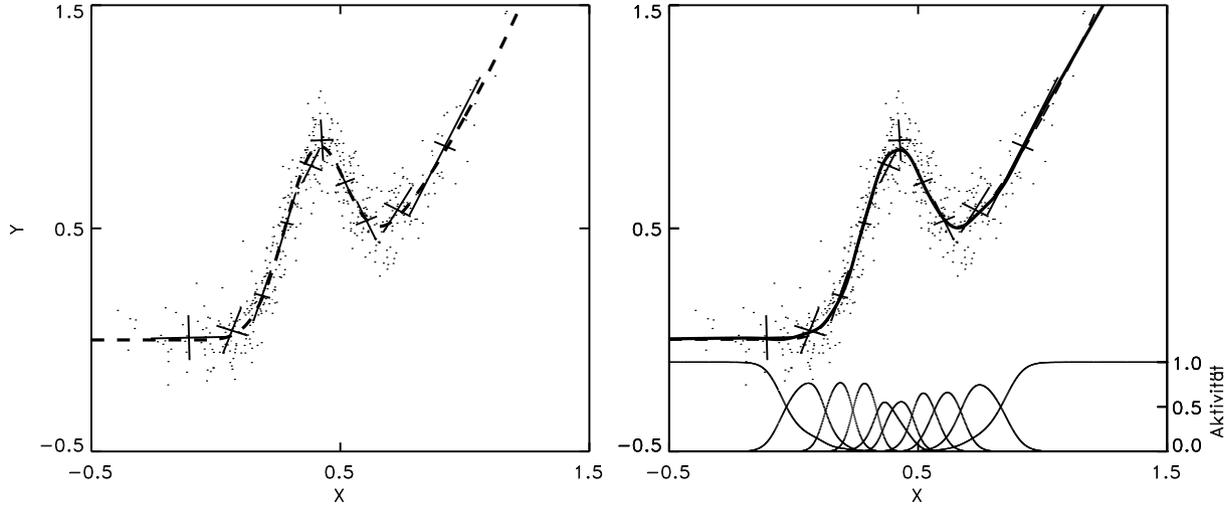


Abbildung 3.6.: **LLM-Approximator mit $M = 10$.** Links ist die multivariate ML-Dichteschätzung der bekannten Daten zusammen mit $f(x)$ gezeigt. Rechts ist die Funktionenapproximation $\hat{f}(x; \Theta^{LLM})$ durch einen LLM-Approximator dargestellt (dicke durchgezogene Linie). Unten sind zusätzlich die Partitionierungsfunktionen $\tilde{a}_r(x; \Theta^{LLM})$ für alle lokalen linearen Modelle eingezeichnet. $f(x)$ und $\hat{f}(x; \Theta^{LLM})$ stimmen in weiten Bereichen nahezu überein.

$(-C_r, A_r)$ gilt. Das entsprechende lokale Modell $\tilde{f}_r(x; \Theta_r^{LLM})$ reduziert sich dann auf die Konstante f_r .

- Ist dagegen $\sigma_{r,1} \gg \sigma_{r,2}$, was der generische Fall für eine glatte und wenig verrauschte Zielfunktion ist, dann berechnet sich die Steigung des lokalen linearen Modells zu $\tilde{\Phi}_r \approx C_r/A_r$, d.h. sie ist durch die Richtung des zu $\sigma_{r,1}$ gehörigen Eigenvektors gegeben.

Dieses Verhalten ist auch bei dem LLM-Approximator aus Abbildung 3.6 sichtbar: wann immer dort stark unterschiedliche $\sigma_{r,i}$ berechnet werden, paßt sich die Steigung des Eigenvektors $w_{r,1}$ an die Steigung der Zielfunktion an. Dieses Beispiel und die obigen Betrachtungen machen deutlich, wie sich die Partitionierungsfunktionen $\tilde{a}_r(\mathbf{x}; \Theta^{LLM})$ und die linearen Modelle $\tilde{f}_r(\mathbf{x}; \Theta_r^{LLM})$ bei der Funktionenapproximation mit einem LLM-Approximator an die lokalen Eigenheiten der Verteilung \mathcal{Z} anpassen. Obwohl sich die lokalen Eigenvektorsysteme \mathbf{W}_r nicht immer optimal an die zu approximierende Funktion $f(x)$ anschmiegen, wird die ursprüngliche Funktion durch die Interpolation der lokalen Modelle $f_r(x; \Theta_r^{LLM})$, welche durch die Gewichtung mit den unscharfen Partitionierungsfunktionen $\tilde{a}_r(\mathbf{x}; \Theta^{LLM})$ erzeugt wird, sehr gut approximiert.

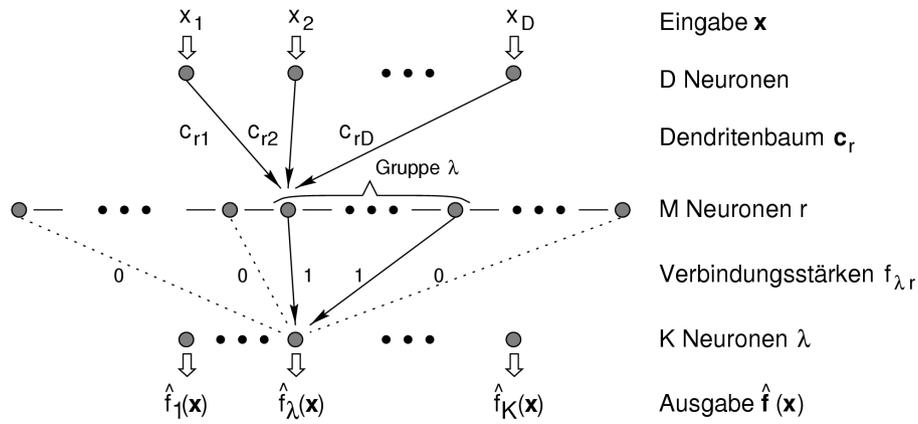


Abbildung 3.7.: Schema eines GRBF-Netzwerks für die Klassifikation eines Datensatzes \mathcal{X} .

Klassifikation mit einem LLM-Approximator

Zum Abschluß soll noch untersucht werden, wie sich der LLM-Approximator für die in Abschnitt 2.3.2 näher erläuterte Aufgabenstellung der Klassifikation verwenden läßt. Bei der Klassifikation wird einem Vektor \mathbf{x} aus einem Eingaberaum \mathbb{R}^D eine Klasse $\lambda \in \{1 \dots K\}$ zugeordnet. Dabei wird davon ausgegangen, daß die Daten eine innere Struktur besitzen, die eine Aufteilung in K verschiedene Klassen erlaubt. Für das Training eines Klassifikators ist zum einen ein Eingabedatensatz $\mathcal{X} = \{\mathbf{x}_k | k = 1 \dots N\} \subset \mathbb{R}^D$ erforderlich, der sich in disjunkte Teilmengen \mathcal{X}_λ mit $\mathcal{X} = \bigcap_{\lambda=1}^K \mathcal{X}_\lambda$ zerlegen läßt. Zum anderen ist ein Datensatz $\mathcal{Y} = \{\lambda_k | k = 1 \dots N\} \subset \mathbb{R}$ notwendig, der jedem Datenpunkt $\mathbf{x}_k \in \mathcal{X}$ eine Klasse λ_k zuweist. Der Beschreibung der Klasse mit einem Index λ ist ein Einheitsvektor $\mathbf{e}_\lambda \in \mathbb{R}^K$ äquivalent, dessen λ -te Komponente gleich Eins ist, also $(\mathbf{e}_\lambda)_j = \delta_{j\lambda}$, $j = 1 \dots K$. Daher kann auch der Datensatz $\mathcal{Y} = \{\mathbf{e}_{\lambda_k} | k = 1 \dots N\} \subset \mathbb{R}^K$ für das Training des Klassifikators verwendet werden.

Für diese spezielle Funktionenapproximation wird nun das Verhalten eines LLM-Approximators untersucht. Da die Verbundwahrscheinlichkeitsdichte $p(\mathbf{x}, \mathbf{y})$ in \mathbf{y} -Richtung aus einem Satz von K δ -Distributionen $\delta(\mathbf{y} - \mathbf{e}_\lambda)$ besteht, divergieren die Matrizen $\hat{\Sigma}_{r,2}^{-1}$ in (3.36). Folglich müssen wir zum ursprünglichen Ansatz (3.2) zurückkehren und diesen für die gegebene Aufgabenstellung auswerten. Dies erfolgt in Anhang A.1.2. Unter Berücksichtigung der speziellen Form der Verbunddichte $p(\mathbf{x}, \mathbf{y})$ bei der Aufgabe der Klassifikation und auf Grund der Eigenschaft der *load balance* des *multivar*-Algorithmus zeigt sich, daß die ML-Dichteschätzung $\hat{p}(\mathbf{x}, \mathbf{y} | \Theta^M)$ mit der Mischung (3.30) in K Teilsummen zerfällt, die jeweils nur die ML-Dichteschätzung eines klassenlokalen Datensatzes \mathcal{X}_λ betreffen. Man erhält daher für (3.2)

$$\hat{\mathbf{f}}(\mathbf{x}; \Theta^{LLM}) = \frac{\sum_{\lambda=1}^K \mathbf{e}_\lambda \sum_{r_\lambda=1}^{M_\lambda} \hat{p}(\mathbf{x} | r_\lambda, \Theta_{r_\lambda}^{PCA})}{\sum_{\lambda=1}^K \sum_{r_\lambda=1}^{M_\lambda} \hat{p}(\mathbf{x} | r_\lambda, \Theta_{r_\lambda}^{PCA})}, \quad (3.58)$$

wobei sich die Anzahl M_λ der verwendeten multivariaten Normalverteilungen aus den *a priori*-Wahrscheinlichkeiten P_λ der Klassen ergibt. Es gilt

$$\frac{M_\lambda}{M} \approx P_\lambda \quad \text{und} \quad \sum_{\lambda=1}^K M_\lambda = M. \quad (3.59)$$

Aus (3.58) wird deutlich, wie der LLM-Approximator für die Aufgabenstellung der Klassifikation zu einem GRBF-Netzwerk mit einer speziellen Verbindungstopologie wird. In Abbildung 3.7 ist dieses GRBF-Netzwerk dargestellt. Die M Zwischenschichtneuronen zerfallen entsprechend (3.59) in K Gruppen mit jeweils M_λ Neuronen. Da mit

$$\hat{p}(\mathbf{x}|\lambda, \Theta_\lambda^M) = \frac{1}{M_\lambda} \sum_{r_\lambda=1}^{M_\lambda} \hat{p}(\mathbf{x}|r_\lambda, \Theta_{r_\lambda}^{PCA}) \quad (3.60)$$

die *klassenlokale* ML-Dichteschätzung durch eine Mischung gleichgewichteter multivariater Normalverteilung einer Klasse λ charakterisiert ist, wird die Gesamtaktivität $A(\mathbf{x}; \Theta_\lambda^M)$ einer Neuronengruppe λ auf der Zwischenschicht äquivalent zur klassenlokalen ML-Schätzung (3.60). Die Parameter $\Theta_\lambda^M = \{\Theta_{r_\lambda}^{PCA} | r_\lambda = 1 \dots M_\lambda\}$ setzen sich dabei aus allen Parametern der multivariaten Normverteilungen (2.36) zusammen, die zu einer Klasse λ gehören. Mit (3.60) kann (3.58) auch zu

$$\hat{\mathbf{f}}(\mathbf{x}; \Theta^{LLM}) = \frac{\sum_{\lambda=1}^K e_\lambda M_\lambda \hat{p}(\mathbf{x}|\lambda, \Theta_\lambda^M)}{\sum_{\lambda=1}^K M_\lambda \hat{p}(\mathbf{x}|\lambda, \Theta_\lambda^M)} \quad (3.61)$$

umgeschrieben werden. Beachtet man nun noch die ML-Schätzung $\hat{P}_\lambda = M_\lambda/M$ der *a priori*-Wahrscheinlichkeit und den Bayesschen Satz (2.24), der dann für die ML-Schätzung $\hat{P}(\lambda|\mathbf{x}, \Theta_\lambda^M)$ der *a posteriori*-Wahrscheinlichkeit, eine Klasse λ bei einem gegebenen Datenpunkt \mathbf{x} zu erhalten,

$$\hat{P}(\lambda|\mathbf{x}, \Theta^M) = \frac{\hat{P}_\lambda \hat{p}(\mathbf{x}|\lambda, \Theta_\lambda^M)}{\sum_{\lambda=1}^K \hat{P}_\lambda \hat{p}(\mathbf{x}|\lambda, \Theta_\lambda^M)} \quad (3.62)$$

liefert, dann hat das GRBF-Netzwerk für die Aufgabenstellung der Klassifikation eines Datensatzes \mathcal{X} die Ausgabe

$$\hat{\mathbf{f}}(\mathbf{x}; \Theta^{LLM}) = \sum_{\lambda=1}^K e_\lambda \hat{P}(\lambda|\mathbf{x}, \Theta^M). \quad (3.63)$$

Da die Verbindungsstärken zwischen allen Neuronen einer Klasse λ und dem Ausgabe-neuron λ Eins und sonst Null sind, liegt an jedem Ausgabeneuron λ eines solchen GRBF-Netzwerks folglich gerade die geschätzte *a posteriori*-Wahrscheinlichkeit $\hat{P}(\lambda|\mathbf{x}, \Theta^M)$ an.

Die geschätzte *a posteriori*-Wahrscheinlichkeit $\hat{P}(\lambda|\mathbf{x}, \Theta^M)$ ist nach (3.62) aus den klassenlokalen ML-Schätzungen $\hat{p}(\mathbf{x}|\lambda, \Theta_\lambda^M)$ aufbaut, so daß nun nicht mehr der gesamte Datensatz $\{\mathcal{X}, \mathcal{Y}\}$ einer ML-Dichteschätzung unterzogen werden muß. Vielmehr ist es ausreichend, die Dichten der klassenlokalen Datensätze \mathcal{X}_λ getrennt zu schätzen. Dadurch wird

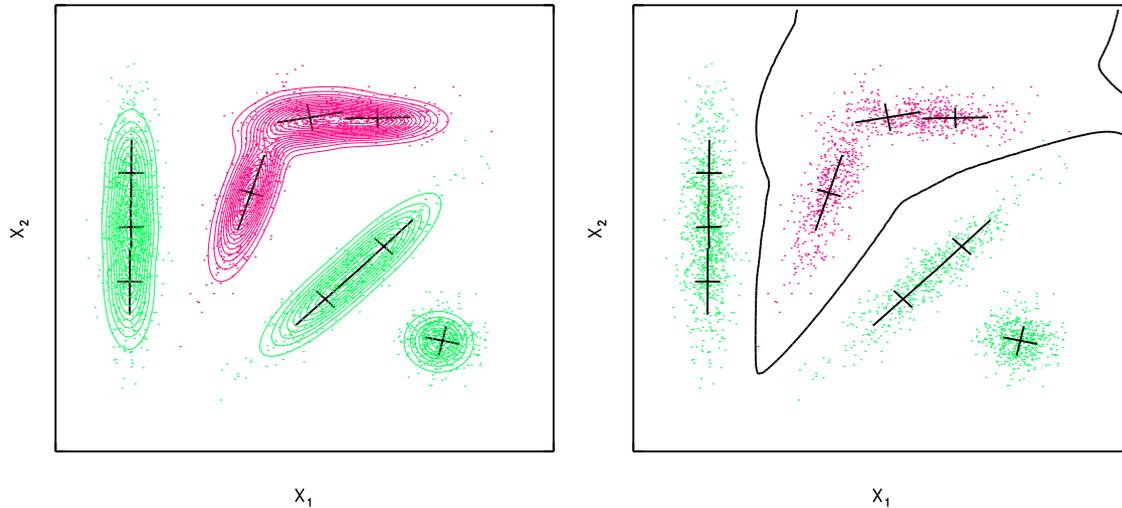


Abbildung 3.8.: **Klassifikation mit einem LLM-Approximator.** Die beiden Klassen $\lambda(\mathbf{x}) = 1$ (grüne Punkte) und $\lambda(\mathbf{x}) = 2$ (rote Punkte) wurden getrennt einer ML-Dichteschätzung durch den *multivar*-Algorithmus unterzogen. Links sind die daraus resultierenden Schätzungen $\hat{p}(\mathbf{x}|1, \Theta_1^M)$ und $\hat{p}(\mathbf{x}|2, \Theta_2^M)$ dargestellt. Die Gleichheit der geschätzten *a posteriori*-Wahrscheinlichkeiten $\hat{P}(1|\mathbf{x}, \Theta_1^M)$ und $\hat{P}(2|\mathbf{x}, \Theta_2^M)$ führt zu der im rechten Teilbild eingetragenen Entscheidungsgrenze.

der Aufwand zur Parameterbestimmung erheblich reduziert. Für die Auswertung von (3.63) ist es dann lediglich notwendig den Klassenindex λ der maximalen, geschätzten *a posteriori*-Wahrscheinlichkeit $\hat{P}(\lambda|\mathbf{x}, \Theta^M)$ zu bestimmen. Bei diesem Klassifikator handelt es sich um einen *optimalen* Bayesschen Klassifikator [DH73], dessen Qualität ausschließlich von der Güte der klassenlokalen ML-Dichteschätzungen abhängt.

In Abbildung 3.8 wird dieser Klassifikator für das schon bekannte Beispiel aus Abbildung 2.8 illustriert. Die *a priori*-Wahrscheinlichkeiten der beiden Klassen sind $P_1 = 2/3$ und $P_2 = 1/3$. Daher wurden zwei ML-Dichteschätzungen $\hat{p}(\mathbf{x}|1, \Theta_1^M)$ und $\hat{p}(\mathbf{x}|2, \Theta_2^M)$ mit einer Mischung aus $M_1 = 6$ und $M_2 = 3$ multivariaten Normalverteilungen ausgeführt. Die in der Abbildung eingetragene Klassengrenze ist durch die Gleichheit der *a posteriori*-Wahrscheinlichkeiten $\hat{P}(1|\mathbf{x}, \Theta_1^M)$ und $\hat{P}(2|\mathbf{x}, \Theta_2^M)$ definiert.

In seiner Diplomarbeit hat J. Busch einen analogen Klassifikator auf der Grundlage eines univariaten GRBF-Netzwerks hergeleitet, indem er die Verbindungsstärken \mathbf{f}_r probabilistisch interpretiert hat [Bus97]. Dieser Klassifikator ist inzwischen für Aufgaben der automatischen Spracherkennung erfolgreich angewendet worden [Wil98, Met98]. Die obige Ableitung aus einer ML-Schätzung der Verbunddichte $p(\mathbf{x}, \mathbf{y})$ wird in [ABK⁺99] veröffentlicht.

Zusammenfassung

In diesem Kapitel wurde zunächst ein GRBF-Netzwerk untersucht, das sich durch rückwärtige Verbindungen von der Ausgabeschicht zur Zwischenschicht auszeichnet. Auf Grund dieser reziproken Verbindungen können alle Parameter dieses GRBF-Netzwerks in einem gemeinsamen Lernprozeß aus den Ein- und Ausgabedaten bestimmt werden. Zur Erlangung einer größeren Flexibilität werden in dieser Netzwerkstruktur zusätzlich Übertragungsfunktionen eingeführt, die aus multivariaten Normalverteilungen aufgebaut sind. Für die Parameter eines solchen Netzwerks geben wir sequentielle stochastische Lernregeln an, die dazu führen, daß die Gesamtaktivität auf der Zwischenschicht einer ML-Schätzung der Trainingsdaten durch eine Mischung multivariater Normalverteilungen entspricht. Diese Lernregeln werden von uns auch als sequentieller *multivar*-Algorithmus bezeichnet [ABK⁺99].

Der Ausgangspunkt der obigen Ergebnisse ist die Funktionenapproximation (3.1), die auf der Schätzung der gemeinsamen Dichte $p(x, y)$ der Ein- und Ausgabedaten beruht. Für eine ML-Dichteschätzung mit einer Mischung multivariater Normalverteilungen führt der gewählte Ansatz zu einer Funktionenapproximation, die aus einer Linearkombination lokaler linearer Modelle besteht. Dieser sog. LLM-Approximator hat eine Reihe von Ähnlichkeiten mit einem GRBF-Netzwerk. Insbesondere zeigt sich, daß für den Fall einer ML-Schätzung mit einer Mischung univariater Normalverteilungen der LLM-Approximator zu einem univariaten GRBF-Netzwerk wird. Wenn der LLM-Approximator für Klassifikationsaufgaben herangezogen wird, reduziert sich dieser auf ein GRBF-Netzwerk, welches ursprünglich von J. Busch vorgeschlagen [Bus97] und inzwischen für Probleme der automatischen Spracherkennung eingesetzt wurde [Wil98, Met98]. Im allgemeinen Fall ist der LLM-Approximator jedoch nicht mehr als GRBF-Netzwerk interpretierbar, sondern liefert eine glatte Interpolation einer vorgegebenen Zahl lokaler Regressionen durch affine Funktionen.

4. Systeme lokaler Experten

Wir verlassen nun die neurobiologisch motivierte Herangehensweise an das Problem der Funktionenapproximation und wenden uns einer formalen Abstraktion des GRBF-Netzwerks und des LLM-Approximators zu. Dabei soll zunächst die Frage im Vordergrund stehen, wie das GRBF-Netzwerk und der schon nicht mehr vollständig neuronal begründbare LLM-Approximator zu einem *System lokaler Experten* verallgemeinert werden können, welches *verschiedene* Approximierungsstrategien in ein Gesamtsystem integriert. Dieses System wird die konzeptionelle und softwaretechnische Grundlage für die in Kapitel 7 zu beschreibende neue Regelung der Rückstandsverbrennung im Werk Burghausen bilden.

Im vorangehenden Kapitel wurde aus einer ML-Schätzung der Verbunddichte $p(x, y)$ durch eine Mischung multivariater Normalverteilungen der LLM-Approximator entwickelt, der eine Funktionenapproximation durch eine interpolierende Mischung lokaler linearer Modelle realisiert und der eine Mischung lokaler linearer Regressionsanalysen repräsentiert. Wie wir gesehen haben, reduziert sich der LLM-Approximator in einer Reihe von Spezialfällen auf das GRBF-Netzwerk. Sowohl das GRBF-Netzwerk als auch der LLM-Approximator beruhen darauf, daß sie zunächst den Eingaberaum durch Partitionierungsfunktionen in unscharf definierte Teilgebiete aufteilen. Diesen Teilgebieten wird jeweils eine lokale Modellfunktion zugewiesen, die beim GRBF-Netzwerk eine konstante und beim LLM-Approximator eine affine Funktion ist. Die Ausgabe erfolgt bei beiden Approximatores durch eine Gewichtung der lokalen Modellfunktionen mit den Partitionierungsfunktionen und durch Summation der gewichteten lokalen Modelle, was zur Interpolation dieser Modelle führt. Im folgenden wird diese Architektur des GRBF-Netzwerks und des LLM-Approximators aufgegriffen und zum Entwurf eines verallgemeinerten Konzepts der Funktionenapproximation genutzt, das eine Überlagerung *beliebiger* lokaler Funktionen in einem System lokaler Experten realisiert.

Die Idee, eine Problemstellung auf verschiedene Experten aufzuteilen und die Entscheidung über das entgeltige Ergebnis einem übergeordneten Experten zu überlassen¹, ist nicht neu. Der erste Anstoß, dieses Vorgehen auf die Architektur eines neuronalen Netzwerks zu übertragen, kam von R. Jacobs et al. [JJNH91]. Diese Autoren haben seitdem verschiedene Fragestellungen untersucht, die bei den Mischungen von Experten auftreten. All diesen Untersuchungen ist dabei gemeinsam, daß jeder Experte durch ein neuronales Netzwerk

¹In Anlehnung an den Grundsatz römischer Machtpolitik *divide et impera* wird dieses Vorgehen in der englischsprachigen Literatur mit *divide and conquer* umschrieben.

repräsentiert wird, welches einen bestimmten, in den Trainingsdaten vorhandenen Zusammenhang, modelliert. Die Ausgabe des Gesamtsystems wird durch ein *gating network* bestimmt, welches, abhängig vom eingehenden Merkmalsvektor x , eine Mischung der verschiedenen Experten vornimmt. Das Training einer solchen *mixture of experts* zielt auf die Maximierung der *log-likelihood* des gesamten Systems ab und erfolgt im einfachsten Fall durch einen Gradientenabstieg [JJ93a, AJ96] oder durch den *expectation maximization* (EM)-Algorithmus [JX93, JJ93b, JJ94].

Ein ähnlicher Ansatz wird von R. Murray-Smith verfolgt [MS94]. Dieser Autor fixiert aber weder den konkreten Aufbau der lokalen Modelle, noch wird das Vorgehen bei der Mischung der verschiedenen Modelle festgelegt. Der Schwerpunkt liegt in [MS94] vielmehr darin, wie der Zuständigkeitsbereich eines Modells geeignet bestimmt werden kann und wie sich eine Hierarchie solcher lokaler Modelle aufbauen läßt. In [MSE97] zeigen R. Murray-Smith und T. Johansen, in welchem engen Zusammenhang das Konzept der lokalen Modelle mit Konzepten aus der *Fuzzy*-Logik, der Regelungstheorie, der Neuroinformatik und der Statistik steht.

Im Gegensatz zum *top-down* Ansatz von M. Jordan et al. handelt es sich bei dem im folgenden vorgestellten Konzept um ein *bottom-up* Vorgehen, bei dem von den Eigenschaften des LLM-Approximators ausgegangen wird, für welchen in den Abschnitten 3.2 und 3.3 Lernregeln entwickelt wurden. Diese Eigenschaften werden stufenweise auf ein System lokaler Experten verallgemeinert. Hierbei wird sich herausstellen, daß für dieses System kein *gating network* notwendig ist. Zunächst sollen aber die Vorteile lokaler Experten gegenüber globalen Methoden diskutiert werden.

4.1. Warum sind lokale Experten vorteilhaft?

Welche Konstruktionsmerkmale zeichnen ein System lokaler Experten aus? Zunächst einmal müssen “Elemente” vorhanden sein, die im Merkmalsraum jeweils einen klar definierten Aufgabenbereich haben. Diese Bereiche können entweder in einem Selbstorganisationsprozeß ermittelt oder vom Systementwickler vorgegeben werden. Häufig spricht man dann – in Analogie zu sozialen Organisationsstrukturen – bei diesen “Elementen” auch von *Experten*². Des weiteren ist ein Verfahren notwendig, mit dem die Ausgaben der Experten sinnvoll verknüpft werden können.

Wie erwähnt, wird das Konzept lokaler Experten seit einigen Jahren verstärkt untersucht wird. Dabei hat sich herausgestellt, daß in solchen Systemen das lokale Lernen eine Reihe

²Allerdings darf dieses Konzept nicht mit den *Expertensystemen* der symbolischen künstlichen Intelligenz verwechselt werden. Diese Expertensysteme beruhen darauf, daß die Erfahrung eines oder vieler menschlicher Experten in symbolischen Regeln formuliert wird, die automatisch ausgewertet werden [Sch86]. Der im folgenden wichtige Aspekt des lokalen Lernens spielt bei diesen Systemen keine Rolle.

von Vorteilen gegenüber Lösungsansätzen bietet, welche durch ein globales Modell, dessen Parameter mit einem globalen Verfahren bestimmt werden, zum Ziel kommen wollen [MS94]:

- Das Training eines einzelnen Experten ist numerisch effizienter als das Training eines großen, alles beschreibenden Netzwerks [MS94].
- Das globale Training führt zu Parametern, die auch eine globale Bedeutung haben. Dadurch werden solche Netzwerke schwer interpretierbar. Im Gegensatz dazu haben die Parameter lokaler Experten nur eine begrenzte Wirkung und sind damit wesentlich besser zu interpretieren [JJ93a].
- Wenn neue Eigenschaften integriert werden sollen, dann ist dies mit einem System lokaler Experten einfach zu realisieren, indem ein neuer Experte mit der entsprechenden Eigenschaft hinzugefügt wird. Bei einem globalen Modell ist in einem solchen Fall ein neues Training notwendig [MS94].
- Auch wenn ein System lokaler Experten nicht immer eine optimale Approximationsqualität aufweist, ist es trotzdem wegen seiner höheren Robustheit und Interpretierbarkeit einer globalen Modellierung vorzuziehen [MS94].
- In ein System lokaler Experten können zwanglos verschiedene Verfahren und Algorithmen integriert werden. Solche heterogenen Systeme sind mit globalen Methoden nicht realisierbar [JJ93a].

Im nächsten Abschnitt wird an Hand des GRBF-Netzwerks und des LLM-Approximators ein System lokaler Experten entwickelt. Das daraus resultierende, allgemeine Konzept wird zunächst vergleichend diskutiert. Zum Abschluß dieses Kapitels wird ein komplexeres Beispiel dargestellt, auf welches für die Regelungsaufgaben in Kapitel 6. und 7 noch zurückgegriffen werden wird.

4.2. Verallgemeinerung des GRBF-Netzwerks auf ein System lokaler Experten

Das in Abschnitt 3.2 vorgestellte multivariate GRBF-Netzwerk ist ebenso wie der LLM-Approximator aus Abschnitt 3.3 zunächst ein sehr spezielles Modell. Erst die Reduktion dieser Modelle auf ihre wesentlichsten Eigenschaften führt zu einem System lokaler Experten. Diese Verallgemeinerung soll in drei Schritten erfolgen, deren Resultat in Abbildung 4.1 illustriert ist:

- 1. Schritt.** Die Zwischenschichtaktivitäten $a_r(\mathbf{x}; \Theta^M)$ des GRBF-Netzwerks und des LLM-Approximators werden aus Normalverteilungen $\hat{p}(\mathbf{x}|r; \Theta_r^{PCA})$ [vgl. (2.36)]

und statistischen Gewichten P_r berechnet. Von nun an sollen die $\hat{p}(\mathbf{x}|r; \Theta_r^{PCA})$ eine beliebige funktionale Form annehmen können. Die neuen Funktionen $k_r(\mathbf{x}; \Theta_r^{Exp})$ werden im folgenden auch als *lokale Kompetenzen* bezeichnet und definieren die Zuständigkeitsbereiche der lokalen Experten r im Merkmalsraum, wobei auf eine globale Normierung (2.59) wie im GRBF-Netzwerk und im LLM-Approximator zunächst verzichtet wird. Somit weist sich jeder Experte zu einer Eingabe \mathbf{x} durch $k_r(\mathbf{x}; \Theta_r^{Exp})$ selbständig eine lokale Kompetenz zu, welche seine erste Ausgabe bildet. Von dieser Ausgabe wird nur gefordert, daß sie Werte größer oder gleich Null annimmt.

2. Schritt. Zusätzlich zu dieser ersten Ausgabe liefert jeder lokale Experte zur Eingabe \mathbf{x} einen Funktionswert $\hat{\mathbf{f}}_r(\mathbf{x}; \Theta_r^{LLM})$, den er aus einer beliebigen Funktion \mathbf{f}_r berechnet. Damit kann ein lokaler Experte durch ein frei wählbares Verfahren eine Modellbildung vornehmen, die nicht auf die konstanten Modelle \mathbf{f}_r des multivariaten GRBF-Netzwerks oder die linearen Modelle $\mathbf{f}_r(\mathbf{x}; \Theta_r^{LLM})$ des LLM-Approximators (3.41) eingeschränkt ist.

3. Schritt. Die statistischen Gewichte P_r der M lokalen Experten werden nicht auf $1/M$ fixiert, sondern zur heuristischen Gewichtung ihrer Aussagen verwendet. Diese Parameter bekommen damit die Bedeutung einer *übergeordneten Kompetenz* und werden fortan mit K_r bezeichnet. Sie sind den lokalen Experten nicht bekannt, sondern werden diesen durch einen hierarchisch übergeordneten, nachfolgenden Experten, den sog. *Entscheider*, zugeordnet.

Ein lokaler Experte r hat also als Eingabe einen Merkmalsvektor $\mathbf{x} \in \mathbb{R}^D$ und gibt eine Schätzung seiner lokalen Kompetenz $k_r(\mathbf{x}; \Theta_r^{Exp})$ und einen Funktionswert $\hat{\mathbf{f}}_r(\mathbf{x}; \Theta_r^{Exp})$ aus, wobei die Funktion $\hat{\mathbf{f}}_r$ ein lokales Modell des funktionalen Zusammenhangs im Zuständigkeitsbereich des Experten darstellt. Abbildung 4.1 zeigt einen Ausschnitt aus einem System solcher lokaler Experten. In der ersten Schicht erhalten die sog. *Spezialisten* Eingaben \mathbf{x} (z.B. von einem Sensorsystem) und berechnen Kompetenzen $k_r(\mathbf{x}; \Theta_r^{Exp})$ und Funktionswerte $\hat{\mathbf{f}}_r(\mathbf{x}; \Theta_r^{Exp})$. Alle diese Informationen werden zu einem neuen Merkmalsvektor $\mathbf{x}'(\mathbf{x}) = \{k_r(\mathbf{x}; \Theta_r^{Exp}), \hat{\mathbf{f}}_r(\mathbf{x}; \Theta_r^{Exp}) | r = 1 \dots M\} \in \mathbb{R}^{D'}$ zusammengefaßt und an einen nachgeschalteten Experten r' , den Entscheider, weitergegeben. Dieser berechnet eine neue Kompetenz $k_{r'}(\mathbf{x}'; \Theta_{r'}^{Exp})$ und einen neuen Funktionswert $\hat{\mathbf{f}}_{r'}(\mathbf{x}'; \Theta_{r'}^{Exp})$.

Im Gegensatz zu den frei wählbaren lokalen Modellen $\hat{\mathbf{f}}_r(\mathbf{x}; \Theta_r^{Exp})$ der vorgeschalteten Experten, ist die funktionale Form von $\hat{\mathbf{f}}_{r'}(\mathbf{x}'; \Theta_{r'}^{Exp})$ durch die Aufgabe des Entscheiders, die Aussagen der vorgeschalteten Experten zusammenzuführen, eingeschränkt. In Analogie zur Bildung des Gesamtergebnisses beim GRBF-Netzwerk (2.62) und beim LLM-Approximator (3.41) wird hier für $\hat{\mathbf{f}}_{r'}(\mathbf{x}'; \Theta_{r'}^{Exp})$ eine kompetitive Interpolation der Form

$$\hat{\mathbf{f}}_{r'}(\mathbf{x}'(\mathbf{x}); \Theta_{r'}^{Exp}) = \frac{\sum_{r=1}^M \hat{\mathbf{f}}_r(\mathbf{x}; \Theta_r^{Exp}) K_r k_r(\mathbf{x}; \Theta_r^{Exp})}{\sum_{s=1}^M K_s k_s(\mathbf{x}; \Theta_s^{Exp})} \quad (4.1)$$

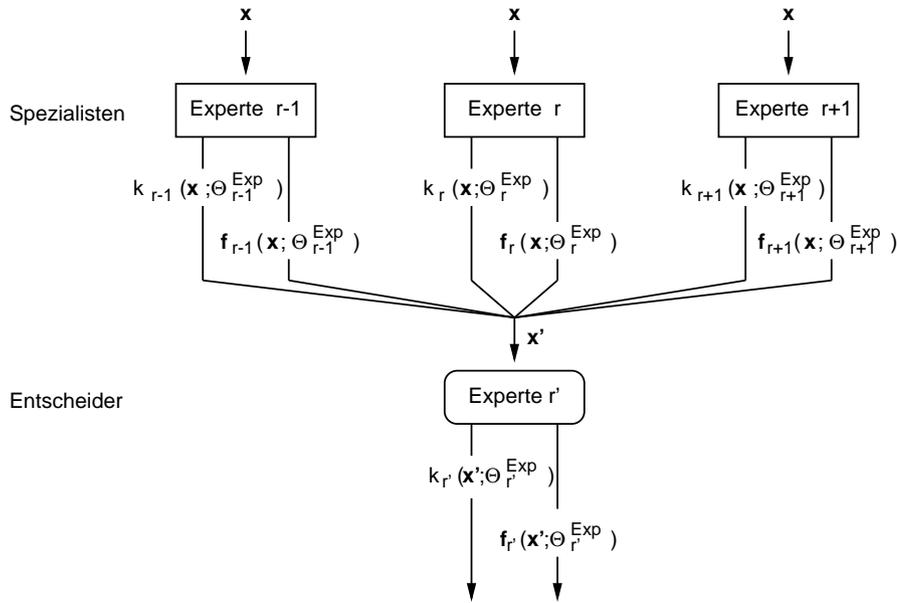


Abbildung 4.1.: **System lokaler Experten.** Ein Spezialist r hat Zugriff auf die Eingabedaten \mathbf{x} . Die Ausgabe eines solchen lokalen Experten setzt sich aus der lokalen Kompetenz $k_r(\mathbf{x}; \Theta_r^{Exp})$ und dem Modell $\hat{f}_r(\mathbf{x}; \Theta_r^{Exp})$ zusammen. Diese Informationen bilden zusammen die Eingabe \mathbf{x}' für den Entscheider r' , der wiederum eine lokale Kompetenz $k_{r'}(\mathbf{x}'; \Theta_{r'}^{Exp})$ und ein Modell $\hat{f}_{r'}(\mathbf{x}'; \Theta_{r'}^{Exp})$ berechnet.

gewählt. Für ein gegebenes \mathbf{x}' bildet $\hat{f}_{r'}(\mathbf{x}'; \Theta_{r'}^{Exp})$ den zweiten Teil der Ausgabe des Entscheiders. Den ersten Teil bildet die Selbsteinschätzung $k_{r'}(\mathbf{x}'; \Theta_{r'}^{Exp})$ seiner Kompetenz, die er gemäß

$$k_{r'}(\mathbf{x}'; \Theta_{r'}^{Exp}) = \sum_{r=1}^M K_r k_r(\mathbf{x}; \Theta_r^{Exp}) \quad (4.2)$$

durch Linearkombination aus den Kompetenzen $k_r(\mathbf{x}; \Theta_r^{Exp})$ der vorgeschalteten Experten, sowie aus einem Satz von Gewichten K_r , über die er verfügt, berechnet. Der Kompetenzbereich des Entscheiders umfaßt also die Kompetenzbereiche der vorgeschalteten Spezialisten und seine Modellbildung repräsentiert eine Interpolation der vorgeschalteten Modelle. Damit läßt sich der Entscheider wieder als einfacher Experte auffassen und beispielsweise mit anderen lokalen Experten parallel schalten, wenn ein weiterer nachgeschalteter Entscheider die Zusammenfassung der Aussagen übernimmt.

In dem oben formulierten Konzept fehlt zunächst jegliche Aussage über das konkrete Vorgehen bei der Modellbildung und der Kompetenzbestimmung. Dies ist auch nicht das Ziel eines Systems lokaler Experten. Vielmehr soll ein solches System lediglich einen allgemeinen konzeptionellen Rahmen bilden, in dem verschiedene Methoden zusammenarbeiten können.

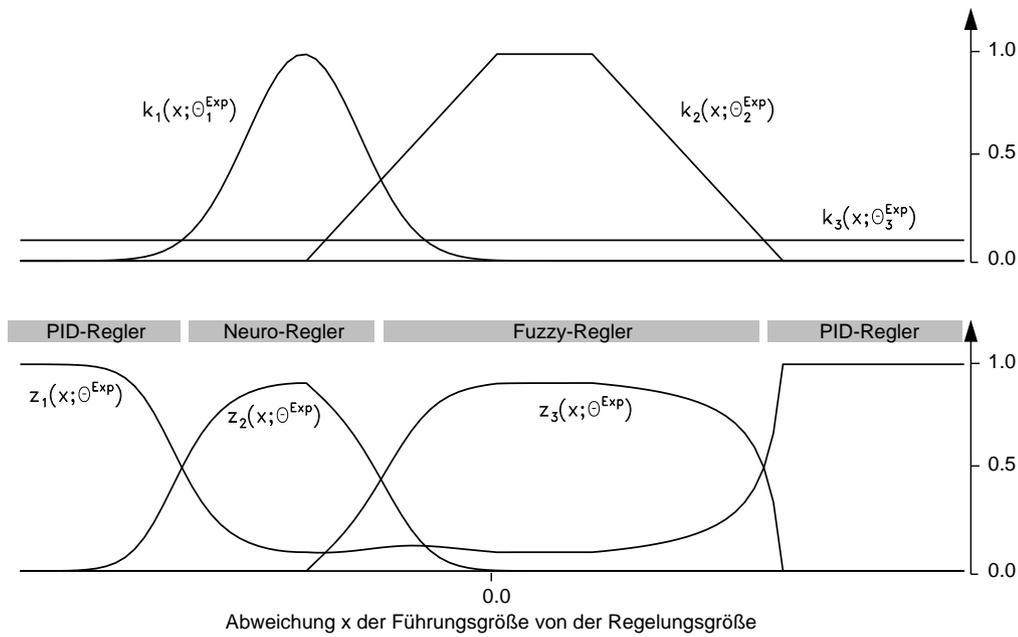


Abbildung 4.2.: **Verhalten der lokalen Kompetenz.** Das Beispiel zeigt im oberen Teilbild die lokalen Kompetenzen eines Neuro-, *Fuzzy*- und PID-Reglers in einem System lokaler Experten. Unten ist dargestellt, wie der Entscheider über die Zuständigkeiten $z_r(\mathbf{x}; \Theta^{Exp})$ daraus eine unscharfe Partitionierung des Merkmalsraums berechnet. In der Abbildung wurden die übergeordneten Kompetenzen K_r aller Experten auf Eins gesetzt.

Als Beispiel für ein System lokaler Experten zeigt Abbildung 4.2, wie ein Neuro-, ein *Fuzzy*- und ein PID-Regler in ein Regelungssystem integriert werden können³. Im oberen Teil der Abbildung sind die heuristisch gewählten lokalen Kompetenzen $k_r(\mathbf{x}; \Theta_r^{Exp})$ der drei Experten dargestellt, wobei der Merkmalsvektor x im gegebenen Fall die Abweichung der zu regelnden Größe von deren aktueller Messung bezeichnet. Der Kompetenzbereich des Neuro-Reglers ist durch die Gaußfunktion $k_1(\mathbf{x}; \Theta_1^{Exp})$ festgelegt, der Bereich des *Fuzzy*-Reglers durch die trapezförmige Funktion $k_2(\mathbf{x}; \Theta_2^{Exp})$, während die lokale Kompetenz des PID-Reglers über dem gesamten Merkmalsraum eine kleine Konstante ist. Im gewählten Beispiel stellt der PID-Regler somit eine Art Notfallsystem dar, welches immer dann regelnd eingreifen soll, wenn der Neuro- und der *Fuzzy*-Regler sich selbst zu kleine Kompetenzen zuweisen. Im unteren Teilbild ist gezeigt, wie der Entscheider r' den einzelnen Experten zunächst *Zuständigkeiten*

$$z_r(\mathbf{x}; \Theta_{r'}^{Exp}) \equiv \frac{K_r k_r(\mathbf{x}; \Theta_r^{Exp})}{\sum_{s=1}^M K_s k_s(\mathbf{x}; \Theta_s^{Exp})} \quad (4.3)$$

³Die verschiedenen Regelungskonzepte, die diese Regler jeweils charakterisieren, werden in Kapitel 5 noch näher erläutert.

zuordnet, welche nach (4.1) gemäß

$$\hat{\mathbf{f}}_{r'}(\mathbf{x}'(\mathbf{x}); \Theta_{r'}^{Exp}) = \sum_{r=1}^M \hat{\mathbf{f}}_r(\mathbf{x}; \Theta_r^{Exp}) z_r(\mathbf{x}; \Theta_{r'}^{Exp}) \quad (4.4)$$

zur Berechnung der Interpolation benötigt werden. Die Zuständigkeiten repräsentieren – ganz analog zu den Zwischenschichtaktivitäten eines GRBF-Netzwerks – eine unscharfe Partition des Merkmalsraums, welche jene Bereiche identifiziert, in denen vornehmlich ein bestimmter Experte die Gesamtausgabe $\hat{\mathbf{f}}(\mathbf{x}; \Theta^{Exp})$ bestimmt.

Ein Vergleich des oben dargestellten Systems von lokalen Experten mit der *mixture of experts* von R. Jacobs et al. [JJ93a, JX93, JJ93b, JJ94, AJ96] macht zwei wesentliche Unterschiede deutlich. Bei der *mixture of experts* ist ein spezielles *gating network* vorgeschaltet, das an Hand der Eingabe \mathbf{x} die Entscheidung trifft, welcher Experte aktiv wird. In [JJ93b] wird als Beispiel die Regelung eines Roboterarms untersucht, welcher durch verschiedene Nutzlasten belastet wird. Die einzelnen Experten bekommen als Eingabedaten lediglich die Raumkoordinaten, die der Roboterarm anfahren soll. Nur das *gating network* erhält zusätzlich die Information über die Masse der Nutzlast und kann so die Auswahl eines geeigneten Experten treffen. In einem System lokaler Experten wird diese Entscheidung dagegen zum Teil “dezentral” von den Experten getroffen werden, indem diese sich selbständig Kompetenzen zuweisen. Insofern der Entscheider die Selbsteinschätzung der lokalen Experten durch Zuweisung übergeordneter Kompetenzgewichte modifizieren kann, hat er allerdings eine ähnliche Funktion wie ein *gating network*.

Der zweite Unterschied betrifft das Training der Experten. R. Jacobs et al. schlagen ein Training mit dem *backpropagation*- oder dem EM-Algorithmus vor. Dabei werden alle Parameter der *mixture of experts* gemeinsam trainiert. Dies macht besonders die nachträgliche Erweiterung der *mixture of experts* um weitere Aufgaben und die Integration von konventionellen Verfahren schwierig. Beim System lokaler Experten ist dies auf Grund der vollständigen Lokalität der Experten einfacher. In Kapitel 7 wird gerade diese Eigenschaft von großer Bedeutung sein, um verschiedene Regelungsverfahren in ein Gesamtsystem zur Regelung der Rückstandsverbrennung zu integrieren. Es muß aber angemerkt werden, daß die Entwicklung von globalen Optimierungs- und Lernszenarien für solche Systeme ein künftig noch näher zu untersuchendes Problem darstellt.

Das sehr offene Konzept der lokalen Experten stellt also zunächst nur eine “Konstruktionsanleitung” für die Entwicklung komplexer, hierarchischer Systeme dar, die z.B. für die Funktionenapproximation oder die Regelung technischer Anlagen verwendet werden können. Im folgenden Abschnitt wird am Beispiel der Funktionenapproximation einer multifunktionalen Datenfolge illustriert, wie zwei konkrete Realisierungen strukturiert sein können. Auch diese beiden Realisierungen von Systemen lokaler Experten werden für die Regelungsaufgaben in Kapitel 6 und 7 noch von Bedeutung sein.

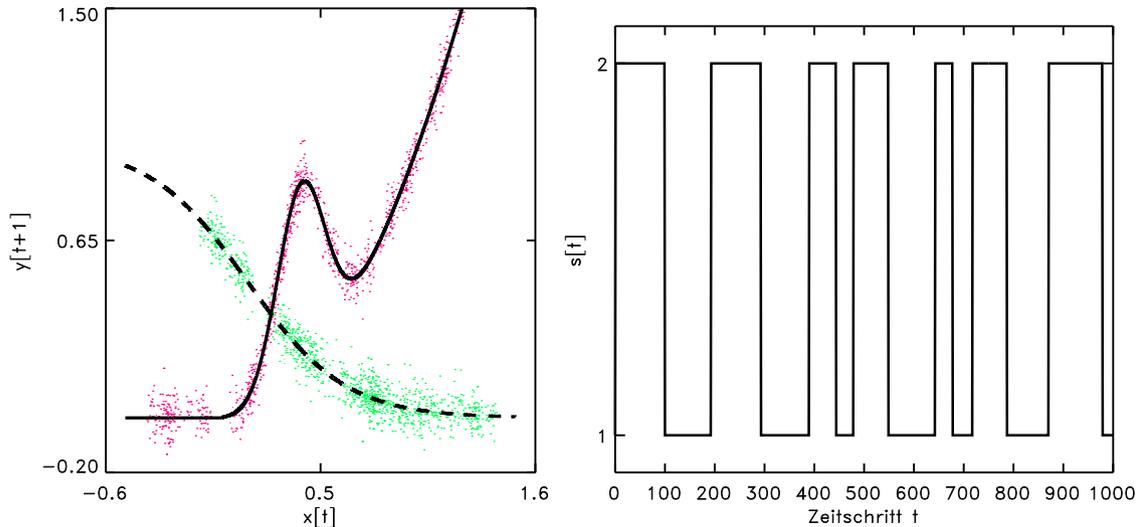


Abbildung 4.3.: **Multifunktionale Datenfolge.** Die links dargestellten Datenpunkte ergeben sich aus einem System, welches auf Grund einer stochastischen Dynamik zwei verschiedene Zustände $s[t] \in \{1, 2\}$ annehmen kann. Abhängig vom Zustand $s[t]$ wird eine Abbildung $y[t+1] = f_{s[t]}(x[t])$ ausgeführt, die zusätzlich noch leicht verrauscht wird (rote und grüne Punkte). Die beiden Funktionen $f_1(x)$ und $f_2(x)$ sind links als gestrichelte und durchgezogene Linie eingezeichnet. Die Datenfolge $\{x[t] \mid t = 1 \dots N\}$ folgt der stochastischen Dynamik eines *random walk*: Die nach jedem Zeitschritt erfolgende Verschiebung $\Delta x = x[t] - x[t-1]$ wird aus einem Intervall $[-1/40, 1/40]$ nach einer Gleichverteilung gewürfelt. Zusätzlich wird $x[t]$ nach jedem Zustandswechsel aus dem Intervall $[-0.5, 1.5]$ neu initialisiert.

4.3. Approximation einer multifunktionalen Datenfolge

Bei einer *multifunktionalen* Datenfolge $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$ werden die Ausgabedaten \mathcal{Y} nicht nur durch *eine* Funktion $f(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^K$ sondern durch einen Satz von S Funktionen $f_s(\mathbf{x})$, $s = 1 \dots S$, generiert. Die Auswahl der jeweils aktuellen Funktion $f_{s[t]}(\mathbf{x})$ kann dabei beispielsweise durch einen diskreten, stochastischen Prozeß erfolgen, der eine diskrete stochastische Zustandsfolge $\mathcal{S} = \{s[t] \in \mathbb{N} \mid 1 \leq s[t] \leq S, t = 1 \dots N\}$ erzeugt. Es werde nun angenommen, daß die Zeitreihe, welche durch die zeitliche Abfolge der Zustände \mathcal{S} gebildet wird, einem homogenen Markov-Prozeß [Hak82] gehorche, der eine langsam veränderliche Dynamik modelliert [Son97]. Dies bedeutet, (i) daß die bedingte Wahrscheinlichkeit $P(s[t], t \mid s[t-1], t-1, s[t-2], t-2, \dots)$, das System zum Zeitpunkt t im Zustand $s[t]$ zu finden, ausschließlich vom Zustand $s[t-1]$ zum Zeitpunkt $t-1$ abhängt und (ii) daß die mittlere Lebensdauer eines Zustands deutlich größer Eins ist.

Die Aufgabe eines Systems lokaler Experten soll nun sein, die gegebene multifunktionale Datenfolge $\mathcal{Z} = \{\mathbf{x}[t], y[t+1] \mid t = 1 \dots N-1\}$ durch eine geeignete Modellfunktion

$\hat{\mathbf{f}}(\mathbf{x}; \Theta)$ zu approximieren. Dabei ist $\mathbf{x}[t]$ die zum Zeitpunkt t beobachtete Eingabe eines dynamischen Systems und $\mathbf{y}[t + 1]$ faßt die Systemantwort zum Zeitpunkt $t + 1$ auf diese Eingabe zusammen. Das Problem der Approximation multifunktionaler Datenfolgen tritt immer dann auf, wenn ein System vorliegt, welches auf Grund von Unstetigkeiten sehr starke Schwankungen aufweist. Ein solches Verhalten werden wir in Kapitel 6 bei der Regelung eines Bioreaktors und in Kapitel 7 bei der Modellierung von Ventilkennlinien wieder antreffen.

Abbildung 4.3 zeigt links eine einfache multifunktionale Datenfolge \mathcal{Z} , die durch $S = 2$ Zustände bestimmt wird. Im rechten Teilbild ist ein Ausschnitt aus der Zeitreihe der Zustände dargestellt, durch welche zwischen den beiden Funktionen $f_1(x)$ und $f_2(x)$ hin- und hergeschaltet wird. Diese Zeitreihe erfüllt die beiden oben beschriebenen Annahmen. Zusätzlich erfüllt die Eingabedatenfolge $\mathcal{X} = \{x[t] \mid t = 1 \dots N - 1\}$ die Dynamik eines *random walk* Prozesses [Hak82]. Um trotzdem eine gewisse Gleichverteilung von \mathcal{X} zu erreichen, wird bei jedem Zustandswechsel $x[t]$ neu aus dem Intervall $[-0.5, 1.5]$ gewürfelt.

4.3.1. mGRBF-System für eine multifunktionale Datenfolge

M. Sonner hat in seiner Diplomarbeit ein Verfahren entwickelt, welches die oben geschilderte Aufgabenstellung bearbeitet [Son97]. Er führt hierzu multiple GRBF-Netzwerke, kurz das *mGRBF-System*, ein. Dieses ist ein System lokaler Experten, die jeweils durch ein univariates GRBF-Netzwerk der Form (2.59) gebildet werden. Die Ausgabe des mGRBF-Systems wird nach (4.3) und (4.4) durch

$$\hat{\mathbf{f}}(\mathbf{x}[t]; \Theta^{mGRBF}) = \sum_{l=1}^L \hat{\mathbf{f}}_l(\mathbf{x}[t]; \Theta_l^{GRBF}) \bar{z}_l[t] \quad (4.5)$$

berechnet. $\bar{z}_l[t]$ sind die Zuständigkeiten eines GRBF-Netzwerks l und werden durch eine gleitende Mittelwertbildung

$$\bar{z}_l[t] = \left(1 - \frac{1}{\tau}\right) \bar{z}_l[t - 1] + \frac{1}{\tau} z_l(\mathbf{x}[t - 1], \mathbf{y}[t]; \Theta^{mGRBF}) \quad (4.6)$$

mit einem *Gedächtnis* τ aus den *aktuellen* Zuständigkeiten $z_l(\mathbf{x}[t - 1], \mathbf{y}[t]; \Theta^{mGRBF})$ gebildet. Die aktuellen Zuständigkeiten werden analog zu den Zwischenschichtaktivitäten (2.59) eines GRBF-Netzwerks durch

$$z_l(\mathbf{x}[t - 1], \mathbf{y}[t]; \Theta^{mGRBF}) \equiv \frac{\exp \left[- \left(\hat{\mathbf{f}}_l(\mathbf{x}[t - 1]; \Theta_l^{GRBF}) - \mathbf{y}[t] \right)^2 / 2\sigma_y^2 \right]}{\sum_{l'=1}^L \exp \left[- \left(\hat{\mathbf{f}}_{l'}(\mathbf{x}[t - 1]; \Theta_{l'}^{GRBF}) - \mathbf{y}[t] \right)^2 / 2\sigma_y^2 \right]} \quad (4.7)$$

bestimmt.

Im mGRBF-System werden also die Zuständigkeiten (4.6) des l -ten GRBF-Netzwerks dadurch festgelegt, wie gut dessen Funktionenapproximationen $\hat{\mathbf{f}}_l(\mathbf{x}[t - 1]; \Theta_l^{GRBF})$ mit den

tatsächlichen Systemantworten $\mathbf{y}[t]$ bei Eingaben $\mathbf{x}[t - 1]$ in der Vergangenheit übereingestimmt haben. Eine gute Funktionenapproximation durch das l -te GRBF-Netzwerk über eine Zeitspanne τ führt somit zu einer großen Zuständigkeit $\bar{z}_l[t]$. Tritt in der multifunktionalen Datenfolge \mathcal{Z} ein Zustandswechsel auf, so wird die Zuständigkeit des nun am besten passenden GRBF-Netzwerks zunehmen und entsprechend eine bessere Funktionenapproximation (4.5) ausgegeben. Im günstigsten Fall wird im mGRBF-System durch jedes der L GRBF-Netzwerke eine der S Funktionen $\mathbf{f}_s(\mathbf{x})$, $s = 1 \dots S$, der multifunktionalen Datenfolge approximiert.

Dieser Idealfall kann natürlich nur durch ein problemangepaßtes Training der Parameter $\Theta^{mGRBF} = \{\Theta_l^{GRBF} \mid l = 1 \dots L\}$ eintreten. M. Sonner untersucht in seiner Arbeit das mGRBF-System nur für den Fall $\hat{f}(x; \Theta^{mGRBF}) : \mathbb{R} \rightarrow \mathbb{R}$, d.h. nur für eine eindimensionale Ein- und Ausgabedatenfolge [Son97]. Außerdem setzt M. Sonner voraus, daß die Eingabedatenfolge \mathcal{X} in einem endlichen Intervall gleichverteilt ist. Für diese Situation schlägt M. Sonner vor, auf ein Training der Zentren $c_r \in \mathbb{R}$ eines GRBF-Netzwerks zu verzichten und diese äquidistant im gegebenen Intervall zu verteilen. Für das Training der Verbindungsstärken der L GRBF-Netzwerke modifiziert M. Sonner die *outstar*-Lernregel (2.64), indem er diese um die Zuständigkeiten (4.6) erweitert. Für den r -ten Axonbaum $\mathbf{f}_{r,l}$ des l -ten GRBF-Netzwerks erhält er somit die Lernregel

$$\Delta \mathbf{f}_{r,l}(\mathbf{x}[t - 1], \mathbf{y}[t]) = a_{r,l}(\mathbf{x}[t - 1]; \Theta_l^{GRBF}) \bar{z}_l[t] (\mathbf{y}[t] - \mathbf{f}_{r,l}). \quad (4.8)$$

Diese Lernregel führt dazu, daß jeweils dasjenige GRBF-Netzwerk am stärksten durch einen Lernreiz $(\mathbf{x}[t - 1], \mathbf{y}[t])$ angepaßt wird, dessen vergangenen Funktionenapproximationen $\hat{\mathbf{f}}_l(\mathbf{x}[t - 1 - t']; \Theta_l^{GRBF})$, $t' = 1 \dots \tau$, am besten mit den Zielausgaben $\mathbf{y}[t - t']$, $\tau = 1 \dots \tau$, übereingestimmt haben.

M. Sonner untersucht in seiner Arbeit die Stationaritäts- und Stabilitätsbedingungen der Lernregel (4.8) in Abhängigkeit von der Gaußbreite σ_y und dem Gedächtnisparameter τ . Er schlägt ein Lernverfahren vor, bei dem zunächst σ_y und τ relativ groß gewählt werden. Der Lernvorgang gliedert sich dann in zwei Phasen. Zunächst wird die Konkurrenz zwischen den lokalen Experten verschärft, indem σ_y langsam verringert wird. Dadurch erfolgt eine Spezialisierung der lokalen Experten auf die jeweiligen funktionalen Zusammenhänge in der Datenfolge \mathcal{Z} . Da nach einer ersten Spezialisierung die Qualität der Funktionenapproximation durch das Mitlernen anderer Zielfunktionen verringert wird, muß in der zweiten Phase der Gedächtnisparameter τ schrittweise reduziert werden, bis schließlich jeder lokale Experte genau eine Zielfunktion approximiert. Ein so trainiertes mGRBF-System ist in der Lage, jede der in der multifunktionalen Datenfolge \mathcal{Z} enthaltenen Zielfunktionen in der Qualität eines GRBF-Netzwerks zu approximieren.

Abbildung 4.4 zeigt die Approximation der multifunktionalen Datenfolge aus Abbildung 4.3 mit einem mGRBF-System aus zwei GRBF-Netzwerken mit jeweils 20 Zwischenschichtneuronen. Bei dem in Kapitel 7 dargestellten neuen Regelungssystem für die Rückstandsverbrennung im Werk Burghausen wird das mGRBF-System dazu verwendet die

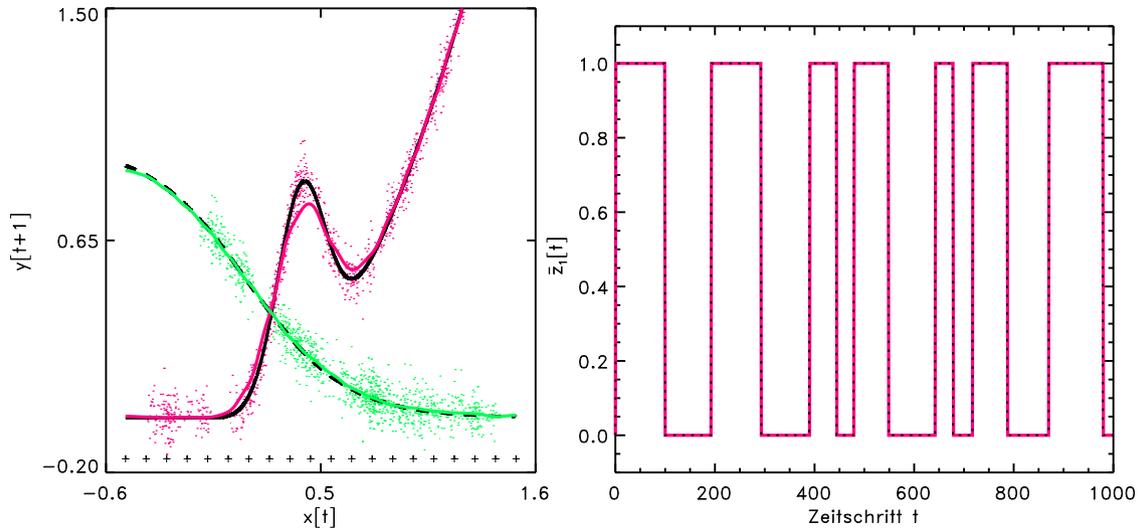


Abbildung 4.4.: **mGRBF-System für eine multifunktionale Datenfolge.** Das verwendete mGRBF-System ist aus zwei GRBF-Netzwerken mit jeweils 20 Zwischenschichtneuronen aufgebaut. Links sind die beiden Funktionenapproximationen $\hat{f}_1(x; \Theta_1^{GRBF})$ und $\hat{f}_2(x; \Theta_2^{GRBF})$ (rote und grüne Linie) der multifunktionalen Trainingsdaten \mathcal{Z} (rote und grüne Punkte) dargestellt. Die Lagen c_r der 20 Zentren sind durch die schwarzen Kreuze markiert. Das rechte Teilbild zeigt die Zugehörigkeit $z_1[t]$ für eine Testdatenfolge $\mathcal{X}^{\text{test}}$ (rote Linie). Diese stimmt nahezu perfekt mit der tatsächlichen Zustandsfolge $s[t]$ überein (schwarze Punkte).

Kennlinien von stark nichtlinearen, diskontinuierlichen Flüssigkeitsventilen zu approximieren. Damals (Oktober 1997) stand der LLM-Approximator noch nicht zur Verfügung, der ebenfalls zur Funktionenapproximation einer multifunktionalen Datenfolge \mathcal{Z} geeignet ist.

4.3.2. LLM-Approximator für eine multifunktionale Datenfolge

Die Funktionenapproximation einer multifunktionalen Datenfolge $\mathcal{Z} = \{\mathbf{x}[t], \mathbf{y}[t+1] \mid t = 1 \dots N-1\} \subset \mathbb{R}^{D+K}$ kann auch zwanglos mit einem LLM-Approximator gelöst werden, bei dem es sich selbstverständlich ebenfalls um ein System lokaler Experten handelt.

Auf Grund der dynamischen Struktur der Datenfolge \mathcal{Z} ist es nun allerdings notwendig, für die Optimierung der Parameter Θ^{LLM} eines LLM-Approximators mit einer anderen Datenfolge $\tilde{\mathcal{Z}}$ zu arbeiten. Die einfachste Wahl ist $\tilde{\mathcal{Z}} = \{\mathbf{x}[t], \mathbf{y}[t], \mathbf{y}[t+1] \mid t = 1 \dots N-1\}$. $\tilde{\mathcal{Z}}$ setzt sich aus der Eingabedatenfolge $\tilde{\mathcal{X}} = \{\mathbf{x}[t], \mathbf{y}[t] \mid t = 1 \dots N-1\} \subset \mathbb{R}^{D+K}$ und der Ausgabedatenfolge $\tilde{\mathcal{Y}} = \{\mathbf{y}[t+1] \mid t = 1 \dots N-1\} \subset \mathbb{R}^K$ zusammen. Die Eingabedatenfolge $\tilde{\mathcal{X}}$ umfaßt neben der aktuellen Eingabe $\mathbf{x}[t]$ auch die Systemantwort $\mathbf{y}[t]$ auf die Eingabe $\mathbf{x}[t-1]$. Da sich $\mathbf{x}[t]$ auf Grund der geforderten *random walk* Dynamik

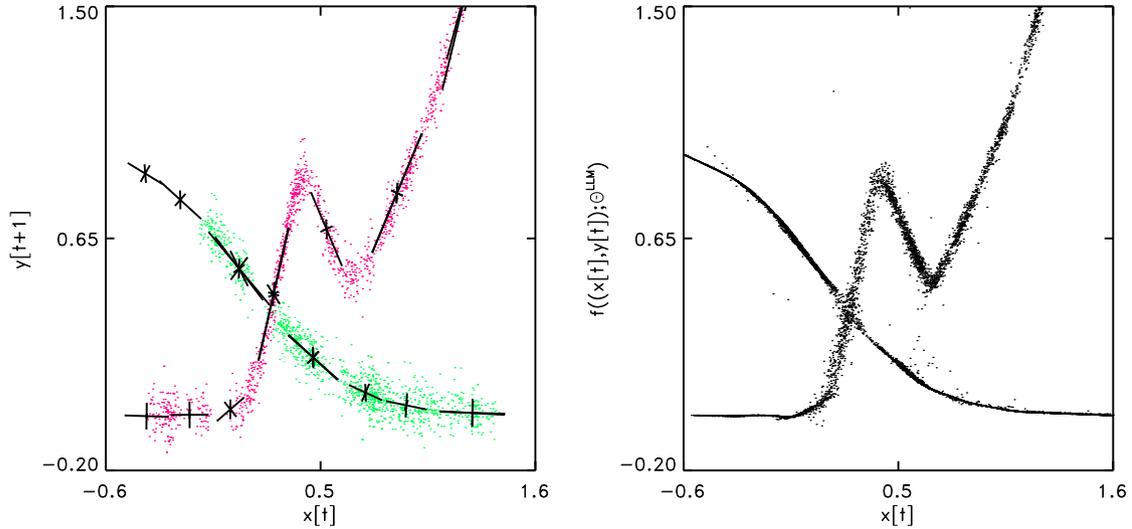


Abbildung 4.5.: **LLM-Approximator für eine multifunktionale Datenfolge.** Zur Funktionenapproximation des in Abbildung 4.3 gezeigten multifunktionalen Zusammenhangs wurde zunächst die Datenfolge $\tilde{\mathcal{Z}}^{\text{train}} = \{x[t], y[t], y[t+1] \mid t = 1 \dots N-1\}$ einer ML-Dichteschätzung mit einer Mischung aus 20 multivariaten Normalverteilungen unterzogen. Im linken Teilbild sind die daraus resultierenden dreidimensionalen lokalen Koordinatensysteme in einer Projektion auf die Ebene dargestellt, die durch die Achsen $x[t]$ und $y[t]$ aufgespannt wird. Nach der Konstruktion eines LLM-Approximators $\hat{f}(\tilde{\mathbf{x}}[t]; \Theta^{LLM})$ ergibt sich schließlich für eine Testdatenfolge $\tilde{\mathcal{X}}^{\text{test}} = \{x[t], y[t] \mid t = 1 \dots N-1\}$ die rechts gezeigte Funktionenapproximation.

nur langsam verändert, enthält ein Datenpunkt $\tilde{\mathbf{x}}[t] \equiv (\mathbf{x}[t], \mathbf{y}[t]) \in \tilde{\mathcal{X}}$ wegen

$$\mathbf{y}[t] \approx \mathbf{f}_{s[t]}(\mathbf{x}[t]) \quad (4.9)$$

nun auch die implizite Information darüber, in welchem Zustand $s[t]$ sich das System gerade befindet. Allerdings kann ein größerer Sprung in der Datenfolge \mathcal{X} einen Approximierungsfehler verursachen, da dann (4.9) nicht mehr erfüllt sein muß.

Mit einer solchen Datenfolge $\tilde{\mathcal{Z}}$ kann entsprechend dem Vorgehen aus Abschnitt 3.3 ein LLM-Approximator generiert werden, welcher die Funktionenapproximation

$$\tilde{\mathbf{y}}[t+1] = \hat{f}(\tilde{\mathbf{x}}[t]; \Theta^{LLM}) \quad (4.10)$$

ausführt.

Die beiden Abbildungen 4.5 und 4.6 veranschaulichen das Vorgehen bei der Konstruktion der Funktionenapproximation (4.10) für die Datenfolge aus Abbildung 4.3. In diesem Fall ist $\tilde{\mathcal{Z}} \subset \mathbb{R}^3$, $\tilde{\mathcal{X}} \subset \mathbb{R}^2$ und $\tilde{\mathcal{Y}} \subset \mathbb{R}$. In Abbildung 4.5, links, ist in einer zweidimensionalen Projektion gezeigt, wie sich die lokalen Modelle des LLM-Approximators in der dreidimensionalen Datenfolge $\tilde{\mathcal{Z}}$ verteilen. Abbildung 4.6 zeigt dieselben lokalen Modelle zusammen mit der Funktionenapproximation (4.10) durch den LLM-Approximator

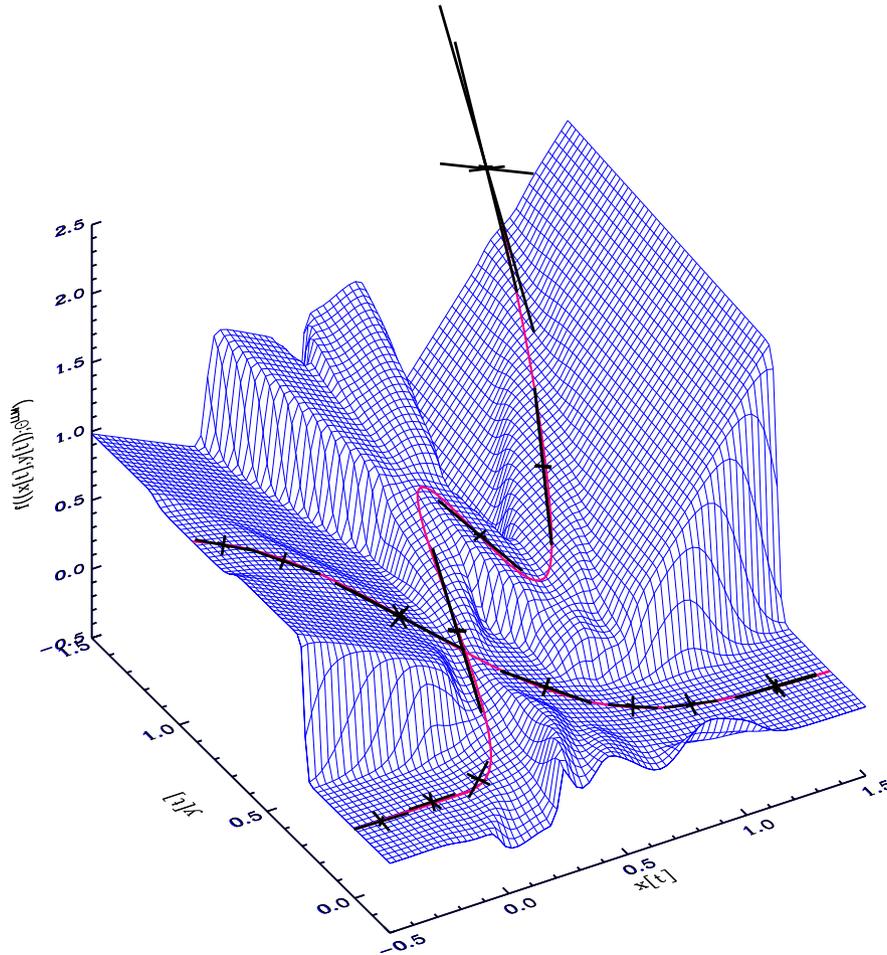


Abbildung 4.6.: **LLM-Approximator für eine multifunktionale Datenfolge.** Die Funktionenapproximation $\hat{f}(\tilde{x}[t]; \Theta^{LLM})$ aus Abbildung 4.5 ist hier in einer dreidimensionalen Darstellung gezeigt. Zusätzlich sind für die 20 Normalverteilungen die lokalen Koordinatensysteme und in rot die beiden Funktionen $f_1(x)$ und $f_2(x)$ eingetragen. Die Abbildung macht deutlich, wie sich beim LLM-Approximator die lokalen Experten nur auf Bereiche im Eingaberaum spezialisieren, in denen auch tatsächlich Daten vorliegen.

in einer dreidimensionalen Darstellung. Das rechte Teilbild der Abbildung 4.5 illustriert die Funktionenapproximation des LLM-Approximators für eine Testdatenfolge $\tilde{\mathcal{X}}^{\text{test}} = \{x[t], y[t] \mid t = 1 \dots N - 1\}$. Da der Komponente $y[t]$ ein kleiner Rauschanteil aufaddiert wurde (siehe auch Abbildung 4.3) erhält man keine glatte Funktionenapproximation. Das in diesen Abbildungen skizzierte Vorgehen kann natürlich auch auf komplexere, höherdimensionalere Datensätze übertragen werden. Entscheidend ist, daß der LLM-Approximator mit einer Eingabedatenfolge $\tilde{\mathcal{X}}$ trainiert wird, in der der Zustand $s[t]$ implizit kodiert ist.

Der LLM-Approximator unterscheidet sich in einer Reihe von Punkten vom mGRBF-System von M. Sonner:

- Das mGRBF-System erlaubt eine Zustandsdetektion von $s[t]$. Dies ist möglich, da jedes der L GRBF-Netzwerke über die Zuständigkeiten $\bar{z}_i[t]$ auf einen Zustand $s[t]$ zeigen kann. Eine solche Zustandsdetektion ist beim LLM-Approximator nicht in dieser expliziten Weise möglich.
- Das mGRBF-System benötigt nicht zwingend eine Eingabedatenfolge, die einer *random walk* Dynamik gehorcht. Diese langsame Dynamik der Eingabedaten, die der LLM-Approximator für die Erfüllung von (4.9) verlangt, ist aber in vielen realen Problemstellungen die generische Situation.
- M. Sonner hat das mGRBF-Netzwerk nur für gleichverteilte eindimensionale Eingabedaten untersucht. In hohen Dimensionen führt die äquidistante Positionierung der Zentren c_r zu sehr großen mGRBF-Systemen. Es ist auch nicht auf Anhieb klar, wie ein Training aller Parameter des mGRBF-Systems für eine Funktionenapproximation $\hat{f} : \mathbb{R}^D \rightarrow \mathbb{R}^K$ auszusehen hat.
- Im Gegensatz dazu ist der LLM-Approximator vollständig dichteorientiert. Nur in Raumgebieten, in denen Datenpunkte vorliegen, wird auch eine Funktionenapproximation durchgeführt. Daher sind für eine mit dem mGRBF-System vergleichbare Approximierungsqualität deutlich weniger Parameter erforderlich.

Zusammenfassung

Im ersten Teil dieses Kapitels wurde aus der formalen Struktur des LLM-Approximators, der auch als eine Verallgemeinerung des GRBF-Netzwerks aufgefaßt werden kann, eine Konstruktionsvorschrift für den Aufbau von Systemen lokaler Experten entwickelt. In den Abschnitten 4.3.1 und 4.3.2 wurden zwei Beispiele für Systeme lokaler Experten vorgestellt, die sich beide zur Funktionenapproximation multifunktionaler Datenfolgen eignen. Diese Aufgabenstellung hat nochmals die gute Approximierungsqualität des LLM-Approximators gezeigt. Auf Grund dieser günstigen Eigenschaften wird in Kapitel 6 der LLM-Approximator für die Regelung eines Bioreaktors eingesetzt, der sich durch eine nicht-lineare Dynamik und stochastische Zustandsänderungen auszeichnet. Auf das Konzept des Systems lokaler Experten werden wir in Kapitel 7 zurückkommen, da es sich dort als notwendig erweisen wird, verschiedene Strategien für die Regelung der Rückstandsverbrennung in einem Gesamtsystem zu integrieren.

5. Regelungstheorie

Wie schon in der Einleitung erklärt wurde, hat die vorliegende Arbeit das Ziel, aufbauend auf den Weiterentwicklungen des GRBF-Netzwerkmodells neue Konzepte zur Regelung industrieller Anlagen zu entwerfen. Komplementär zu den hierzu erforderlichen Grundlagen aus der Neuroinformatik, welche im 2. Kapitel referiert wurden, ist es daher nun notwendig, zunächst auch entsprechendes Basiswissen aus der Regelungstheorie einzuführen. Diese Kenntnisse sollen es ermöglichen, die in den Kapiteln 6 und 7 darzustellenden Aufgaben in der Sprache der Regelungstheorie zu beschreiben. Insbesondere wird im folgenden auf zwei etablierte Regelungsverfahren, den PID- und den *Fuzzy*-Regler, genauer eingegangen, da diese Methoden zur vergleichenden Evaluation der von mir vorgeschlagenen Verfahren verwendet werden.

5.1. Grundbegriffe der Regelungstheorie

Im Jahre 1963 begründete N. Wiener mit seinem Buch “Kybernetik” [Wie63] einen damals neuen Forschungszweig. Dieser beschäftigte sich mit der mathematischen Beschreibung und modellartigen Erklärung von dynamischen Systemen, welche durch die Rückführung von Informationen in einem Rückkopplungssystem geprägt sind. Der inzwischen etwas überholte Begriff der Kybernetik umfaßte damit Disziplinen wie die System-, die Nachrichten- und die Informationstheorie. Im besonderen Maße ist aber auch die heutige Regelungstheorie durch die Kybernetik beeinflusst worden.

Mit *Regelung* wird in Biologie und Technik der Vorgang bezeichnet, das zeitliche Verhalten eines Systems so zu beeinflussen, daß ein bestimmtes, erwünschtes Ziel erreicht wird [Föl92]. Wir hatten in der Einleitung mit der Beschreibung der Bewegungskoordination bei Lebewesen ein sehr komplexes Beispiel kennengelernt. Nun sind zwar technische Systeme von viel geringerer Komplexität, aber dennoch ist die Regelungstheorie im Leben des modernen Menschen von großer Bedeutung. Allerdings ist diese Tatsache häufig nicht offensichtlich, da die Regelungstheorie in der unauffälligen Arbeitsweise vieler effizienter technischer Geräte verborgen ist.

In der Regelungstheorie wird das zu regelnde, dynamische System als *Strecke* bezeichnet. Dabei werden auch die verschiedenen Meßeinrichtungen – die *Sensoren* – und die Stellglieder – die *Aktoren* – zur Strecke gezählt. Über die *Stellgrößen* $\mathbf{x}(t) \in \mathbb{R}^D$ werden die

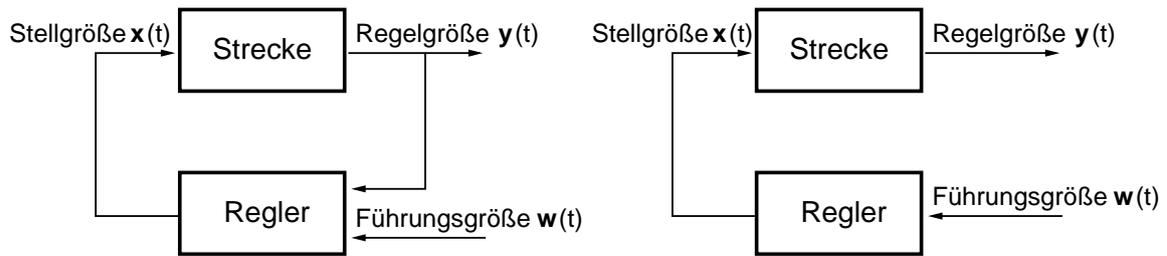


Abbildung 5.1.: Geschlossener und offener Regelkreis.

Veränderungen der an den D Aktoren einzustellenden Werte vorgegeben. Die *Regelgröße* $y(t) \in \mathbb{R}^K$ liefert Informationen über die Signale der K Sensoren und macht somit Aussagen über den gegenwärtigen Zustand der Strecke. Die Aufgabe des Reglers besteht nun darin, die Strecke so zu beeinflussen, daß die Regelgröße $y(t)$ möglichst gut mit einer vorgegebenen *Führungsgröße* $w(t) \in \mathbb{R}^K$ übereinstimmt. Häufig wird dazu eine *Führungsgrößenabweichung* $e(t) \equiv w(t) - y(t)$ eingeführt. Dabei beschreibt $w(t)$ die Zieltrajektorie, die das System auf Grund der Regelung durchlaufen soll. Das Gesamtsystem aus Strecke und Regler wird auch als *Regelkreis* bezeichnet.

Abbildung 5.1 zeigt zwei grundsätzlich verschiedene Methoden, eine technische Regelung zu realisieren. Links ist das Schema eines rückgekoppelten Regelkreises dargestellt. Bei einem solchen System generiert der Regler an Hand von Führungs- und Regelgrößen eine geeignete Stellgröße. Als einfaches Beispiel läßt sich hier die Regelung der Raumtemperatur durch einen Menschen nennen. Abhängig von seinem jeweiligen Befinden ist ihm eine bestimmte Raumtemperatur (Führungsgröße) erwünscht. Er vergleicht diese Wunschtemperatur mit der aktuellen Temperatur (Regelgröße) und verstellt dementsprechend das Heizkörperventil (Stellgröße). Bei dem Gesamtsystem Mensch-Heizung handelt es sich also um einen geschlossenen, rückgekoppelten Regelkreis. Wenn der Regler jedoch keine Informationen zur Regelgröße hat, spricht man von einem offenen, bzw. nicht rückgekoppelten Regelkreis. Dieser auch als *Steuerung* bezeichnete Vorgang wird häufig bei der Ablaufkontrolle von Geräten angewendet.

Die Stellgröße $x(t) \in \mathbb{R}^D$, die Regelgröße $y(t) \in \mathbb{R}^K$ und die Führungsgröße $w(t) \in \mathbb{R}^K$ können skalare oder vektorielle Größen sein. Gilt $D = K = 1$ spricht man von einem *single input-single output* (SISO) Regelkreis. Bei $D \neq 1$ und $K \neq 1$ wird ein *multiple input-multiple output* (MIMO) Regler notwendig. Entsprechend sind auch MISO- und SIMO-Regler zu verstehen.

Nach welchen Prinzipien kann nun ein Regler konzipiert werden? Für die Konstruktion eines funktionstüchtigen Reglers ist es unabdingbar, daß die zu regelnde Strecke möglichst gut bekannt ist. Nach der Art der Repräsentation dieses Wissens können drei grundlegend verschiedene Reglerklassen unterschieden werden:

- Bei der **modellbasierten Regelung** wird die Strecke durch ein analytisches Modell beschrieben. Der weitverbreitete Proportional-Integral-Differential-Regler (PID-

Regler) ist ein Vertreter dieses Prinzips und wird in Abschnitt 5.2 ausführlicher erläutert. Der PID-Regler beruht auf einer Modellbeschreibung durch eine lineare Differentialgleichung. Neuere Reglertypen, wie z.B. der Kalman-Filter [Kal60, Sor70] oder der *self-tuning regulator* [AW73, MSW92] versuchen die Qualität der Modellbeschreibung – und damit auch des Reglers – durch die Verwendung von nicht-linearen Differentialgleichungen zu verbessern.

- Die **erfahrungsbasierte Regelung** ist dann notwendig, wenn die Strecke analytisch sehr schwer oder gar nicht beschreibbar ist, aber durch die manuelle Regelung der Strecke ein großer Erfahrungsschatz verfügbar ist. Diese von Menschen gewonnene Erfahrung kann dann z.B. mit Hilfe von wissensbasierten Expertensystemen [Sch86] oder der *Fuzzy*-Logik [Zim93, DeS95] so formalisiert werden, daß das ursprüngliche Vorgehen bei der manuellen Regelung durch eine entsprechende Automatik kopiert wird. In Abschnitt 5.3 wird die Arbeitsweise eines *Fuzzy*-Reglers beschrieben.
- **Datenbasierte Regelung** ist prädestiniert für Fälle, bei denen weder ein analytisches Modell der Strecke existiert, noch eine formalisierbare menschliche Erfahrung zur Regelung dieser Strecke vorliegt. Für die Parametrisierung solcher Regler sind Meßdaten notwendig, die das Verhalten der Strecke in verschiedenen Situationen und über einen längeren Zeitraum erfassen. Ferner wird ein Verfahren benötigt, welches es gestattet, die Parameter aus den Daten zu extrahieren. Daher bieten sich neuronale Netze, die datenbasierte Funktionenapproximationen ermöglichen, als Regelungswerkzeuge an.

Die meisten Methoden der klassischen Regelungstheorie sind den modellbasierten Verfahren zuzuordnen [Föl92]. Auf Grund methodischer Fortschritte und der kostengünstigen Verfügbarkeit hoher Rechenleistung¹ sind die numerisch aufwendigeren erfahrungsbzw. datenbasierten Regelungsverfahren in den letzten Jahrzehnten aber zunehmend aktueller geworden. Anzumerken ist, daß bei praktischen Lösungen die obige Einteilung nicht eindeutig greift, da dort häufig verschiedene Methoden zu hybriden Verfahren kombiniert werden.

5.2. Modellbasierte Regelung mit einem PID-Regler

Viele technische Strecken lassen sich nach Einführung eines *Zustandsvektors* $\mathbf{z} \in \mathbb{R}^Z$ durch ein lineares Differentialgleichungssystem 1. Ordnung mit konstanten Koeffizienten modellieren. Der Zustandsvektor \mathbf{z} umfaßt alle für eine Beschreibung der Strecke relevanten Größen, für die aber nicht unbedingt eine Messung vorliegt. Für den Fall eines

¹Universell einsetzbare und programmierbare *micro-controller* werden in sehr hohen Stückzahlen produziert und sind seit etwa 10 Jahren so preisgünstig, daß sich der Einsatz auch in alltäglichen Geräten lohnt.

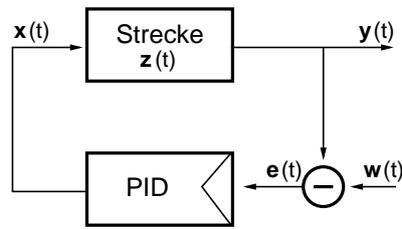


Abbildung 5.2.: Schema eines PID-Reglers.

SISO-Regelkreises (d.h. $D = K = 1$) kann eine solche Strecke dann durch

$$\frac{d}{dt}\mathbf{z}(t) = \mathbf{A} \cdot \mathbf{z}(t) + \mathbf{b}x(t) \quad (5.1)$$

mit den zeitunabhängigen Parametern $\mathbf{A} \in \mathbb{R}^{Z \times Z}$ und $\mathbf{b} \in \mathbb{R}^D$ beschrieben werden. In vielen Fällen kann die Regelgröße $y(t)$ als linear abhängig vom Zustandsvektor $\mathbf{z}(t)$ angenommen werden. Dieses Verhalten wird durch die algebraische Gleichung

$$y(t) = \mathbf{c}^T \cdot \mathbf{z}(t) \quad (5.2)$$

mit $\mathbf{c} \in \mathbb{R}^Z$ modelliert [Lud89].

Die meisten in der Industrie eingesetzten Standardregler für eine Strecke, die annähernd einer Beschreibung durch (5.1) und (5.2) gehorcht, sind lineare Regler [Czi89]. Der wichtigste Vertreter dieser Reglerklasse ist der PID-Regler [Föl92]

$$x(t) = K_P \left[e(t) + \frac{1}{T_N} \int_0^t e(\tau) d\tau + T_V \frac{d}{dt} e(t) \right], \quad (5.3)$$

wobei hier der Fall einer SISO-Strecke angenommen wird. Abbildung 5.2 zeigt einen Regelkreis mit einem solchen PID-Regler. Der PID-Regler berechnet eine Stellgröße $x(t)$ auf Grundlage von drei Termen:

- Der **P-Anteil** $K_P e(t)$ ist proportional zur Abweichung $e(t)$. Allein mit dem P-Anteil gelingt es aber nicht, eine Strecke auf eine Führungsgröße $w(t)$ einzuregeln, da immer eine Abweichung $e(t) \neq 0$ verbleiben wird [Föl92].
- Erst der **I-Anteil** $K_P/T_N \int_0^t e(\tau) d\tau$ vermag in Verbindung mit dem P-Anteil die Abweichung $e(t)$ zum Verschwinden zu bringen, indem die Stellgröße $x(t)$ proportional zur Summe der bis zum aktuellen Zeitpunkt erfolgten Abweichungen berechnet wird. Bei einer konstanten Abweichung $e(t) = 1$ entspricht der I-Anteil zum Zeitpunkt der *Nachhaltezeit* T_N gerade dem Beitrag des P-Anteils. Die nur langsam einsetzende Wirkung des I-Anteil führt zu einem Überschwingen der Stellgröße $x(t)$ und damit auch der Führungsgröße $y(t)$ [Föl92].

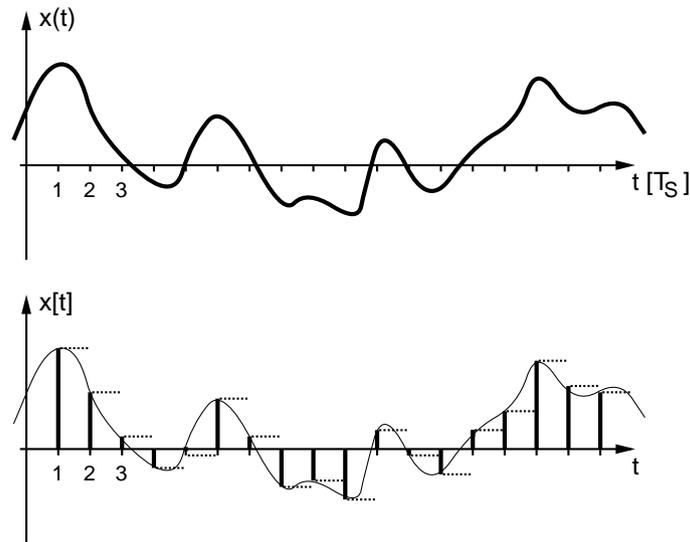


Abbildung 5.3.: **Zeitdiskretisierung**. Das oben gezeigte zeitkontinuierliche Signal $x(t)$ wird mit einer Abtastperiode T_S abgetastet. Das untere Bild zeigt das so resultierende zeitdiskrete Signal $x[t]$ mit $t \in \mathbb{N}$. Die punktierten Linien deuten an, daß der Wert von $x[t]$ bis zum nächsten Abtastzeitpunkt als konstant angenommen wird.

- Beim **D-Anteil** $K_P T_V de(t)/dt$ ist die Neigung zum Überschwingen geringer als beim I-Anteil. Es stellt sich auch hier eine bleibende Abweichung $e(t)$ ein, die aber geringer ist als beim P-Anteil. Da der D-Anteil bei jedem Führungsgrößenprung zu einer stoßartigen Stellgröße $x(t)$ führt, kann der D-Anteil in der Praxis nur mit einer Tiefpaßfilterung vor der Differenzierung verwendet werden [Lud89]. Zur *Vorhaltezeit* T_V ist der Gleichstand zwischen D-Anteil und P-Anteil erreicht, wenn $e(t) = t$ gilt.

Bei geeigneter Kombination der P-, I-, und D-Anteile erhält man einen PID-Regler, der ein geringes Überschwingen aufweist und zu einer verschwindenden Führungsgrößenabweichung $e(t)$ führt [Lud89]. Das zentrale Entwurfproblem bei einem PID-Regler ist daher die Wahl der Parameter K_P , T_N und T_V . Ist die Strecke annähernd durch (5.1) beschreibbar, dann ist die Parameterbestimmung relativ einfach [Föl92, TCA71]. Ist allerdings die Strecke nicht analytisch genau bekannt, wird die Festlegung der PID-Parameter im wesentlichen zu einem *trial and error* Problem. Kommen noch Nichtlinearitäten, zeitlich variable Systemparameter und – für den Fall eines MIMO-Regelkreises – starke Wechselwirkungen zwischen den verschiedenen Stellgrößen hinzu, wird die Aufgabe zunehmend schwieriger. Für diese Situationen sind Verfahren notwendig, die die PID-Parameter kontinuierlich an die Erfordernisse neu anpassen. Eine der vielen Möglichkeiten ist hier z.B. der *self-tuning regulator* [AW73, MSW92].

Für die praktische Realisierung durch einen Algorithmus kann der PID-Regler nicht in der obigen zeitkontinuierlichen Form (5.3) verwendet werden. Vielmehr müssen zunächst

die zeitkontinuierlichen Größen diskretisiert werden. Abbildung 5.3 veranschaulicht diesen Vorgang. Bei einer solchen *Abtastung* wird mit einer *Abtastperiode* T_S die zeitkontinuierliche Größe $x(t)$ zu den Zeitpunkten kT_S , $k \in \mathbb{N}$, ausgelesen und der zeitdiskreten Größe $x[k] = x(kT_S)$ zugewiesen. Es handelt sich hierbei um eine eindeutige, aber nicht umkehrbare Abbildung. Der Vorgang der Abtastung wird im Folgenden durch die Schreibweise $x[k]$ bzw. $x[t]$ mit $k, t \in \mathbb{N}$ angedeutet. Eine Abtastung ist für alle Regler notwendig, die auf einer Informationsverarbeitung durch digitale Rechner beruhen.

Nach [DL89] ergibt sich nach einer solchen Abtastung für den PID-Regler der folgende diskrete Regelungsalgorithmus:

0. Initialisierung

$$d_I = K_P \frac{T_S/2}{T_N}, \quad c_D = \frac{T_D - T_S/2}{T_D + T_S/2}, \quad d_D = K_P \frac{T_V}{T_D + T_S/2}$$

1. Einlesen von $w[t]$ und $y[t]$

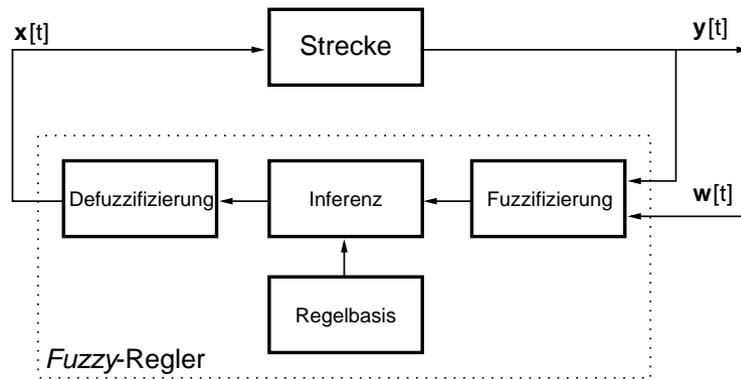
2. Regelungsschritt

$$\begin{aligned} e[t] &= w[t] - y[t] \\ x_P[t] &= K_P e[t] \\ x_I[t] &= x_I[t-1] + d_I(e[t] + e[t-1]) \\ x_D[t] &= c_D x_D[t-1] + d_D(e[t] - e[t-1]) \\ x[t] &= x_P[t] + x_I[t] + x_D[t] \end{aligned}$$

3. Ausgeben von $x[t]$

4. Abwarten der Abtastperiode T_S : $t \leftarrow t + 1$ Neuen Regelungsschritt mit 1. beginnen.

In diesem diskreten PID-Algorithmus gibt es zusätzlich zur Nachhaltezeit T_N und Vorhaltezeit T_V noch eine Dämpfungszeit T_D . Dieser Parameter charakterisiert die für den D-Anteil notwendige Tiefpaßfilterung der Führungsgrößenabweichung $e[t]$. Besonders für Führungsgrößen $y[t]$, die stark mit einem Rauschen behaftet sind, ist eine solche Dämpfung wichtig, da sonst das Rauschen zu einer Destabilisierung des Regelkreises führt [Gee94]. Die Abtastperiode T_S eines solchen diskreten PID-Reglers sollte etwa 10 mal kleiner sein als die kleinste Zeitkonstante, die für die Strecke von Bedeutung ist [Lud89].

Abbildung 5.4.: Prinzip eines *Fuzzy*-Reglers.

5.3. Erfahrungsbasierte Regelung mit einem *Fuzzy*-Regler

Konventionelle Regelungstechniken, wie der PID-Regler, haben viele Vorteile. Wenn die zu Grunde liegenden Annahmen erfüllt und die Parameter korrekt eingestellt sind, dann garantieren diese Verfahren eine optimale Regelung; häufig weisen sie sogar eine hohe Robustheit gegenüber kleineren Störungen und Modellunsicherheiten auf [DeS95, Lud89]. Es sind aber auch einige Nachteile vorhanden. Die Methoden sind relativ unflexibel und erlauben keine Grobeinstellung der Parameter auf Basis von unvollständigen oder unsicheren Informationen. Auch können solche modellbasierten Verfahren nicht auf qualitative menschliche Erfahrungen zurückgreifen, die durch eine manuelle Regelung der Strecke gesammelt wurden. Erfahrungsbasierte Techniken, wie z.B. der *Fuzzy*-Regler, bieten hier eine Alternative zu modellbasierten Regelungen.

Abbildung 5.4 zeigt den prinzipiellen Aufbau eines *Fuzzy*-Reglers. Die Idee eines *Fuzzy*-Reglers, wie auch der *Fuzzy*-Logik, besteht darin, zunächst eine *Regelbasis* aufzustellen und an Hand dieser Regeln Entscheidungen zu treffen. Es handelt sich dabei aber nicht um scharf definierte Handlungsvorschriften, sondern um unscharfe, sog. *fuzzy* Regeln. Eine solche Regel könnte z.B. lauten: *Wenn die Temperatur sehr hoch ist, dann schließe das Ventil schnell*. Um auf der Grundlage solcher Regeln Entscheidungen treffen zu können, ist zunächst eine *Fuzzifizierung* der Eingabedaten $y[t]$ und $w[t]$ notwendig. Dazu wird eine scharf definierte Größe (z.B. der Meßwert einer Temperatur) einer unscharfen Menge (z.B. *sehr hohe Temperatur*) zugeordnet. Damit trägt das Verfahren möglichen Meßfehlern und Unsicherheiten bei der Bestimmung der Eingabedaten Rechnung. Die eigentliche Entscheidungsfindung erfolgt durch die *kompositorische Inferenzregel* [Zad73]. Abschließend muß noch eine *Defuzzifizierung* vorgenommen werden, um zu scharf definierten Ausgaben zu kommen, die als Stellgrößen $x[t]$ verwendet werden können.

In den folgenden drei Teilabschnitten wird ein *Fuzzy*-System nach E. Mamdani beschrieben [MG87]. Seine Arbeitsweise wird – parallel zu den folgenden Ausführungen – in Ab-

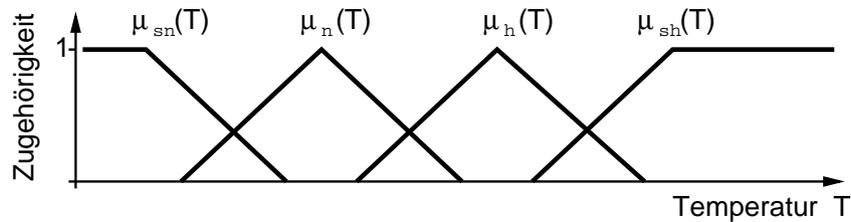


Abbildung 5.5.: Zugehörigkeitsfunktionen.

Abbildung 5.6 graphisch illustriert. Dabei handelt es sich um keine umfassende Diskussion, sondern um eine knappe Darstellung der wichtigsten Elemente eines *Fuzzy*-Reglers. Für eine eingehende Diskussion von Verfahren der *Fuzzy*-Theorie sei auf die entsprechende Literatur verwiesen [Zad73, Zim93, Gra95, DeS95].

Fuzzifizierung

Die Fuzzifizierung ordnet eine scharf definierte Größe x einer *fuzzy* Menge A zu. Sei X der Wertebereich der Größe x , d.h. sei $x \in X$. Dann ist eine *fuzzy* Menge A ein Teilbereich von X , dessen Grenzen – im Unterschied zu klassischen Mengen – nicht scharf, sondern unscharf definiert sind. Das Maß der Zugehörigkeit von x zur *fuzzy* Menge A wird durch eine Zugehörigkeitsfunktion $\mu_A(x)$ ausgedrückt, welche die Elemente x aus X auf einen Wert aus dem Intervall $[0, 1]$ abbildet, d.h.

$$\mu_A : x \in X \rightarrow \mu_A(x) \in [0, 1]. \quad (5.4)$$

Der Wert $\mu_A(x) = 0$ bedeutet dabei, daß x nicht in der *fuzzy* Menge A enthalten ist. Für Werte zwischen 0 und 1 ist eine partielle Zugehörigkeit zu A gegeben und für $\mu_A(x) = 1$ ist x auf jeden Fall Element von A , was in beiden Fällen durch $x \in A$ gekennzeichnet wird. Für die analytische Form von $\mu_A(x)$ sind die verschiedensten Ausdrücke vorgeschlagen worden. In dieser Arbeit wird die Zugehörigkeitsfunktion aber immer durch die – auch in der Literatur sehr üblichen – Trapez-, bzw. Dreiecksfunktionen dargestellt [Gra95]. Es sei angemerkt, daß eine Menge $\{A_i\}$ von *fuzzy* Mengen A_i genau dann eine unscharfe Zerlegung von X bildet, wenn gilt:

$$\sum_i \mu_{A_i}(x) = 1, \quad \forall x \in X. \quad (5.5)$$

In diesem Fall lassen sich die Zugehörigkeitsfunktionen als bedingte Wahrscheinlichkeiten $P(A_i|x) \equiv \mu_{A_i}(x)$ der Zuordnung von x zu heuristisch definierten Klassen A_i auffassen [vgl. auch Gleichung (2.43)].

Abbildung 5.5 veranschaulicht das Vorgehen an einem Beispiel. Der Wertebereich X ist die Menge aller möglichen Temperaturen, die durch einen Meßvorgang ermittelt werden

können. Innerhalb von X werden heuristisch vier *fuzzy* Mengen sn, n, h und sh definiert, die den unscharfen Begriffen *sehr niedrige*, *niedrige*, *hohe* und *sehr hohe Temperatur* zugeordnet sind. In der Abbildung sind die Zugehörigkeitsfunktionen $\mu_{sn}(x)$, $\mu_n(x)$, $\mu_h(x)$ und $\mu_{sh}(x)$ dieser vier Mengen gezeigt. Für sehr niedrige bzw. sehr hohe Temperaturen nehmen die Zugehörigkeitsfunktionen $\mu_{sn}(x)$ und $\mu_{sh}(x)$ den konstanten Wert Eins an, so daß auch extreme kleine und große Temperaturen erfaßt werden.

Bekannterweise können auf den scharf definierten Mengen der klassischen Mengenlehre Operationen der zweiwertigen Logik ausgeführt werden [BS91]. In der *Fuzzy*-Logik werden diese Operationen verallgemeinert und durch Zugehörigkeitsfunktionen entsprechender *fuzzy* Mengen ausgedrückt. Im folgenden werden jene Verallgemeinerungen zweiwertiger logischer Operationen dargestellt, die für einen *Fuzzy*-Regler benötigt werden. In der Literatur finden sich eine ganze Reihe verschiedener Formulierungen [Gra95]. Hier ist daher nur die gebräuchlichste beschrieben [DeS95]. Dabei seien A, B *fuzzy* Mengen aus dem Wertebereich X und C eine *fuzzy* Menge aus dem Wertebereich Y .

- Das **Komplement** \bar{A} einer *fuzzy* Menge A ist gegeben durch die Zugehörigkeitsfunktion

$$\mu_{\bar{A}}(\mathbf{x}) = 1 - \mu_A(\mathbf{x}). \quad (5.6)$$

- Die **Disjunktion** $A \cup B$ zweier *fuzzy* Mengen A und B ist definiert durch die Zugehörigkeitsfunktion

$$\mu_{A \cup B}(\mathbf{x}) = \max\{\mu_A(\mathbf{x}), \mu_B(\mathbf{x})\}. \quad (5.7)$$

- Die **Konjunktion** $A \cap B$ zweier *fuzzy* Mengen A und B ist definiert durch

$$\mu_{A \cap B}(\mathbf{x}) = \min\{\mu_A(\mathbf{x}), \mu_B(\mathbf{x})\}. \quad (5.8)$$

- Die *fuzzy* **Implikation** $A \rightarrow C$ ist eine Abbildung der beiden *fuzzy* Mengen A und C auf eine *fuzzy* Menge $A \rightarrow C$ aus dem kartesischen Produkt $X \times Y$, die üblicherweise durch

$$\mu_{A \rightarrow C}(\mathbf{x}, \mathbf{y}) = \min\{\mu_A(\mathbf{x}), \mu_C(\mathbf{y})\} \quad (5.9)$$

definiert wird [DeS95]. Da diese Operation im Gegensatz zur Implikation der zweiwertigen Logik kommutativ ist, ist (5.9) im Grunde nicht befriedigend. Verwendet man sie jedoch im Zusammenhang von wenn-dann-sonst-Strukturen, die in der *Fuzzy*-Logik durch wenn A dann B sonst $C \equiv (A \rightarrow B) \cup (\bar{A} \rightarrow C)$ gegeben sind, so liefert sie dort ähnliche Resultate wie die entsprechenden Strukturen der binären Logik [DeS95].

Regelbasis und kompositorische Inferenz

Am Anfang der Konstruktion eines *Fuzzy*-Reglers steht die Entwicklung einer geeigneten Regelbasis \mathcal{R} auf Grundlage einer sprachlichen Beschreibung der Kontrollregeln durch einen menschlichen Experten. Dieser Schritt ist analog zur Identifikation der Parameter bei der Entwicklung eines modellbasierten Reglers. Die Regelbasis ist aus einer Reihe von wenn-dann-Regeln aufgebaut, welche die Eingaben y und w in Bezug zur Stellgröße x setzen. Während der Regelung wird die Regelbasis mit Hilfe der kompositorischen Inferenzregel periodisch auf die aktuellen Eingabegrößen angewendet und so die Stellgröße bestimmt. Dieses Vorgehen wird nun für den Fall eines SISO-Reglers näher erläutert (vgl. auch Abbildung 5.6)

Y_i , W_j und X_k seien *fuzzy* Mengen auf den Wertebereichen der Regelgröße y , Führungsgröße w und Stellgröße x . Da für jede Kombination von Y_i und W_j eine *fuzzy* Stellgrößenmenge X_k vorhanden sein muß, ist der Index k eine Funktion von (i, j) , d.h. $k \equiv k(i, j)$. Eine Regel $R_{k(i,j)}$ aus der Regelbasis \mathcal{R} ist dann eine *fuzzy* Menge im Produktraum $Y \times W \times X$ und kann als

$$R_{k(i,j)} : \quad \text{wenn } (y \in Y_i) \text{ und wenn } (w \in W_j) \text{ dann } (x \in X_{k(i,j)}) \quad (5.10)$$

formuliert werden. Die Ausdrücke vom Typ $(y \in Y_i)$ werden dabei durch die entsprechenden Zugehörigkeitsfunktionen $\mu_{Y_i}(y)$ quantifiziert. Da die Regel (5.10) aus einer Konjunktion und einer Implikation aufgebaut ist, kann die Regel $R_{k(i,j)}$ auch mit Hilfe der beiden Vorschriften (5.8) und (5.9) ausgedrückt werden,

$$\mu_{R_{k(i,j)}}(y, w, x) = \min\{\mu_{Y_i}(y), \mu_{W_j}(w), \mu_{X_{k(i,j)}}(x)\}, \quad (5.11)$$

welche die *fuzzy* Menge $R_{k(i,j)}$ definiert.

Alle Regeln aus \mathcal{R} sind durch Disjunktionen (5.7) verknüpft, so daß \mathcal{R} selbst eine *fuzzy* Menge ist. Daher kann die Zugehörigkeitsfunktion $\mu_{\mathcal{R}}(y, w, x)$ der gesamten Regelbasis \mathcal{R} auch aus den Zugehörigkeitsfunktionen $\mu_{R_{k(i,j)}}(y, w, x)$ der Einzelregeln berechnet werden [DP80, DeS95]:

$$\mu_{\mathcal{R}}(y, w, x) = \max_{i,j} \{\mu_{R_{k(i,j)}}(y, w, x)\}. \quad (5.12)$$

Mit Hilfe von (5.12) kann also das Maß der Zugehörigkeit eines Wertetripels (y, w, x) zur Regelbasis \mathcal{R} aus den Zugehörigkeitsfunktionen von Y_i , W_j und $X_{k(i,j)}$ bestimmt werden. Die Zugehörigkeitsfunktion $\mu_{\mathcal{R}}(y, w, x)$ ist im allgemeinen in einem hochdimensionalen Raum definiert und beinhaltet das gesamte *fuzzy* Regelwerk.

Ist einmal das *fuzzy* Regelwerk als *fuzzy* Menge \mathcal{R} definiert, so benötigt man noch einen Algorithmus, um aus der aktuellen Führungsgröße $w[t]$ und der Regelgröße $y[t]$ auf eine Stellgröße $x[t]$ zu schließen. Dazu bemerkt man, daß die Zugehörigkeitsfunktion (5.12)

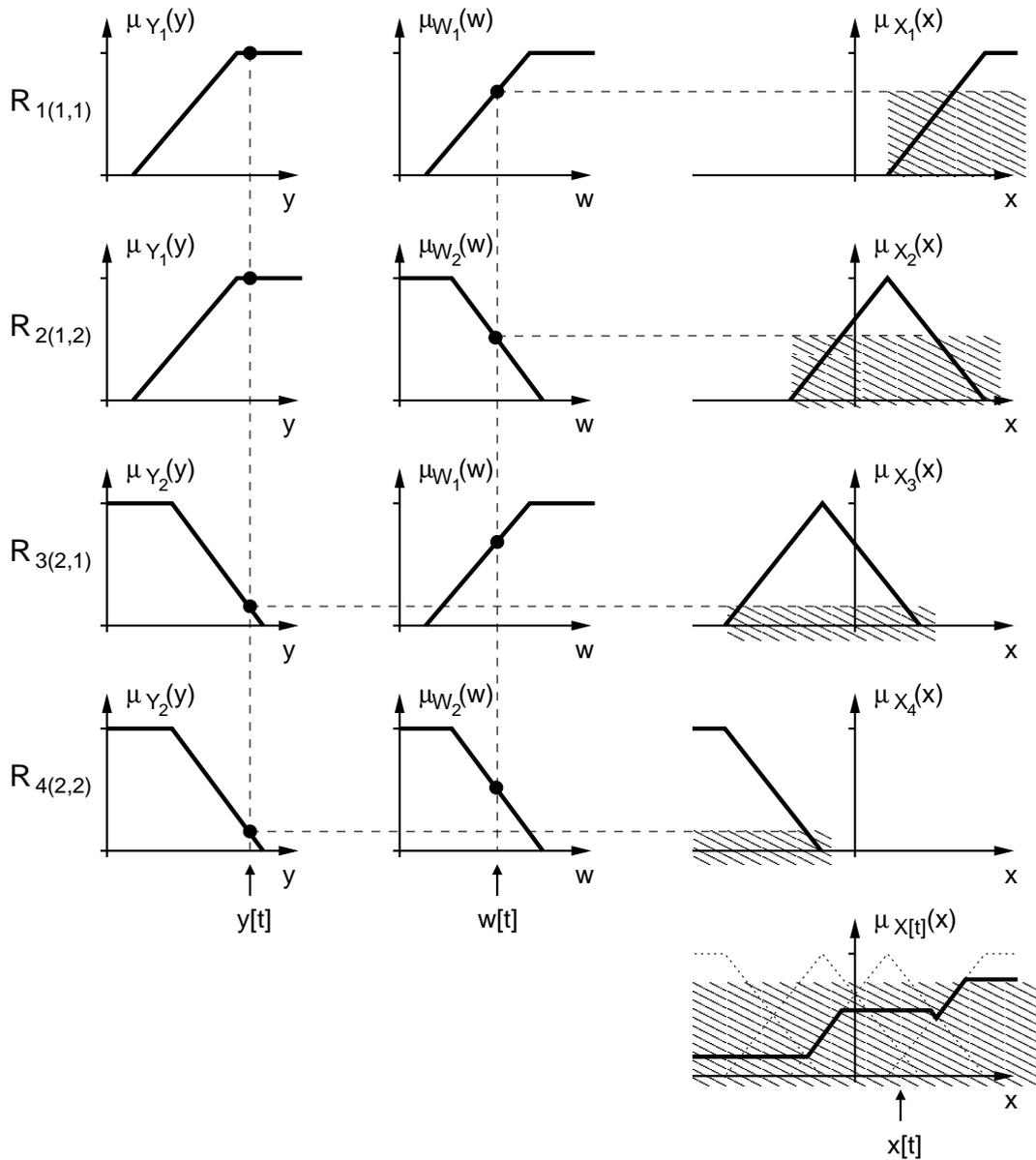


Abbildung 5.6.: **Kompositorische Inferenzregel.** Skizziert wird das Vorgehen für einen *fuzzy* SISO-Regelkreis, für den vier *fuzzy* Regeln $R_{k(i,j)}$ aufgestellt worden sind (Zeilen 1–4). Entsprechend wurden die Regel- und Führungsgrößen mit jeweils zwei Zugehörigkeitsfunktionen fuzzifiziert, die in den beiden linken Spalten dargestellt sind. Die Zugehörigkeitsfunktionen der entsprechenden vier Stellgrößen sind in der rechten Spalte gezeigt. Für eine Situation $y[t], w[t]$ werden zunächst die durch (5.11) definierten $\mu_{R_{k(i,j)}}(y, w, x)$ der Regeln $R_{k(i,j)}$ auf Zugehörigkeitsfunktionen $\mu_{R_{k(i,j)}}(y[t], w[t], x)$ eingeschränkt (schraffierte Funktionen in der rechten Spalte). Die Ausführung der kompositorischen Inferenzregel (5.12) führt zur Zugehörigkeitsfunktion $\mu_{X[t]}(x)$ (unten rechts). Die Defuzzifizierung nach der Schwerpunktmethode (5.15) liefert schließlich die exakte Stellgröße $x[t]$.

bei vorgegebenen Werten $y[t]$ und $w[t]$ die Regelbasis $\mathcal{R} \subset Y \times W \times X$ auf eine *fuzzy* Stellgrößenmenge $X[t] \subset X$ einschränkt, deren Zugehörigkeitsfunktion durch

$$\mu_{X[t]}(x) = \mu_{\mathcal{R}}(y[t], w[t], x) \quad (5.13)$$

gegeben ist. Im Rahmen der *Fuzzy*-Logik ist diese *fuzzy* Menge $X[t]$ das eigentliche Ergebnis der Anwendung der Regeln $R_{k(i,j)}$. Wegen (5.11) und (5.12) kann sie auch gemäß

$$\mu_{X[t]}(x) = \max_{i,j} [\min\{\mu_{Y_i}(y[t]), \mu_{W_j}(w[t]), \mu_{X_{k(i,j)}}(x)\}] \quad (5.14)$$

aus den einzelnen Regeln berechnet werden. Die Auswertung von (5.14) wird als kompositorische Inferenzregel bezeichnet und wird in Abbildung 5.6 an einem Beispiel illustriert. Zur Festlegung der einzustellenden Größe $x[t]$ muß dieses Ergebnis anschließend noch defuzzifiziert werden.

Defuzzifizierung

Hierzu sind verschiedene Verfahren vorgeschlagen worden [Gra95]. Das üblichste ist die Schwerpunktmethode, bei der der Schwerpunkt der Zugehörigkeitsfunktion $\mu_{X[t]}(x)$ als exakter Wert für die Stellgröße verwendet wird:

$$x[t] = \frac{\int x \mu_{X[t]}(x) dx}{\int \mu_{X[t]}(x) dx}. \quad (5.15)$$

$x[t]$ wird als Stellgröße an die Strecke weitergegeben. Dieses Defuzzifizierungs-Verfahren führt – im Gegensatz zu anderen Verfahren – zu einer etwas trägen aber recht robusten Regelung [DeS95]. Abbildung 5.6 illustriert das hier skizzierte Vorgehen an einem Beispiel.

Zwar läßt sich das Konzept der *Fuzzy*-Regelung auf MIMO-Regler verallgemeinern. In diesem Fall müssen aber wesentlich mehr Zugehörigkeitsfunktionen und Regeln definiert werden, so daß die Erarbeitung eines konsistenten und sicher arbeitenden Regelwerks zu einer sehr zeitaufwendigen und komplizierten Aufgabe werden kann.

Fuzzy-Regler nach M. Sugeno

Für die Evaluation von verschiedenen Regelungsaufgaben in Kapitel 6 und 7 wurde ein *Fuzzy*-Regler nach M. Sugeno eingesetzt, der einen Spezialfall des oben dargestellten Mamdani *Fuzzy*-Reglers darstellt [Sug88]. Bei einem Sugeno *Fuzzy*-Regler werden die Zugehörigkeitsfunktionen der Stellgröße x in (5.10) durch *fuzzy singletons* $\mu_{X_{k(i,j)}}(x) = \delta(x - x_{k(i,j)})$ beschrieben, wobei $X_{k(i,j)}$ eine entartete Menge ist, die genau den Wert $x_{k(i,j)}$ umfaßt. Wird mit dieser Zugehörigkeitsfunktion die Defuzzifizierung (5.15) ausgeführt, so erhält man den Ausdruck

$$x[t] = \frac{\sum_k x_k \mu_{R_k}(y[t], w[t])}{\sum_{k'} \mu_{R_{k'}}(y[t], w[t])}, \quad (5.16)$$

wobei $\mu_{R_k}(y, w)$ nun die Zugehörigkeitsfunktion zur k -ten Regel ist. Sowohl ein *Fuzzy*-System nach E. Mamdani als auch nach M. Sugeno kann stetige Funktionen beliebig genau approximieren [Jou92]. Da die Auswertung von (5.16) aber wesentlich einfacher ist als von (5.15), werden die *Fuzzy*-Regler in Kapitel 6 und 7 nach (5.16) realisiert.

Der Vergleich von (2.62) und (5.16) zeigt die enge Beziehung zwischen dem GRBF-Netzwerk und einem *Fuzzy*-System. Wie die Aktivierungsfunktion (2.59) beim GRBF-Netzwerk kann der Ausdruck $\mu_{R_k}(y[t], w[t]) / \sum_{k'} \mu_{R_{k'}}(y[t], w[t])$ als Partitionierungsfunktion eines Merkmalsraums verstanden werden, der von $y[t]$ und $w[t]$ aufgespannt wird. Auf Grund der vielfältigen Gemeinsamkeiten von *Fuzzy*-Systemen und künstlichen neuronalen Netzwerken ist in den letzten Jahren eine Reihe verschiedener *Neuro-Fuzzy*-Systemen entwickelt worden (z.B. [Kos92, Hay92, NK95]).

5.4. Datenbasierte Regelung mit neuronalen Netzen

Datenbasierte Regelungsverfahren werden vorzugsweise auf Regelkreise angewendet, bei denen entweder keine analytische Modellbildung möglich ist oder bei denen eine Formalisierung von Erfahrungen zu kompliziert ist [MSW92]. In diesen Fällen verbleibt nur noch die Möglichkeit, durch die Analyse von Meßdaten, die für den Regelkreis archiviert wurden, eine Regelung zu konstruieren. Eine solche datenbasierte Regelung zielt darauf ab, das Eingabe-Ausgabe-Verhalten der Strecke oder die Vorgehensweise bei manueller Regelung durch analytische Funktionen zu approximieren. Wie in Abschnitt 2.3.1 dargestellt wurde, stellt sich hier das Problem der Wahl einer geeigneten Funktionenklasse, welches durch das Fehlen analytischer Modellvorstellungen über das Eingabe-Ausgabe-Verhalten der Strecke oder über die Regelungsstrategie des Streckenbetreibers verschärft wird. Da neuronale Netze, wie das GRBF-Netzwerk, universelle Funktionenapproximatoren sind, die für hinreichend glatte Zielfunktionen sowohl gute Approximations- als auch Generalisierungsfähigkeiten aufweisen und deren Parameterzahl dennoch in Grenzen gehalten werden kann, scheinen sie für die genannte Aufgabenstellung geradezu prädestiniert zu sein. Diese Hypothese war, wie ich mehrfach betont habe, auch der Grund für meine eigene Anstrengung, ein Regelungskonzept zu entwickeln, das Architekturprinzipien von GRBF-Netzwerken übernimmt.

Bei der Entwicklung einer neuronalen Regelung kann zwischen drei grundsätzlich verschiedenen Ansätzen gewählt werden, die unabhängig von der Wahl des neuronalen Netzes sind [Bar90]. Bei der Konstruktion eines speziellen Reglers wird häufig auf Erweiterungen oder Kombinationen dieser drei Ansätze zurückgegriffen (siehe [MSW92] für eine Reihe von Anwendungen). Bei den drei Möglichkeiten handelt es sich um (i) das Kopieren einer vorhandenen Regelung, (ii) die Systemidentifikation und (iii) die inverse Systemidentifikation. Im folgenden werden diese drei Ansätze für den Fall eines SISO-Regelkreises vorgestellt und diskutiert, ohne daß auf die konkrete Realisierung einer solchen Regelung eingegangen wird. Es wird hier wieder davon ausgegangen, daß die für eine Regelung not-

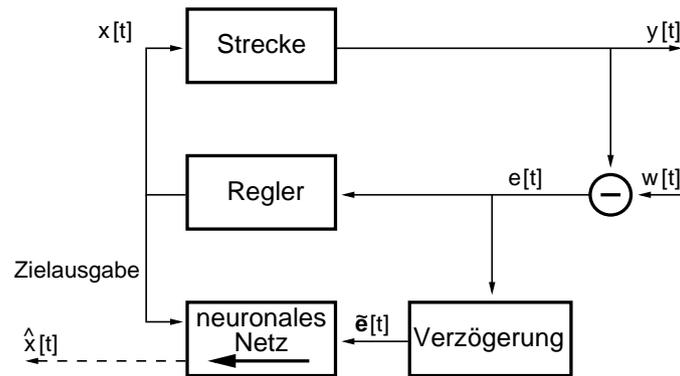


Abbildung 5.7.: **Kopieren einer Regelung.** Die Skizze zeigt den Informationsfluß für das Vorgehen bei der Kopie einer schon vorhandenen Regelung. Hierzu wird das neuronale Netzwerk parallel zur Regelung geschaltet. Die Arbeitsrichtung des Netzwerks wird durch den dicken Pfeil innerhalb der Box angedeutet. Während des Trainings werden dem Netzwerk die Eingabe- und Zielausgabedaten (dünne Pfeile) zugeführt.

wendigen Führungs-, Regel- und Stellgrößen in zeitlich diskretisierter Form $w[t]$, $y[t]$ und $x[t]$ vorliegen.

Kopieren einer Regelung

Dieses Verfahren nimmt eine Zwitterstellung zwischen erfahrungs- und datenbasierter Regelung ein, da vorausgesetzt wird, daß schon Erfahrungen bei der Regelung der Strecke vorliegen und entsprechend ein einigermaßen zuverlässig arbeitender Regler existiert. Dieser Regler – z.B. ein PID-Regler oder erfahrenes Bedienpersonal – soll durch das Training eines neuronalen Netzwerks kopiert werden. Erfahrungen, die in Bezug auf die Regelung einer Strecke gesammelt wurden, sollen also nicht durch einen Regelsatz, wie z.B. bei der *Fuzzy*-Regelung, formalisiert, sondern durch das Training mit den archivierten Daten auf ein neuronales Netz abgebildet werden.

Wie im Schaltbild aus Abbildung 5.7 gezeigt ist, bilden beim Kopieren eines Reglers durch ein neuronales Netz die Führungsgrößenabweichungen $e[t]$ die Eingabedaten für dieses Netz und die Stellgrößen $x[t]$ des Reglers die von ihm zu approximierenden Zielausgabedaten. Während des Trainings werden der vorhandene Regler und das neuronale Netz parallel geschaltet. Wenn das Training abgeschlossen ist, wird der ursprüngliche Regler aus dem Regelkreis entfernt und die Strecke mit der Stellgrößenabgabe $\hat{x}[t]$ des neuronalen Netzes geregelt. Um auch die dynamischen Eigenschaften der Regelung kopieren zu können, ist es sinnvoll, nicht nur die aktuellen Führungsgrößenabweichungen $e[t]$ für das Training zu verwenden, sondern Eingabevektoren $\tilde{e}[t] = (e[t], e[t - 1] \dots e[t - T_e])$ zu generieren, die die $T_e - 1$ vergangenen Führungsgrößenabweichungen umfassen [WS85]. Dies kann in einer sog. *Verzögerungsstrecke* erfolgen. Für das Kopieren wird also ein Ein-

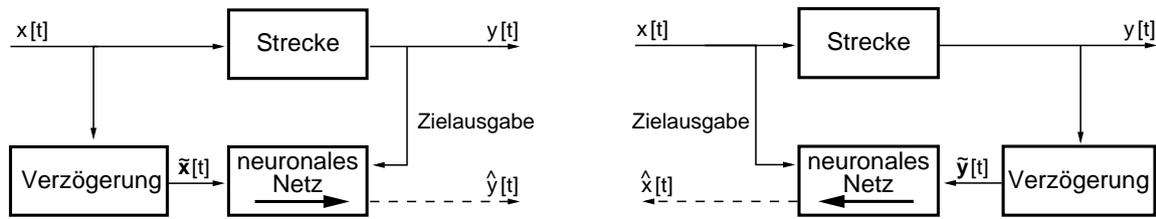


Abbildung 5.8.: **Systemidentifikation und inverse Systemidentifikation.** Bei der Systemidentifikation (links) sind Ein- und Ausgang der Strecke und des neuronalen Netzes parallel geschaltet. Bei der inversen Systemidentifikation (rechts) werden Strecke und Netzwerk antiparallel verbunden.

gabedatensatz $\mathcal{X} = \{\tilde{e}[t] \mid t = 1 \dots N\}$ und ein Ausgabedatensatz $\mathcal{Y} = \{x[t] \mid t = 1 \dots N\}$ benötigt.

Das Kopieren einer schon vorhandenen Regelung kann aus zwei Gründen angemessen sein [Bar90]:

- Handelt es sich um eine manuelle Regelung, dann kann durch das Kopieren die Reaktionsgeschwindigkeit und die Zuverlässigkeit der Regelung deutlich erhöht werden.
- Wird für die Kopie ein neuronales Netzwerk herangezogen, dessen interne Repräsentation der Funktionenapproximation gut interpretierbar ist (wie z.B. beim GRBF-Netzwerk), dann können die vom Personal verwendeten Kontrollregeln durch die Analyse der Netzwerkparameter offengelegt werden. Durch eine Justierung dieser Parameter kann so eine Verbesserung der Regelung erreicht werden [WS64].

Systemidentifikation

Im Gegensatz zum obigen Vorgehen beruht die Systemidentifikation ausschließlich auf Meßdaten, die das Eingabe-Ausgabe-Verhalten der Strecke beschreiben [Bar90]. Abbildung 5.8 zeigt den Informationsfluß für dieses Verfahren, bei dem die Ein- und Ausgabedaten der Strecke die Ein- und Ausgabedaten des neuronalen Netzes bilden. Um die dynamischen Eigenschaften der Strecke zu approximieren, kann wiederum durch eine Verzögerungsstrecke ein neuer Eingabevektor $\tilde{x}[t] = (x[t], x[t-1] \dots x[t-T_x])$ gebildet werden. Es gilt also $\mathcal{X} = \{\tilde{x}[t] \mid t = 1 \dots N\}$ und $\mathcal{Y} = \{y[t] \mid t = 1 \dots N\}$. Man erhält so durch das Training des neuronalen Netzes (2.50) mit den Trainingsdaten $\mathcal{Z} = \{\mathcal{X}, \mathcal{Y}\}$ ein rein datenbasiertes Modell der Strecke, welches die Abbildung $\hat{y}[t] = \hat{f}(\tilde{x}[t]; \Theta)$ ausführt.

Wie kann mit Hilfe einer solchen Systemapproximation $\hat{f}(\tilde{x}[t]; \Theta)$ ein Regler konstruiert werden? Eine erste Möglichkeit bietet sich an, wenn aus der für die Systemapproximation verwendeten Funktionenschar $\hat{f}(x; \Theta)$ direkt ein Regler entwickelt werden kann. Dieses Vorgehen setzt nicht zwingend die Anwendung von neuronalen Netzen voraus und wird

z.B. auch bei der Konstruktion *adaptiver Regler* verfolgt [GS84]. Wenn die Funktionenschar $\hat{f}(\mathbf{x}; \Theta)$ ein analytisches Modell der Strecke ist, dann zeigen solche adaptiven Regler eine große Ähnlichkeit mit den modellbasierten Reglern. Der wesentliche Unterschied liegt dann nur noch darin, daß die Parameter eines modellbasierten Reglers (z.B. PID-Regler) *vor* und die Parameter eines adaptiven Reglers *während* der Regelung angepaßt werden [Föl92].

Eine weitere Möglichkeit zur Reglerkonstruktion ergibt sich, wenn die Systemapproximation $\hat{f}(\tilde{\mathbf{x}}[t]; \Theta)$ als prädiktives Modell der Strecke aufgefaßt wird. Wenn zu einem Zeitpunkt t die Führungsgröße $w[t]$ durch eine – zunächst nicht bekannte - Stellgröße $x[t]$ eingestellt werden soll, dann muß gerade $w[t] = \hat{y}[t]$ und damit

$$w[t] = \hat{f}((x[t], x[t-1] \dots x[t-T_x]); \Theta) \quad (5.17)$$

gelten. Da alle vergangenen Stellgrößen $x[t-1] \dots x[t-T_x]$ bekannt sind, kann (5.17) nach der Stellgröße $x[t]$ aufgelöst werden. Diese Stellgröße führt zu einer Netzausgabe $\hat{y}[t]$, die im Rahmen der Qualität der Systemapproximation mit der gewünschten Ausgabe $y[t] = w[t]$ der Strecke übereinstimmen wird [MSW92].

Inverse Systemidentifikation

Die in Abbildung 5.8, rechts, skizzierte inverse Systemidentifikation stellt einen direkteren Zugang zum Reglerentwurf dar [Bar90]. Hier wird die verzögerte Regelgröße $\tilde{y}[t] = (y[t], y[t-1] \dots y[t-T_y])$ als Eingabe und die Stellgröße $x[t]$ als Zielausgabe eines neuronalen Netzes gewählt. Der Trainingsdatensatz setzt sich also aus $\mathcal{X} = \{\tilde{y}[t] \mid t = 1 \dots N\}$ und $\mathcal{Y} = \{x[t] \mid t = 1 \dots N\}$ zusammen. Wird ein Netzwerk mit einem solchen Datensatz trainiert, dann erhält man mit $\hat{x}[t] = f(\tilde{y}[t]; \Theta)$ ein inverses Modell der Strecke. Dieses inverse Streckenmodell kann direkt zur Regelung der Strecke verwendet werden, indem an die Eingabe des Netzwerks die Führungsgröße $\tilde{w}[t] = (w[t], y[t-1] \dots y[t-T_y])$ angelegt wird. Die Netzwerkausgabe $\hat{x}[t]$ ist dann gerade diejenige Stellgröße, die an die Strecke weitergegeben wird.

Das wesentliche Problem bei der Approximation eines inversen Modells tritt auf, wenn verschiedene Stellgrößen $x[t]$ eine ähnliche Streckenausgabe $y[t]$ zur Folge haben, d.h. wenn die Strecke nicht eine eineindeutige Abbildung darstellt [Bar90]. Die meisten neuronalen Netze neigen in diesem Fall dazu, über die verschiedenen Zielausgaben $x[t]$ zu mitteln. Trotz dieser Schwierigkeit ist die inverse Systemidentifikation auf Grund der einfachen Möglichkeit des Reglerentwurfs für eine Reihe von Kontrollaufgaben dokumentiert, wie z.B. bei der Kontrolle eines Roboterarms [WMS91] (siehe auch [MSW92] für weitere Beispiele).

Die gerade vorgestellten Methoden zum Entwurf einer datenbasierten Regelung sind zunächst nur als prinzipielle Lösungsvorschläge zu verstehen. Bei der konkreten Realisierung

einer Kontrollaufgabe wird es aber zumeist notwendig sein, eine Kombination bzw. Modifikation dieser drei Verfahren anzuwenden. In Kapitel 6 wird als Beispiel für eine datenbasierte Regelung die Kontrolle eines einfachen biochemischen Prozesses diskutiert. Dazu wird eine Systemidentifikation der Strecke mit einem modifizierten GRBF-Netzwerk durchgeführt. Um gute Reglereigenschaften zu erhalten, wird es dort notwendig sein, nicht nur die Stellgröße $\tilde{x}[t]$ sondern auch die Führungsgröße $y[t]$ als Eingabe für das neuronale Netz zu verwenden.

6. Regelung eines Bioreaktors im Computereperiment

Die beiden letzten Kapitel haben im wesentlichen zu zwei Ergebnissen geführt: (i) es wurde ein sicher konvergierender Algorithmus zur Dichteschätzung durch eine Mischung multivariater Normalverteilungen bereitgestellt und darauf aufbauend der LLM-Approximator abgeleitet; daraus wurde (ii) eine Konstruktionsvorschrift für ein System lokaler Experten entwickelt, welches in enger Beziehung zum GRBF-Netzwerk steht. Das vorliegende und das folgende Kapitel haben zum Ziel, diese beiden Ergebnisse auf jeweils ein Regelungsproblem anzuwenden. Dabei werden wir uns zunächst dem Problem widmen, ein nur als mathematisches Modell existierendes System – einen *Bioreaktor* – durch eine inverse Systemidentifikation mit einem LLM-Approximator zu regeln. Die Dynamik eines solchen Bioreaktors ist verschiedenen nichtlinearen Einflüssen unterworfen [ALLR82]. Entsprechend schwierig ist die Regelung eines solchen komplexen, dynamischen Systems [PJ93, KLP⁺94, MSW92].

Das folgende Kapitel ist in drei Teile gegliedert. Zunächst wird die Systembeschreibung vorgestellt und die Dynamik des Bioreaktors untersucht. Anschließend werden mit einem PID-, *Fuzzy*-, und Neuro-Regler drei verschiedene Möglichkeiten zur Regelung beschrieben und diskutiert. Den Abschluß bildet die Darstellung der quantitativen Ergebnisse, die mit den untersuchten Reglern erzielt wurden.

6.1. Dynamik des Bioreaktors

Der Zweck eines Bioreaktors ist die Herstellung biologischer Produkte (z.B. von Hefe oder von Insulin) unter Ausnutzung der Reproduktionsfähigkeit und des Stoffwechsels geeigneter Zelltypen. Im einfachsten Fall ist ein Bioreaktor aus einem Tank aufgebaut, der Wasser und Zellen enthält. Abbildung 6.1 zeigt eine Skizze eines solchen Reaktors zusammen mit einem Regler, der Einfluß auf die Ventilstellung $v(t)$ nimmt. Die Zellen im Reaktortank verbrauchen Nährstoffe, erzeugen durch ihren Stoffwechsel erwünschte wie unerwünschte Produkte und reproduzieren sich, falls die Umgebungsbedingungen (z.B. Temperatur, Konzentrationen von Nähr- und Abfallstoffen) geeignet eingestellt sind. Der Tank verfügt über

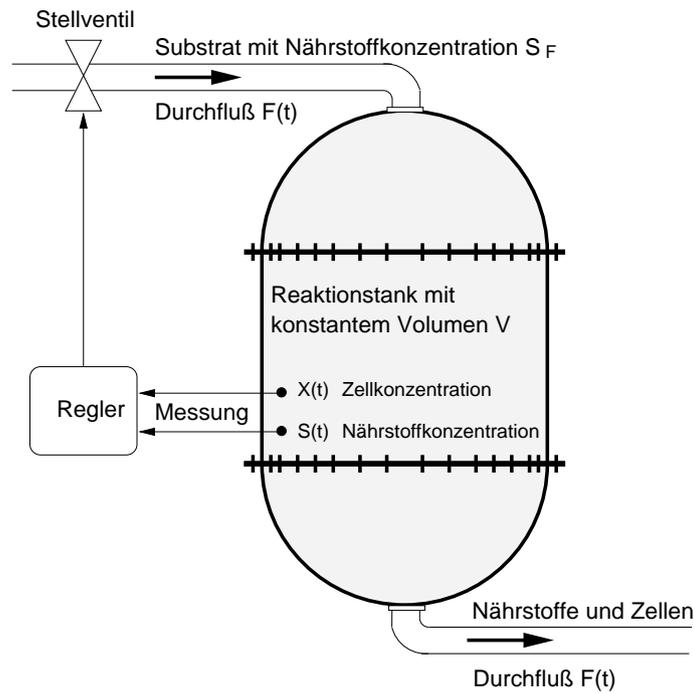


Abbildung 6.1.: Schematische Darstellung eines Bioreaktors.

einen Zu- und einen Abfluß, so daß eine nährstoffhaltige Flüssigkeit bei konstanter Tankfüllung eingebracht werden kann. Bioreaktoren können in ihrem dynamischen Verhalten sehr komplex werden, da die Zellen selbstregelnde Systeme sind, deren Wachstums- und Produktionsrate stark von der Umgebung abhängt. Die einfachste mathematische Beschreibung eines Bioreaktors ist der *continuous flow stirred tank reactor* (CFSTR) [MSW92]. In diesem Modell hängt die Zellkonzentration $X(t)$ nur von der Nährstoffkonzentration $S(t)$ ab, die sich bei einem Durchfluß $F(t)$ durch einen Behälter des Volumens V einstellt. Die Konzentration S_F der hinzugefügten Nährstoffe sei konstant. Damit läßt sich das Differentialgleichungssystem

$$\frac{d}{dt}X(t) = -F(t)\frac{X(t)}{V} + \tilde{\mu}(S(t))X(t) \quad (6.1)$$

$$\frac{d}{dt}S(t) = F(t)\frac{S_F - S(t)}{V} - \tilde{\sigma}(S(t))X(t) \quad (6.2)$$

aufstellen. Die zeitliche Änderung der Zellkonzentration $X(t)$ hängt linear vom Durchfluß $F(t)$ und über die *Wachstumsrate* $\tilde{\mu}(S(t))$ nichtlinear von der aktuellen Zellkonzentration $X(t)$ ab. Analog nimmt die Nährstoffkonzentration $S(t)$ zu, wenn mehr Nährstoffe in den Tank eingebracht werden als in diesem gerade vorhanden sind. Die Nährstoffkonzentration $S(t)$ hängt wegen der *Nährstoffverbrauchsrate* $\tilde{\sigma}(S(t))$ wieder in nichtlinearer Weise von der Zellkonzentration $X(t)$ ab. Das System ist also linear im Durchfluß $F(t)$ und wegen der Wachstumsrate $\tilde{\mu}(S(t))$ und der Nährstoffverbrauchsrate $\tilde{\sigma}(S(t))$ im allgemeinen nichtlinear in der Nährstoffkonzentration $S(t)$.

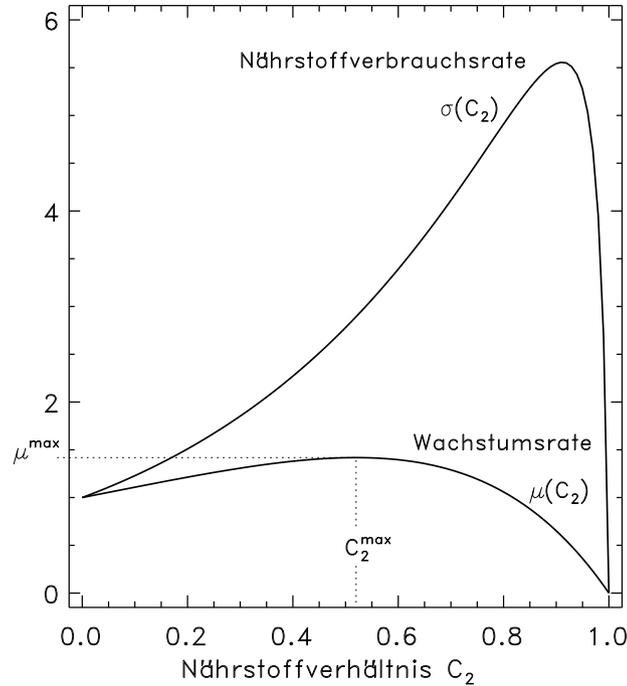


Abbildung 6.2.: Wachstumsrate $\mu(C_2)$ und Nährstoffverbrauchsrate $\sigma(C_2)$ als Funktion des Nährstoffverhältnisses C_2 . Das Maximum μ^{\max} der Wachstumsrate bei C_2^{\max} hat einen entscheidenden Einfluß auf die Existenz von Fixpunkten (vgl. auch Tabelle 6.1).

Mit den Definitionen [ALLR82]

$$C_1(t) \equiv \frac{X(t)}{S_F}, \quad C_2(t) \equiv \frac{S_F - S(t)}{S_F}, \quad u(t) \equiv \frac{F(t)}{V} \quad (6.3)$$

und

$$\mu(C_2(t)) \equiv \tilde{\mu}(S_F(1 - C_2(t))), \quad \sigma(C_2(t)) \equiv \tilde{\sigma}(S_F(1 - C_2(t))) \quad (6.4)$$

können diese Gleichungen in eine dimensionslose und kompaktere Form gebracht werden. $C_2(t)$ ist ein dimensionsloses *Nährstoffverhältnis*, welches immer im Intervall $[0, 1]$ liegt. Ein großes $C_2(t)$ bedeutet eine niedrige Nährstoffkonzentration und vice versa. Im folgenden wird auch die Zellkonzentration $C_1(t)$ auf das Intervall $[0, 1]$ und der Durchfluß $u(t)$ auf $[0, 2]$ eingeschränkt [ALLR82]. Mit den Definitionen (6.3) und (6.4) kann das Differentialgleichungssystem (6.1) und (6.2) zu

$$\frac{d}{dt}C_1(t) = -C_1(t)u(t) + \mu(C_2(t))C_1(t) \equiv f_1(\mathbf{C}(t)) \quad (6.5)$$

$$\frac{d}{dt}C_2(t) = -C_2(t)u(t) + \sigma(C_2(t))C_1(t) \equiv f_2(\mathbf{C}(t)) \quad (6.6)$$

Fixpunkt $C^{S,0}$ <i>washout</i>	$0 < u < 1$	Sattelpunkt
	$1 \leq u \leq \mu^{\max}$	stabiler Knoten
Fixpunkt $C^{S,1}$	$0 < u \lesssim 0.2402$	stabiler Knoten
	$0.2402 \lesssim u \lesssim 0.8288$	stabiler Focus
	$0.8288 \lesssim u \leq \mu^{\max}$	instabiler Focus
Fixpunkt $C^{S,2}$	$1 \leq u \leq \mu^{\max}$	Sattelpunkt

Tabelle 6.1.: Eigenschaften der Fixpunkte des Bioreaktors.

umgeformt werden. Der Punkt $\mathbf{C} \equiv (C_1, C_2)$ beschreibt zusammen mit dem Durchfluß $u(t)$ vollständig den Zustand des Bioreaktors.

Ein einfaches Modell für die Wachstumsrate $\mu(C_2)$ und die Nährstoffverbrauchsrate $\sigma(C_2)$ ist das sog. *substrate inhibition model* [ALLR82] mit

$$\mu(C_2) \equiv (1 - C_2)e^{C_2/\gamma} \quad (6.7)$$

und

$$\sigma(C_2) \equiv \mu(C_2) \frac{1 + \beta}{1 + \beta - C_2}. \quad (6.8)$$

Die beiden Parameter γ und β werden entsprechend dem Vorgehen von [MSW92] für den Rest dieses Kapitels auf $\gamma = 0.48$ und $\beta = 0.02$ festgehalten. Abbildung 6.2 zeigt den Verlauf der beiden Funktionen (6.7) und (6.8). Bemerkenswert und für das dynamische Verhalten des Systems von großer Bedeutung ist die Existenz eines Maximums der Wachstumsrate $\mu(C_2)$. Eine analytische Diskussion der Funktion $\mu(C_2)$ zeigt, daß das Maximum bei $C_2^{\max} = 1 - \gamma = 0.52$ liegt und daß

$$\mu^{\max} \equiv \mu(C_2^{\max}) = \gamma e^{\frac{1-\gamma}{\gamma}} \simeq 1.418 \quad (6.9)$$

gilt.

Die Dynamik des durch (6.5)–(6.8) definierten Modells eines Bioreaktors wird in Anhang A.2 detailliert untersucht. Im Besonderen werden dort die Fixpunkte und ihre Stabilitätseigenschaften berechnet. Tabelle 6.1 und Abbildung 6.3 fassen die Ergebnisse dieser Diskussion zusammen. Der Bioreaktor hat für $u < \mu^{\max}$ drei Fixpunkte $C^{S,0}$, $C^{S,1}$ und $C^{S,2}$, wobei lediglich der Fixpunkt $C^{S,1}$ im Bereich $0 < u \lesssim 0.8288$ stabil ist. Der Fixpunkt $C^{S,0}$ kann zwar auch stabil sein, beschreibt aber die wenig wünschenswerte Situation, daß bei maximaler Nährstoffkonzentration alle Zellen aus dem Reaktortank gespült werden. Dieser Zustand wird auch als *washout* bezeichnet.

Konsequenzen für die Regelung des Bioreaktors

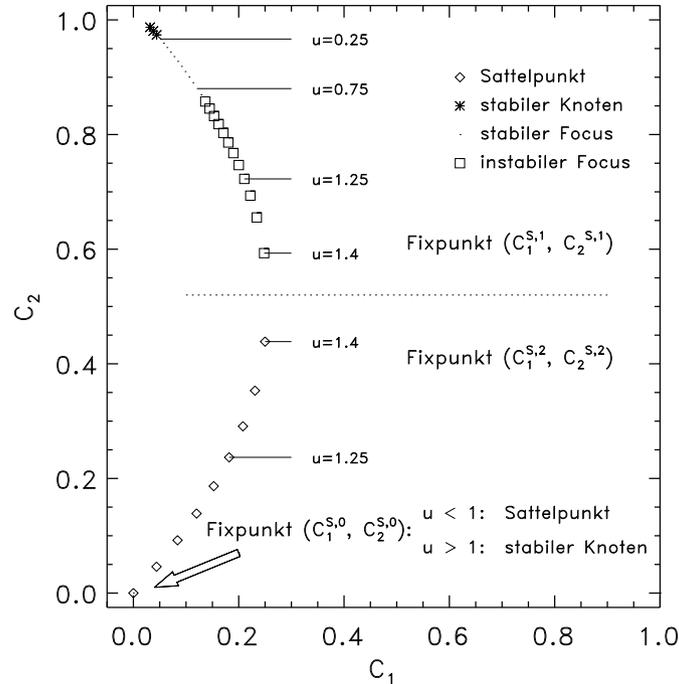


Abbildung 6.3.: **Lage der Fixpunkte im Phasenraum in Abhängigkeit vom Durchfluß u .** Die Lage der Fixpunkte ist durch die Lösungen der Gleichungen (A.32) bzw. (A.33) und (A.34) gegeben. Die Stabilität eines Fixpunkts, die durch den Symboltyp gekennzeichnet ist, wird aus der Analyse der Jacobimatrix (A.40) bestimmt.

Soll in der industriellen Praxis mit dem oben skizzierten Bioreaktor ein bestimmtes Produkt hergestellt werden, so ist es wünschenswert den Reaktor so zu regeln, daß eine möglichst hohe Ausbeute an Zellen erreicht wird. Die Ausbeute an Zellmaterial wird von der Konzentration $C_1(t)$ und vom Durchfluß $u(t)$ durch den Reaktor bestimmt. Daher wird in der Diskussion der verschiedenen Regelungen die Ausbeute $A(t) = C_1(t)u(t)$ bzw. die mittlere Ausbeute

$$\bar{A}(T) = \frac{1}{T} \int_0^T C_1(t)u(t) dt \quad (6.10)$$

eine wesentliche Rolle spielen. Die Frage ist nun, mit welchem Phasenraumpunkt eine maximale Ausbeute erreicht werden kann, d.h. welches die Zielgrößen für die Regelung des Bioreaktors sein sollen. Aus Abbildung 6.3 läßt sich schließen, daß an den beiden Fixpunkten $C^{S,1}$ und $C^{S,2}$ für $u = \mu^{\max}$ die gleiche maximale Ausbeute A^{\max} erreicht wird. Da der Fixpunkt $C^{S,2}$ immer ein Sattelpunkt ist und daher für die Untersuchung der Leistungsfähigkeit eines Reglers wenig interessant ist, wird die Diskussion der Ausbeute auf den Fixpunkt $C^{S,1}$ beschränkt, der sowohl stabil als auch instabil sein kann. Den Zusammenhang zwischen der Zellkonzentration $C_1^{S,1}$ und der dort erreichten Ausbeute zeigt

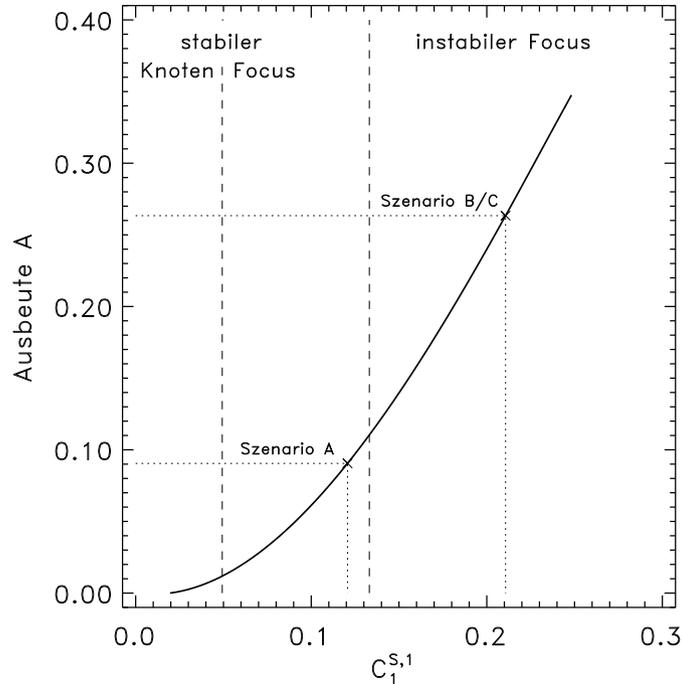


Abbildung 6.4.: **Ausbeute in Abhängigkeit von der Zellkonzentration $C_1^{S,1}$.** Mit den gestrichelten Linien sind Gebiete verschiedener Fixpunkte getrennt. Die punktierten Linien markieren die Position der Zielgrößen für Szenario A bzw. B/C.

Abbildung 6.4. Man erkennt, daß die Ausbeute nichtlinear vom Wert der Zellkonzentration $C_1^{S,1}$ am Fixpunkt $C^{S,1}$ abhängt. Ebenfalls eingetragen sind die Bereiche, in denen die entsprechende Ausbeute im Rahmen einer stabilen bzw. instabilen Dynamik erreicht werden kann. Offensichtlich wird eine Erhöhung der Ausbeute durch eine zunehmende Destabilisierung der Dynamik erkaufte. Demzufolge ist zu erwarten, daß eine Regelung, die auf die Einhaltung eines großen Sollwerts C_1^* und damit einer hohen Ausbeute abzielt, zunehmend schwieriger wird.

Für die Untersuchung der Regelungen werden im folgenden drei verschiedene Szenarien genauer betrachtet:

- Im **Szenario A** soll der Bioreaktor auf eine Zellkonzentration von $C_1^{A,*} = 0.1207$ geregelt werden. Wie aus Abbildung 6.3 abgelesen werden kann, führt dies zu einem stabilen Focus. Der Wert $C_1^{A,*}$ wird nach (A.33) und (A.34) für einen Durchfluß $u^{A,*} = 0.75$ und ein Nährstoffverhältnis $C_2^{A,*} = 0.8801$ erreicht. Wenn die Zielgröße $C_1^{A,*}$ durch einen Regler eingehalten wird, sollte die mittlere Ausbeute bei $\bar{A}^{A,*} = C_1^{A,*} u^{A,*} = 0.0905$ liegen.
- Das **Szenario B** ermöglicht eine höhere Ausbeute, indem die Zielgröße auf $C_1^{B,*} =$

0.2107 erhöht wird. Dieser Punkt ist mit $u^{B,*} = 1.25$ und $C_2^{B,*} = 0.7226$ ein instabiler Focus und wird voraussichtlich durch eine Regelung schwieriger zu halten sein. Die mittlere Ausbeute wird so aber auf $\bar{A}^{B,*} = 0.2634$ angehoben. In den Szenarien A und B wird angenommen, daß das Durchflußventil, durch welches ein Regler Einfluß auf die Dynamik des Bioreaktors nehmen kann, eine lineare Kennlinie $h_1(v)$ hat, d.h. es gilt

$$u = h_1(v) = v, \quad (6.11)$$

wobei die Ventilstellung v im Intervall $[0, 2]$ liegt.

- **Szenario C** hat dieselben Zielgrößen wie Szenario B. Allerdings wird nun dem Bioreaktor eine schnelle Systemveränderung hinzugefügt, indem das Durchflußventil zufällig und schlagartig zwischen einer linearen und einer quadratischen Kennlinie wechseln kann. Das Durchflußventil kann daher im Szenario C neben (6.11) auch einer Kennlinie

$$u = h_2(v) = \frac{v^2}{2} \quad (6.12)$$

folgen. Die beiden Zustände $s = 1$ und $s = 2$ sollen gleich wahrscheinlich sein und eine mittlere Lebensdauer τ haben. Durch diese Modifikation erhält der Bioreaktor ein multifunktionales Verhalten, welches in Abschnitt 4.3 näher dargestellt wurde. Es ist naheliegend, daß dadurch die Regelung des Bioreaktors nochmals erschwert wird.

Die zunächst artifiziell erscheinende Erweiterung des Bioreaktors im Szenario C spiegelt das Verhalten realer Ventile wider, die zur Regelung der Anlage für die Rückstandsverbrennung der Wacker-Chemie GmbH benötigt werden. In Abschnitt 7.2.3 wird noch näher auf diese Problematik eingegangen.

6.2. Regelung des Bioreaktors

6.2.1. Simulation des Bioreaktors

Im folgenden soll die Regelung des Bioreaktors mit verschiedenen Methoden untersucht werden. Dazu müssen für die Simulation des Bioreaktors die Gleichungen (6.5) und (6.6) integriert werden. Allerdings wird ein chemischer Prozeß, der im industriellen Maßstab eingesetzt wird, in der Realität nie exakt irgendwelchen analytischen Beschreibungen gehorchen. Vielmehr wird durch wechselnde Umgebungsbedingungen, variierende Grundstoffe und verschiedene Verschmutzungen ein *ad hoc* nicht bekanntes Rauschen in das System eingebracht. Hinzu kommen Sensoren, die zu Meßrauschen führen, und Aktoren,

die vorgegebene Stellaktionen nicht exakt ausführen. Eine weitere Erschwernis der Regelung stellen Meßgeräte dar, die nicht kontinuierlich ein Meßsignal liefern, sondern eine gewisse Totzeit benötigen, bis eine neue Messung vorgenommen werden kann.

Diese analytisch nur schwer beschreibbaren Effekte werden für die Simulation des Bioreaktors durch drei künstliche Störungsquellen beschrieben: (i) Nicht optimale Bedingungen zu Beginn des Produktionsprozesses werden durch ein starkes Verrauschen der Startwerte erreicht. (ii) Die nur endliche Genauigkeit einer Messung wird durch ein leichtes Verrauschen der Meßwerte während der Integration von (6.5) und (6.6) simuliert. (iii) Die Totzeit der Meßgeräte wird dadurch berücksichtigt, daß die Regelung nur alle T_M Zeitschritte in das System eingreifen kann.

Für die Integration des Differentialgleichungssystems (6.5) und (6.6) muß eine zeitliche Diskretisierung vorgenommen werden (siehe auch Abschnitt 5.2). Die Integrations-schrittweite Δ wird entsprechend der Literatur [MSW92] auf 0.01 festgelegt. Um die Qualität des Reglers beurteilen zu können, werden T_B Simulationsläufe mit jeweils T_L Regelungsschritten berechnet. Für das gesamte Simulationsszenario ergibt sich der in Abbildung 6.5 dokumentierte Ablauf.

Die Größen R_i und R_m sind Zufallsvariablen, die gleichverteilt aus einem endlichen Intervall gewürfelt werden, und im Mittel Null sind. Die Streuung von R_i beträgt $\sigma_i = 0.1$ und die Streuung des Meßrauschens R_m ist $\sigma_m = 0.001$. Für die Beurteilung einer Regelung werden immer $T_B = 100$ Simulationsläufe ausgeführt. Jeder Simulationslauf setzt sich aus $T_L = 500$ Regelungsschritten zusammen, zwischen denen das System jeweils T_M Zeitschritte simuliert wird. $T_M = 1$ bedeutet, daß die Messung ohne Totzeit arbeitet, d.h. die Regelung sofort auf Abweichungen reagieren kann. Mit $T_M > 1$ wird eine Messung mit Totzeit realisiert, was sicherlich die Regelung des Bioreaktors erschweren wird.

Ein Datenpunkt $(\tilde{C}_1[t], \tilde{C}_2[t], u[t]) \in \mathbb{R}^3$ beschreibt den Zustand der Simulation zum diskreten Zeitpunkt t . Wegen dieser Diskretisierung umfaßt t die Zeitdauer von der Bereitstellung der Messungen $\tilde{C}_1[t]$ und $\tilde{C}_2[t]$ bis zum Erreichen eines Durchflusses $u[t]$ auf Grund der Ventilstellung $v[t]$, die vom Regler ausgegeben wurde. Es ist also zu beachten, daß $\tilde{C}_1[t]$ und $\tilde{C}_2[t]$ nicht von $u[t]$, sondern vom Durchfluß $u[t - T_M]$ abhängen, der T_M Simulations-schritt vorher herrschte.

6.2.2. Kriterien zur Beurteilung einer Regelung

Der Rest dieses Kapitels wird von der qualitativen Beschreibung dreier Regler für den Bioreaktor und der Darstellung der quantitativen Ergebnisse, die diese Regler bei der Kontrolle des Bioreaktors erreichen, eingenommen. Für die Konstruktion der Regler und die Diskussion der quantitativen Ergebnisse ist es zunächst notwendig, Kriterien einzuführen, mit deren Hilfe die verschiedenen Regler beurteilt werden können. Diese Beurteilung wird sich an drei Gesichtspunkten orientieren:

Schleife über T_B Simulationsläufe:

bestimme Anfangsbedingungen:

$$C_1[0] = C_1^* + R_i$$

$$C_2[0] = C_2^* + R_i$$

$$u[0] = u^* + R_i$$

Schleife über T_L Regelungsschritte:

Schleife über T_M Simulationsschritte:

$$t \leftarrow t + 1$$

$$C_1[t] = C_1[t - 1] + \Delta(-C_1[t - 1]u[t - 1] + C_1[t - 1]\mu(C_2[t - 1]))$$

$$C_2[t] = C_2[t - 1] + \Delta(-C_2[t - 1]u[t - 1] + C_1[t - 1]\sigma(C_2[t - 1]))$$

Messung:

$$\tilde{C}_1[t] = C_1[t] + R_m$$

$$\tilde{C}_2[t] = C_2[t] + R_m$$

Regelung:

$$v[t] = \text{Regler}(\tilde{C}_1[t], \tilde{C}_2[t], C_1^*, C_2^*)$$

Ventil:

$$u[t] = \begin{cases} \text{Szenario A und B} & : h_1(v[t]) \\ \text{Szenario C} & : h_{s[t]}(v[t]) \end{cases} \quad \text{wobei } s[t] \in \{1, 2\} \quad \text{mit } \tau = 350$$

Abbildung 6.5.: Vorgehen bei der Simulation des Bioreaktors.

- Das **washout** bezeichnet die Situation, daß die Zellen vollständig aus dem Reaktor gespült wurden. Beim sog. **steady state** schließt der Regler das Durchflußventil ($u = 0$), die Nährstoffkonzentration nimmt stetig ab ($C_2 \rightarrow 1$), was schließlich zu einer kleinen, aber konstanten Zellkonzentration führt. Sowohl **washout** als auch **steady state** sind äußerst unerwünschte Zustände, die der Regler möglichst vermeiden sollte. Tritt einer der beiden Situationen in einem Simulationslauf auf, so wird der komplette Lauf als Verlust aufgefaßt. Wir sprechen dann auch von einem **verlorenen Lauf** und bezeichnen die Zahl der verlorenen Läufe mit \bar{T}_B .
- Die oben schon eingeführte mittlere **Ausbeute** (6.10) ist ein Maß für die mittlere Produktionsrate an Zellen. Da diese Größe maßgeblich die Wirtschaftlichkeit eines Bioreaktors und damit die Auswahl einer Regelung bestimmt, wird sie in der Praxis ein großes Gewicht haben. Für die Berechnung der mittleren Ausbeute $\bar{A}[T]$, die während aller Simulationsläufe erreicht wird, wird über die Gesamtzeit $T = T_B T_L T_M$

des Versuchs mit

$$\bar{A}[T] = \frac{1}{T_B T_L T_M} \sum_{t=1 \dots (T_B - \bar{T}_B) T_L T_M} C_1[t] u[t] \quad (6.13)$$

gemittelt. Die Summe in (6.13) umfaßt nur diejenigen $T_B - \bar{T}_B$ Simulationsläufe, bei denen kein *washout* oder *steady state* aufgetreten ist. Verlorene Simulationsläufe gehen in (6.13) also mit einer Ausbeute von Null ein. Die mittlere Ausbeute $\bar{A}[T]$ kann zum Einen durch die Wahl des Szenarios B beeinflußt werden. Dabei muß aber in Kauf genommen werden, daß die Regelung auf Grund der Nichtlinearität der Strecke zunehmend schwieriger zu handhaben sein wird. Zum Anderen kann die Ausbeute durch eine Regelung erhöht werden, der es gelingt, mit einem kleinen Fehler zuverlässig den Sollwert einzuhalten. In diesem Fall wird die mittlere Ausbeute (6.13) die theoretisch erwartete Ausbeute \bar{A}^* erreichen.

- Der **Fehler** $\bar{e}[T]$ ist ein Maß für die Streuung der Meßwerte $\tilde{C}_1[t]$ um den Sollwert C_1^* und quantifiziert die Genauigkeit einer Regelung. Beiträge für den mittleren Fehler

$$\bar{e}[T] = \sqrt{\frac{1}{(T_B - \bar{T}_B) T_L} \sum_{t=1, T_M, 2T_M \dots (T_B - \bar{T}_B) T_L T_M} (\tilde{C}_1[t] - C_1^*)^2} \quad (6.14)$$

werden für alle $T_B - \bar{T}_B$ Simulationsläufe und für jeden Regelungsschritt genommen, d.h. nur zu jenen Zeitpunkten, zu denen tatsächlich auch Messungen vorliegen. Es ist zu erwarten, daß $\bar{e}[T]$ immer über der Streuung σ_m des Rauschens liegt, das bei der Messung auf den Wert $C_1[t]$ aufaddiert wird.

Die oben definierte mittlere Ausbeute $\bar{A}[T]$ und der Fehler $\bar{e}[T]$ sind zwei Größen, die deutlich mit der Zahl \bar{T}_B der verlorenen Läufe korreliert ist. Viele verlorene Läufe werden einerseits dazu führen, daß die mittlere Ausbeute abnimmt. Andererseits wird man in einem solchen Fall einen relativ kleinen Fehler erhalten, obwohl bei einem *washout* oder *steady state* die Zellkonzentration stark von der Sollkonzentration C_1^* abweicht, da verlorene Läufe nicht in die Berechnung des mittleren Fehlers eingehen.

6.2.3. Drei Regler für den Bioreaktor

Nachdem im letzten Abschnitt dargestellt wurde, wie der Bioreaktor im Detail simuliert wird und nach welchen Kriterien eine Regelung beurteilt werden kann, soll nun darauf eingegangen werden, welche Regler verwendet werden sollen und wie diese parametrisiert und an die Strecke angekoppelt werden. Entsprechend der Klassifikation, die in Kapitel 5 vorgenommen wurde, soll jeweils ein modellbasierter, ein erfahrungsbasierter und ein datenbasierter Regler untersucht werden. Bei den folgenden Ausführungen sollte im Gedächtnis behalten werden, daß diese Regler ohne Kenntnis der auf den letzten Seiten untersuchten Dynamik des Bioreaktors konstruiert werden sollen. Die obige Diskussion der

Dynamik und Stabilität des Bioreaktors diente nur einem besseren Verständnis der Strecke, nicht jedoch einer Verbesserung der Regelung.

PID-Regler

Der PID-Regler (siehe auch Abschnitt 5.2) ist ein klassischer Regelungsalgorithmus. Zur Konfiguration und Parametrisierung sind eine Vielzahl von Verfahren veröffentlicht worden, die auf der Kenntnis eines analytischen Modells der Strecke beruhen (z.B. [Föl92]). Da im folgenden davon ausgegangen wird, daß ein solches analytisches Modell nicht zur Verfügung steht, können diese Verfahren nicht verwendet werden. Dies entspricht auch der üblichen Situation in der Praxis, in der häufig die zu regelnde Strecke nur durch Erfahrungswerte beschrieben werden kann. In diesen Fällen wird häufig der Parametersatz des PID-Reglers durch eine grobe Abschätzung des Streckenverhaltens initialisiert und anschließend in einem zyklischen Prozeß der leichten Variation der Parameter optimiert.

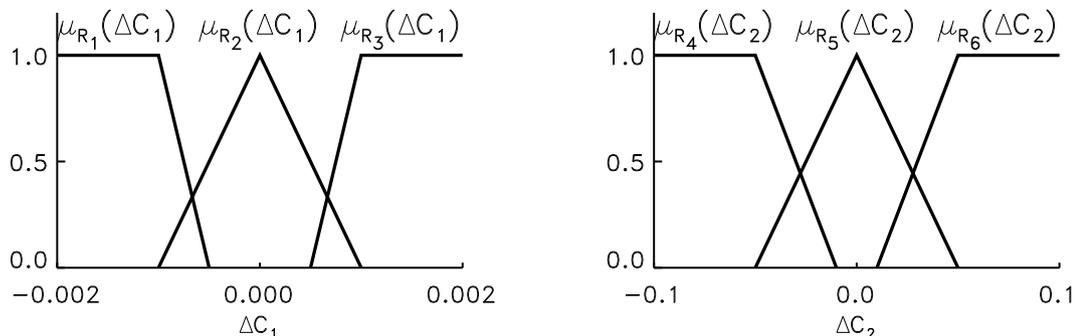
Dieses Vorgehen wird für die Optimierung der Regelung des Bioreaktors mit einem PID-Regler durch die Verwendung der *downhill simplex* Methode [PTVF92, NM65] imitiert. Dieses Verfahren dient der Minimierung einer Funktion $f : \mathbb{R}^N \rightarrow \mathbb{R}$. Dabei werden in jedem Iterationsschritt $N + 1$ Punkte $\mathbf{x} \in \mathbb{R}^N$ ausgewertet. Diese Punkte bilden – verbindet man jeden Punkt mit allen anderen – einen sog. Simplex. Der Simplex wird nun, abhängig von den Funktionsauswertungen $f(\mathbf{x}_i)$, $i = 1 \dots N + 1$, und unter Erhaltung seines Volumens, so deformiert, daß er in Richtung eines Minimums von $f(\mathbf{x})$ wandert. Der Vorteil dieser Methode liegt darin, daß keine partiellen Ableitungen $\partial f(\mathbf{x})/\partial x_j$ benötigt werden.

Die Stellgröße des PID-Regler für den Bioreaktor ist die Ventilstellung $v[t]$, die Regelgröße ist $\tilde{C}[t]$ und die Führungsgröße ist abhängig vom gewählten Szenario, $C_1^{A,*} = 0.75$ oder $C_1^{B,*} = 1.25$. Für die Optimierung des Reglers wird derjenige Punkt $\mathbf{x} = (K_p, T_N, T_V, T_D)$ im 4-dimensionalen Parameterraum gesucht, der zu einer minimalen Fehlerfunktion $f(\mathbf{x}) \equiv \bar{e}[T] + \lambda \bar{A}[T]$ führt. Ausgehend von einem Startpunkt \mathbf{x}_0 wird das Verfahren so lange iteriert, bis keine wesentliche Verringerung der Funktion $f(\mathbf{x})$ feststellbar ist. Üblicherweise sind dazu ca. 150 Auswertungen von $f(\mathbf{x})$ notwendig.

Fuzzy-Regler

Für eine erfahrungsbasierte Regelung des Bioreaktors soll der in Abschnitt 5.3 beschriebene *fuzzy* Sugeno-Regler verwendet werden, dessen Parameter mit der oben erwähnten *downhill simplex* Methode optimiert werden. Dazu wird mit den Definitionen

$$\Delta C_1^*[t] \equiv C_1^* - \tilde{C}_1[t] \quad \text{und} \quad \Delta C_2^*[t] \equiv C_2^* - \tilde{C}_2[t] \quad (6.15)$$


 Abbildung 6.6.: Die Zugehörigkeitsfunktionen μ_{R_k} , $k = 1 \dots 6$, zu den *fuzzy* Regeln.

ein kleiner Satz von sechs *fuzzy* Regeln

$$\begin{array}{ll}
 \text{wenn } (\Delta C_1^*[t] \in R_1) \text{ dann } \Delta v_1 = \Delta_a & \text{wenn } (\Delta C_2^*[t] \in R_4) \text{ dann } \Delta v_4 = \Delta_b \\
 \text{wenn } (\Delta C_1^*[t] \in R_2) \text{ dann } \Delta v_2 = 0 & \text{wenn } (\Delta C_2^*[t] \in R_5) \text{ dann } \Delta v_5 = 0 \\
 \text{wenn } (\Delta C_1^*[t] \in R_3) \text{ dann } \Delta v_3 = -\Delta_a & \text{wenn } (\Delta C_2^*[t] \in R_6) \text{ dann } \Delta v_6 = -\Delta_b
 \end{array} \quad (6.16)$$

aufgestellt, deren Zugehörigkeitsfunktionen μ_{R_k} , $k = 1 \dots 6$, in Abbildung 6.6 dargestellt sind. Die Regeln (6.16) beinhalten die einfache Aussage, daß bei einer Abweichung der Meßwerte $\tilde{C}_1[t]$ bzw. $\tilde{C}_2[t]$ von den Sollgrößen $C_1^*[t]$ bzw. $C_2^*[t]$ das Durchflußventil um einen kleinen Betrag geöffnet bzw. geschlossen wird.

Wenn gefordert wird, daß $\mu_{R_2}(\Delta C_1)$ und $\mu_{R_5}(\Delta C_2)$ jeweils symmetrisch um Null angeordnet sind und $\mu_{R_1}(\Delta C_1) = \mu_{R_3}(-\Delta C_1)$ bzw. $\mu_{R_4}(\Delta C_2) = \mu_{R_6}(-\Delta C_2)$ gilt, dann sind sechs Parameter für die Beschreibung der Zugehörigkeitsfunktionen notwendig. Da die Ausgaben der Regeln (6.16) nur die beiden weiteren Parameter Δ_a und Δ_b enthalten, umfaßt dieser *fuzzy* Regler acht Parameter.

Die Ausgabe des durch die Regeln (6.16) definierten Sugeno-Reglers ergibt sich nach (5.16) durch

$$v[t] = v[t - T_M] + \frac{\sum_{k=1}^6 \Delta v_k \mu_{R_k}(\Delta C_1^*[t], \Delta C_2^*[t])}{\sum_{k=1}^6 \mu_{R_k}(\Delta C_1^*[t], \Delta C_2^*[t])}. \quad (6.17)$$

Für die Optimierung eines solchen *fuzzy* Reglers wird mit der *downhill simplex* Methode ein Punkt $\mathbf{x} \in R^8$ im achtdimensionalen Parameterraum gesucht, der die Funktion $f(\mathbf{x}) \equiv \bar{e}[T] + \lambda \bar{A}[T]$ minimiert. Dieses Minimum wird üblicherweise nach ca. 250 Auswertungen von $f(\mathbf{x})$ gefunden.

Regelung mit einem LLM-Approximator

Für die Konstruktion eines datenbasierten Reglers soll der in Abschnitt 3.3 entwickelte LLM-Approximator verwendet werden. Dieser wird mit Daten trainiert, die eine Prädiktion des dynamischen Verhaltens des Bioreaktors erlauben. Zu diesem Zweck wird eine Trainingsdatenfolge

$$\mathcal{X}^{\text{train}} = \{C_1[t], C_2[t], v[t], \Delta C_1[t + T_M], \Delta C_2[t + T_M] \mid t = 1, T_M, 2T_M \dots T_B T_L T_M\} \quad (6.18)$$

mit

$$\Delta C_1[t + T_M] \equiv C_1[t + T_M] - C_1[t] \quad \text{und} \quad \Delta C_2[t + T_M] \equiv C_2[t + T_M] - C_2[t] \quad (6.19)$$

erzeugt, wobei die $v[t]$ von einem ‘‘Regler’’ herrühren, der die Ventilstellungen $v[t]$ unabhängig von jeglicher Messung würfelt. Die $v[t]$ sind gerade so gewählt, daß sich ein mittlerer Durchfluß von $u^{A,*} = 0.75$ bzw. $u^{B,*} = 1.25$ mit einer Streuung von 0.12 einstellt. Für die Erstellung von $\mathcal{X}^{\text{train}}$ wurde $T_B = 500$ und $T_L = 1000$ gewählt. In der Trainingsdatenfolge $\mathcal{X}^{\text{test}}$ ist somit kodiert, mit welcher ‘‘Geschwindigkeit’’ ($\Delta C_1[t + T_M], \Delta C_2[t + T_M]$) sich der Bioreaktor vom Zustand $(C_1[t], C_2[t], v[t])$ zum Zeitpunkt t entfernen wird.

Diese dermaßen generierte Testdatenfolge, die abhängig vom gewählten Szenario und der Totzeit T_M ist, wird nun einer ML-Dichteschätzung aus einer Mischung von M multivariaten Normalverteilungen unterzogen. Dazu wird der in Abschnitt 3.2 entwickelte *multivar*-Algorithmus verwendet. Wie dort gezeigt, arbeitet dieses Verfahren dichteorientiert, d.h. er verteilt die Zentren c_r der Normalverteilungen (2.36) entsprechend der tatsächlichen Dichte $p(\mathbf{x})$ so im Merkmalsraum, daß die Komponenten der Mischung (2.35) die *load balance* (2.21) zeigen. Dieses Verhalten ist aber für den mit (6.18) gegebenen Trainingsdatensatz $\mathcal{X}^{\text{train}}$ nicht unbedingt erwünscht, da gerade große aber seltene Geschwindigkeiten ($\Delta C_1[t + T_M], \Delta C_2[t + T_M]$) für den zu entwickelnden Regler von Interesse sind. Daher wird der *multivar*-Algorithmus so modifiziert, daß diese eher seltenen Ereignisse überproportional genau durch die Normalverteilungen (2.36) repräsentiert werden. Dies wird dadurch erreicht, daß der Lernparameter ϵ_c in der Lernregel (2.19) für die Zentren c_r durch den neuen Lernparameter

$$\tilde{\epsilon}_c[t] = \epsilon_c \frac{|(\Delta C_1[t + T_M], \Delta C_2[t + T_M])|}{|(\Delta C_1^{\text{max}}, \Delta C_2^{\text{max}})|} \quad (6.20)$$

ersetzt wird. Dadurch wird die Lernschrittweite jeweils an die vorliegende Geschwindigkeit ($\Delta C_1[t + T_M], \Delta C_2[t + T_M]$) angepaßt. Dabei ist $(\Delta C_1^{\text{max}}, \Delta C_2^{\text{max}})$ die im Betrag größte Geschwindigkeit, die im Datensatz $\mathcal{X}^{\text{train}}$ vorkommt. Mit dem durch (6.20) modifizierten *multivar*-Algorithmus erhält man schließlich eine Repräsentation der Trainingsdatenfolge $\mathcal{X}^{\text{train}}$, die gerade im Bereich großer Geschwindigkeiten aussagekräftig ist.

Aus dieser Dichteschätzung kann nun ein LLM-Approximator extrahiert werden, der durch inverse Systemidentifikation als Regler verwendet werden kann. Dieser Regler wird im folgenden auch als *LLM-Regler* bezeichnet. Der LLM-Approximator muß dazu gerade eine Abbildung von einem Zustand $(C_1[t], C_2[t])$ und einer Geschwindigkeit $(\Delta C_1[t + T_M], \Delta C_2[t + T_M])$ auf eine Ventilstellung $v[t]$ ausführen, also

$$\hat{f} : (C_1[t], C_2[t], \Delta C_1[t + T_M], \Delta C_2[t + T_M]) \in \mathbb{R}^4 \rightarrow \mathbb{R}. \quad (6.21)$$

Der LLM-Approximator prädiziert also aus einem Zustand und einer Geschwindigkeit eine Ventilstellung, die zu gerade dieser Geschwindigkeit führt. Mit der Zielgeschwindigkeit (6.15) erhält man schließlich durch

$$v[t] = \hat{f}((C_1[t], C_2[t], \Delta C_1^*[t], \Delta C_2^*[t]); \Theta^{LLM}) \quad (6.22)$$

einen datenbasierten Regler für den Bioreaktor.

6.3. Ergebnisse für verschiedene Szenarien und Regler

Die drei oben dargestellten Regler sollen nun an Hand der drei in Abschnitt 6.1 definierten Szenarien verglichen werden. Dabei wird immer eine Totzeit von $T_M = 10$ eingehalten. Jede dieser Simulationen wird anschließend nach den in Abschnitt 6.2.2 eingeführten Kriterien beurteilt. Da der einzige freie Parameter des LLM-Reglers die Anzahl M der Normalverteilungen (2.36) für die ML-Dichteschätzung (2.35) ist, werden die entsprechenden Simulationsläufe in Abhängigkeit von M ausgeführt. In Abbildung 6.7 sind die Ergebnisse aller Versuche zusammengefaßt.

Szenario A: Regelung im stabilen Bereich

Im Szenario A soll die Zellkonzentration und das Nährstoffverhältnis auf den stabilen Fixpunkt $C^{S,1}$ mit den Werten $C_1^{A,*} = 0.1207$ und $C_2^{A,*} = 0.8801$ geregelt werden. Bei diesen Zielgrößen ist eine mittlere Ausbeute von $\bar{A}^{A,*} = 0.0905$ zu erwarten.

Die erste Spalte in Abbildung 6.7 verdeutlicht, daß ein offener Regelkreis für diesen Fall zu einem sehr guten Ergebnis führt und daß der mittlere Fehler $\bar{e}[T]$ für alle Regler ähnlich klein ist. Da aber der Anteil der nicht verlorenen Simulationsläufe für den PID- und den *Fuzzy*-Regler bei nur 50% liegt, ergibt sich für diese beiden Regler eine geringe mittlere Ausbeute $\bar{A}[T]$. Der LLM-Regler regelt dagegen die Zielgrößen zuverlässig ein und erreicht damit die optimale mittlere Ausbeute $\bar{A}[T] = \bar{A}^{A,*}$ bei einem sehr geringen mittleren Fehler $\bar{e}[T]$.

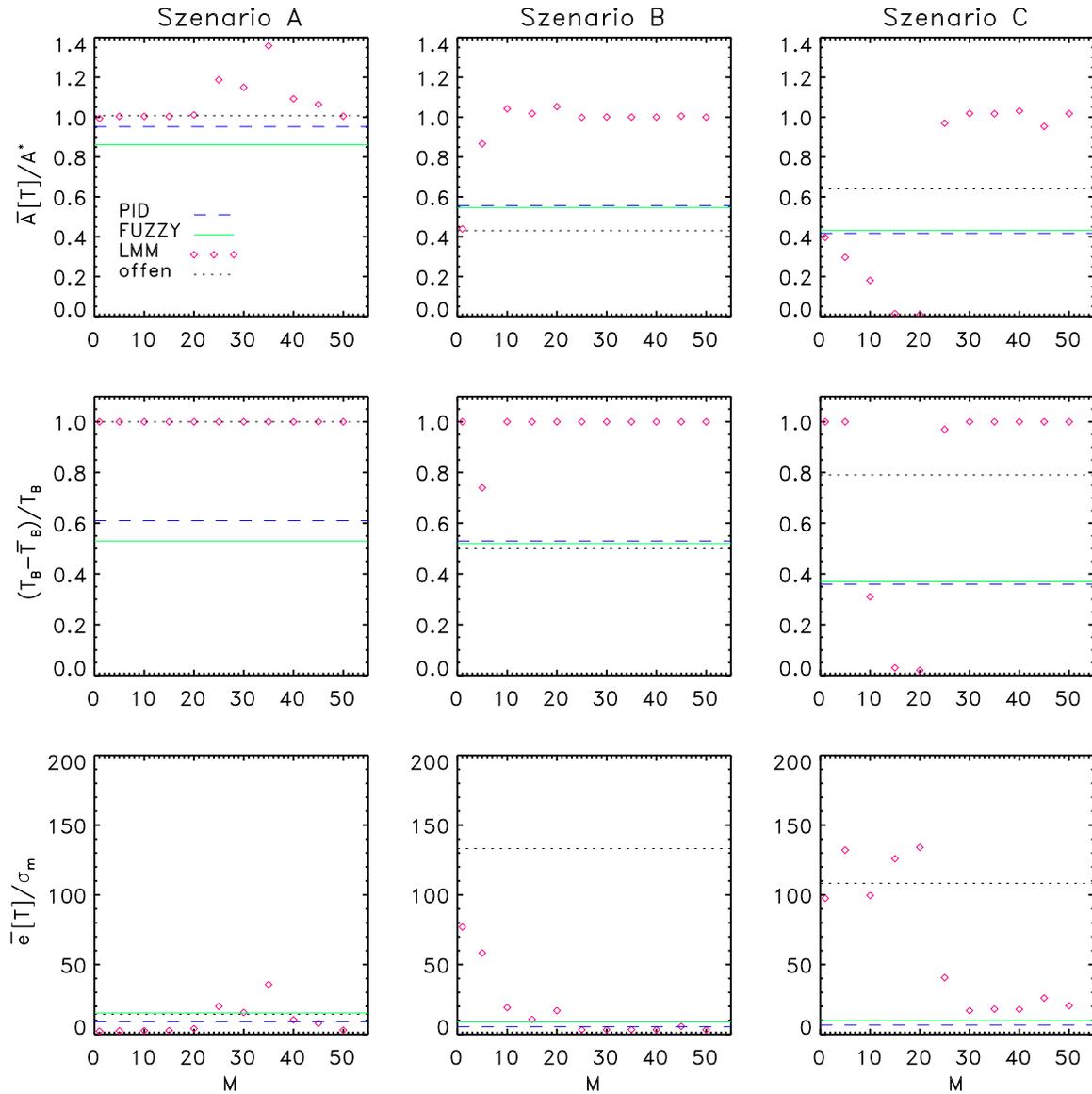


Abbildung 6.7.: **Ergebnisse für die Regelung des Bioreaktors.** In den drei Spalten sind die Ergebnisse für die Szenarien A, B und C dargestellt. Die drei Zeilen zeigen jeweils Resultate für die drei verschiedenen Kriterien. Dabei wird die mittlere Ausbeute $\bar{A}[T]$ auf die für das jeweilige Szenario erwartete mittlere Ausbeute \bar{A}^* bezogen. Die mittlere Zeile zeigt das Verhältnis der nicht verlorenen Simulationen $T_B - \bar{T}_B$ zur Gesamtzahl T_B aller Simulationenläufe und die letzte Zeile den mittleren Fehler $\bar{e}[T]$ normiert mit der Streuung σ_m des Meßrauschens R_m (vgl. auch Abbildung 6.5). Auf Grund dieser Normierungen ist ein Wert von Eins für alle Kriterien optimal. Die Regelung des Bioreaktors mit einem LLM-Approximator wurde in Abhängigkeit von der Anzahl M der verwendeten Normalverteilungen (2.36) durchgeführt. Neben dem PID-, *Fuzzy*- und LLM-Regler sind auch die Resultate für eine *offene* Strecke, d.h. einen nicht geregelten Bioreaktor, dargestellt.

Szenario B: Regelung im instabilen Bereich

Im Szenario B soll die Zellkonzentration und das Nährstoffverhältnis auf den instabilen Focus $C^{S,1}$ mit den Werten $C_1^{B,*} = 0.2107$ und $C_2^{B,*} = 0.7226$ geregelt werden. Bei diesen Zielgrößen ist eine mittlere Ausbeute von $\bar{A}^{B,*} = 0.2634$ zu erwarten.

Auch hier ist der Anteil der nicht verlorenen Simulationsläufe für den PID- und den *Fuzzy*-Regler, aber auch für den offenen Regelkreis, mit ca. 50% gering. Diese geringe Zuverlässigkeit der Regler zeigt sich auch in einer geringen mittleren Ausbeute $\bar{A}[T]$. Der mittlere Fehler ist dagegen im Vergleich zum offenen Regelkreis gering, da wegen (6.14) die verlorenen Simulationsläufe nicht in die Berechnung von $\bar{e}[T]$ eingehen. Der LLM-Regler zeigt für $M > 20$ eine nahezu optimale mittlere Ausbeute und zuverlässiges Verhalten mit einem geringen mittleren Fehler.

Szenario C: Regelung bei schnell wechselnden Systemzuständen

Bei diesem Szenario stimmen die Zielgrößen $C_1^{C,*}$ und $C_2^{C,*}$ mit denjenigen von Szenario B überein. Allerdings wird nun angenommen, daß das Stellventil auf Grund eines stochastischen Prozesses zwischen der linearen (6.11) und der quadratischen (6.12) Kennlinie wechseln kann.

In dieser Situation führen der PID- und der *Fuzzy*-Regler zu einem geringen Anteil nicht verlorener Simulationsläufe von ca. 35%. Entsprechend gering ist auch die mittlere Ausbeute $\bar{A}[T]$. Abgesehen vom mittleren Fehler $\bar{e}[T]$ liefert ein offener Regelkreis deutlich bessere Ergebnisse als der PID- oder *Fuzzy*-Regler. Dagegen meistern LLM-Regler, die aus mehr als 30 multivariaten Normalverteilungen aufgebaut sind, die gestellte Aufgabe hervorragend. Sowohl die mittlere Ausbeute $\bar{A}[T]$ als auch die Zahl der nicht verlorenen Simulationsläufe ist im optimalen Bereich. Da der mittlere Fehler $\bar{e}[T]$ von PID- und *Fuzzy*-Regler auf Grund der großen Zahl verlorener Simulationsläufe gering ist, erscheint allerdings der mittlere Fehler der sehr zuverlässigen LLM-Reglers relativ hoch.

Es bleibt anzumerken, daß die sehr gute Qualität des LLM-Reglers auch mit einem datenbasierten Regler erreicht wird, der auf dem im Abschnitt 4.3.1 beschriebenen mGRBF-System von M. Sonner beruht [Son97]. Allerdings verfügt ein solcher Regler wegen der äquidistanten Verteilung der Zentren c_r im fünfdimensionalen Merkmalsraum über wesentlich mehr Parameter als der LLM-Regler und ist daher auch entsprechend zeitintensiv im Training.

Zusammenfassung

In diesem Kapitel wurde der Bioreaktor als Testsystem für den Vergleich eines datenbasierten Reglers, der auf einem LLM-Approximator beruht, mit einem PID- und einem *Fuzzy*-Regler, verwendet. Ein einfacher Bioreaktor kann durch ein nichtlineares, gekoppeltes Differentialgleichungssystem beschrieben werden. Die daraus resultierende, komplexe

Dynamik wurde zunächst näher diskutiert, um einen Eindruck von der Schwierigkeit der Regelung eines Bioreaktors zu gewinnen. Der Vergleich des sog. LLM-Reglers mit einem PID- und einem *Fuzzy*-Regler wurde für drei Szenarien vorgenommen, die sich durch eine zunehmende Nichtlinearität der Dynamik des Bioreaktors auszeichnen. Es zeigte sich, daß es dem LLM-Regler in den drei untersuchten Fällen gelingt, ein ausreichend gutes Modell der Dynamik zu erstellen, um – im Vergleich zu konventionellen Reglern – den Bioreaktor sehr sicher und genau zu regeln.

7. Regelung der Rückstandsverbrennung in einer industriellen Anlage

Bei den Beispielen, an Hand derer in den letzten Kapiteln Verfahren demonstriert und verglichen wurden, handelte es sich immer um künstliche und konstruierte Systeme, die nur auf dem Computer simuliert wurden. Will man aber die Praxistauglichkeit von Regelungs-
algorithmen testen, so muß man sich auch an die Kontrolle von Systemen wagen, die tatsächlich existieren, und die in der Praxis auch von Bedeutung sind. Dies soll in diesem Kapitel geschehen.

Durch eine Kooperation mit der Wacker-Chemie GmbH bestand die Möglichkeit, neue Regelungskonzepte für ein reales, industrielles Regelungsproblem an der Anlage für Rückstandsverbrennung im Werk Burghausen zu implementieren und zu testen. Bei dieser Anlage handelt es sich um ein großes und komplexes System, das mit allen Eigenarten ausgestattet ist, die die Umsetzung theoretischer Verfahren in die Praxis erschweren: unzuverlässige und instabile Messungen, stochastische Störungen, Stellglieder mit unbekanntem Kennlinien, ein Prozeßleitsystem mit nur eingeschränkter Funktionalität, harte Randbedingungen durch Wirtschaftlichkeit und Umweltschutz, Forderung nach sehr hoher Verfügbarkeit und Widerstände der Betreiber bei der Umsetzung neuer Methoden. Langfristiges Ziel der Wacker-Chemie GmbH war eine Stabilisierung des instabilen Verhaltens der Anlage unter der Nebenbedingung, den Verbrauch an Stützbrennstoff zu reduzieren. Im Rahmen einer Machbarkeitsstudie wurde für diesen Zweck ein Programmpaket entwickelt, welches unter Zuhilfenahme des vorhandenen Prozeßleitsystems ein System lokaler Experten realisiert. Dieses neue System gliedert sich in zwei Teile – Regelung der Abgase und der Flüssigkeiten – und ersetzt bzw. unterstützt die schon vorhandene Regelung. Auf Grund der Komplexität der Anlage und der Forderung, den laufenden Betrieb möglichst wenig zu stören, konnten nur bestimmte Teilbereiche der Anlage mit den in Kapitel 4 und 6 beschriebenen Methoden kontrolliert werden. Dazu zählt eine Verbesserung der Lastenverteilung auf die drei vorhandenen Öfen und eine Linearisierung der Flüssigkeitszufuhr.



Abbildung 7.1.: **Gesamtansicht der Anlage für Rückstandsverbrennung.** Links sind die aus dem Werk einlaufenden Abgasleitungen zu erkennen, die in die Abgasübernahme münden. Dieser Anlagenteil verdeckt die beiden kleineren Öfen 1 und 2 und das Betriebsgebäude. Der mittlere Teil des Bilds ist von verschiedenen Filteranlagen (Rauchgaswäsche und Naßelektrofilter) und einem großen Flüssigkeitstank geprägt. Rechts sind Ofen 3 und ein Teil des Katalysators zu sehen. Im Vordergrund fließt der Alz-Salzach-Kanal vorbei.

7.1. Die Anlage zur Rückstandsverbrennung

Um die Bedeutung und die Aufgabe der Anlage für Rückstandsverbrennung besser einordnen zu können, soll zunächst der Standort des Chemiewerks kurz beschrieben werden. Das Werk Burghausen der Wacker-Chemie GmbH umfaßt etwa 30 Produktionsbetriebe auf einer Fläche von ca. 2 qkm. Das um 1910 gegründete Stammwerk produziert hauptsächlich Grundstoffe für die chemische Industrie. Hauptzweig war lange Zeit die Chlorchemie für Vorprodukte der PVC-Herstellung. Neuerdings wird dieser Bereich aber zunehmend von der Produktion verschiedener Silikone, organischer Zwischenstoffe und Reinstsiliziums abgelöst [WCG99]. Die auch auf dem Werksgelände angesiedelte Wacker Silitronic AG produziert seit 1962 Siliziumwafer für die Halbleiterindustrie und hält weltweit inzwischen einen Marktanteil von 14%. 1997 waren etwa 10000 Beschäftigte im Werk Burghausen angestellt. Der Gesamtumsatz der Wacker-Chemie GmbH lag 1997 bei rund 5 Milliarden DM [WCG97].

Der 1967 gebaute Betriebsteil für Rückstandsverbrennung (RV) entsorgt neben der werkeigenen Kläranlage und Deponie feste, flüssige und gasförmige Abfälle, die innerhalb des Werks anfallen. Abbildung 7.1 zeigt die Anlage in der Totalen. Die folgende Beschreibung wird kurz auf die wichtigsten Anlagenteile eingehen. Anschließend werden diejeni-

gen Aspekte, die für die weiter unten vorgestellte Regelung von Bedeutung sind, genauer dargestellt. Die RV läßt sich in vier Teile gliedern:

- **Abgasübernahme und Flüssigkeitsanlieferung.** Die gasförmigen Abfallstoffe werden über zum Teil lange Rohrleitungen (1-2 km) zur RV transportiert und in der Abgasübernahme¹ auf die verschiedenen Öfen verteilt. Bemerkenswert ist, daß die Abgase (bis auf Abgas 10) von den Teilbetrieben nicht gepuffert werden. Die RV ist also *gezwungen*, alle angelieferten Abgase sofort zu verbrennen. Die Flüssigkeiten werden dagegen in Tankwagen angeliefert und können in zwei großen Tanks zwischengelagert werden.
- **Drei Verbrennungsöfen.** Dies sind zwei sog. Muffelöfen, die seit 1956 (Ofen 1) und 1964 (Ofen 2) in Betrieb sind und ein neuerer Drehrohrofen aus dem Jahr 1982. Da die für die Regelung wesentlichen Prozesse in diesen Öfen ablaufen, wird im nächsten Abschnitt noch näher auf den Aufbau der Öfen eingegangen.
- **Filteranlagen.** Auf Grund der immer schärfer werdenden Ofenabgasverordnungen sind in diesem Anlagenteil die größten Veränderungen, bzw. Anbauten vorgenommen worden. In den fünfziger Jahren arbeiteten die beiden Muffelöfen noch ohne irgendwelche Filtervorrichtungen. Seitdem ist die Anlage schrittweise mit verschiedenen Rauchgaswäschern, einem Naßelektrofilter und einem Katalysator aufgerüstet worden.
- **Leitstand.** Hier laufen die Werte von ca. 1000 Messungen, die in der Anlage vorgenommen werden, zusammen. Alle wichtigen Steuerungs- und Regelungsaufgaben werden dort mit Hilfe eines Prozeßleitsystems (PLS) zentral ausgeführt. Im regulären Betrieb wird das Prozeßleitsystem von mindestens einem/einer Chemiefacharbeiter/in bedient. Zusätzlich ist mindestens eine Person für die Bedienung und Überwachung der Anlage vor Ort notwendig. Die normale Besetzungstärke liegt tagsüber bei 4-5 Arbeitern und ca. 6 Angestellten für die verfahrenstechnische Betreuung und die Verwaltung.

7.1.1. Öfen und Brennrecken

Bei den beiden älteren und kleineren Öfen (Ofen 1 und 2) handelt es sich um Muffelöfen. Bei diesen Öfen ist der zylinderförmige Brennraum aufrecht stehend angeordnet, wobei die Brennlanzen, welche die zu verbrennenden Stoffe dem Brennraum zuführen, radial in gleicher Höhe angebracht sind. Ofen 1 hat eine Höhe von 4.2 Metern bei einem Durchmesser von 2.2 Metern. Mit 5.9 m × 2.5 m ist der Ofen 2 etwas größer. Der neuere Ofen 3 ist als moderner Drehrohrofen ausgeführt. Bei diesem ist der zylindrische Brennraum (ca. 5 m ×

¹Im folgenden werden die gasförmigen Abfallstoffe auch als Abgase bezeichnet. Um Verwechslungen zu vermeiden, werden die Abgase des Ofens als Ofenabgas bezeichnet.

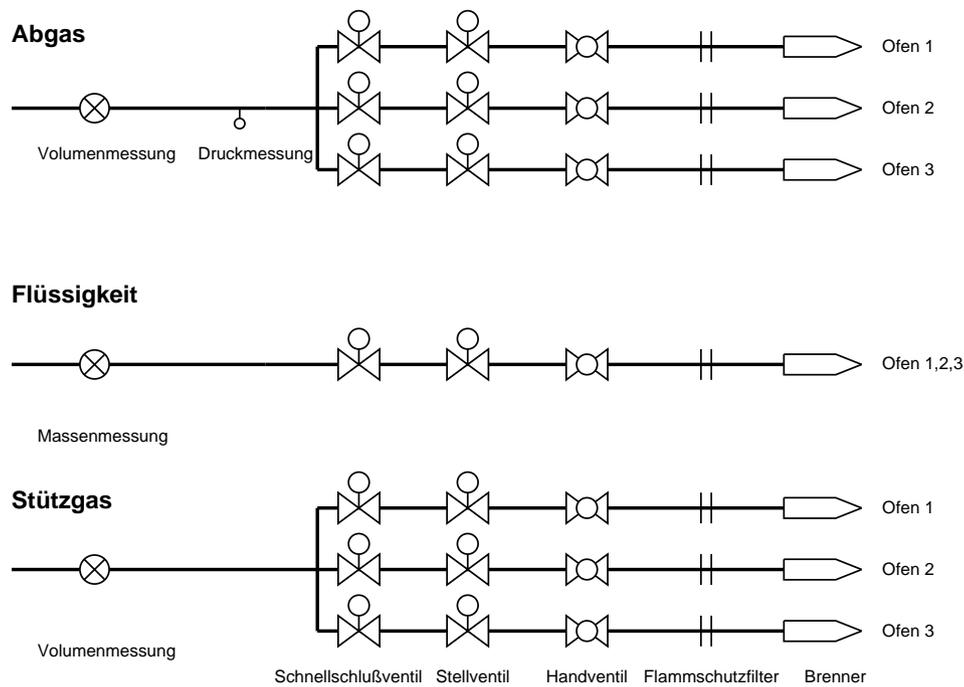


Abbildung 7.2.: Aufbau der Brennstrrecken von Abgas, Flüssigkeit und Stützgas.

4 m) waagrecht mit einer leichten Schräglage und um die Längsachse drehbar montiert. Durch die langsame Rotation des Brennraums während des Betriebs wird vor allem für die Feststoffe eine bessere Verbrennung und ein Abfließen der Schlacke erreicht.

Die für die Kontrolle der Öfen wichtigsten Meßgrößen sind die Temperatur und die Sauerstoffkonzentration. Jeder Ofen ist mit jeweils zwei Temperaturfühlern im Feuerraum, in der Ausmauerung und in der Nachbrennkammer ausgestattet. Die Messungen zeigen verschiedene Zeitkonstanten. So spricht die Messung der Feuerraumtemperatur mit 30 s Mittelungszeit am schnellsten an. Die Messungen der Nachbrennkammer- und Ausmauerungstemperatur sind mit ca. 45 s bzw. 60 s deutlich langsamer. Die mit etwa 30 s relativ schnelle Messung der Sauerstoffkonzentration zeigt, wie durch eine Meßdatenanalyse festgestellt werden konnte (vgl. Abbildung 7.3 weiter unten), erwartungsgemäß eine deutliche Antikorrelation mit den verschiedenen Temperaturmessungen. Für die Messung des Ofeninnendrucks existiert nur eine Meßstelle. Neben diesen für die Kontrolle wichtigen Größen, werden im Ofenabgas die Konzentrationen verschiedener Schadstoffe gemessen (Staub, Stickoxide, Kohlenmonoxid, organische Kohlenwasserstoffe).

Bei allen Öfen kann die für die Verbrennung notwendige Luftmenge durch eine Luftklappe beeinflußt werden. Die Messung des Luftstroms liefert typischerweise Werte von 2000-4000 m³/h. Das maximale Volumen, welches durch einen Ofen bewegt werden kann, die sog. hydraulische Grenze, liegt jeweils bei etwa 10000-12000 m³/h.

Abgase

Jeder Ofen verfügt über jeweils 12 Brennlampen, die sog. *Brenner*, für gasförmige Abfallstoffe. Da die Abgasleitung 7 nicht mehr existiert, werden die Abgasleitungen mit 1...6, 8...13 durchnummeriert². Abbildung 7.2, oben, zeigt das Schema der Aufteilung eines Abgases auf die drei Öfen. Für jedes Abgas ist eine Volumen- und eine Druckmessung installiert. Diese Messungen erfolgen, *bevor* die Abgase auf die drei Öfen aufgeteilt werden. Der Volumenstrom liegt typischerweise bei ca. 100 m³/h. Abgas 6 und Abgas 8 können aber auch über längere Zeit Durchflüsse von etwa 600 m³/h erreichen. Jede Brennstrecke besteht aus der Sequenz³ Schnellschlußventil, Stellventil, Handventil, Flammschutzfilter und Brenner. Mit dem Schnellschlußventil (SS-Ventil) kann ein Brenner im Notfall innerhalb einiger Zehntelsekunden abgeschaltet werden. Um ein SS-Ventil zu öffnen, muß allerdings aus Sicherheitsgründen vor Ort am SS-Ventil ein Taster betätigt werden. Die eigentliche Steuerung des Volumenstroms wird aber durch die Stellventile bewirkt. Mit dem Handventil kann zusätzlich der Durchfluß manuell angedrosselt werden. Die Stellung des Handventils wird aber nicht gemessen und liegt somit auch nicht in elektronisch auswertbarer Form vor. Der kurz vor dem Brenner angebrachte Flammschutzfilter hat die Aufgabe, ein Zurückschlagen der Flamme in die Abgasleitung zu verhindern. Oben wurde schon erwähnt, daß alle Abgase (bis auf Abgas 10) ungepuffert angeliefert werden, d.h. es besteht für die RV ein Abnahmepflicht. Da die Produktionsbetriebe einen annähernd konstanten Unterdruck am Ende der Abgasleitungen für ihre Prozeßführung erwarten, kann das teilweise recht große Leitungsvolumen (bis 30 m³) nur eingeschränkt als Notpuffer verwendet werden. Entsprechend muß der Abgasdruck durch Variation der drei Stellventile auf einen konstanten Wert (üblicherweise im Bereich von etwa 40 bis 150 mbar) geregelt werden.

Flüssigkeiten

Der Aufbau der Brennerstrecken für die Flüssigkeiten (siehe Abbildung 7.2) ähnelt sehr den oben beschriebenen Abgasbrennstrecken. Allerdings werden die Flüssigkeiten nicht auf die drei Öfen aufgeteilt. Vielmehr ist jeder Brenner einem bestimmten Tank oder dem Stellplatz eines Tankwagens zugeordnet. Ofen 1 verfügt über vier Flüssigkeitsbrenner (Stellplätze CADMA, LOESM, ANFAHR und Tank B202), Ofen 2 über 8 Brennerstrecken (Stellplätze CADMA, LOESM, ANFAHR, KONDENS, DIKETEN, SILAN und die Tanks B104, B202) und Ofen 3 über 8 Brenner. Man beachte, daß die Namen die Brennerstrecken bezeichnen und nur bedingt Rückschlüsse auf die Art der Flüssigkeiten zulassen. Auf Grund ihrer sehr unterschiedlichen physikalischen und chemischen Eigenschaften werden die Flüssig-

² Auf Grund der langsamen Veränderungen der Anlage im Laufe der letzten Jahrzehnte hat sich eine Reihe von verwirrenden Bezeichnungen eingebürgert.

³ Tatsächlich verfügt jede Brennstrecke über deutlich mehr Elemente; so z.B. über Ventile zum Spülen der Leitungen mit Stickstoff. Für die folgenden Ausführungen ist dies aber nicht von Bedeutung.

keiten nicht mit Pumpen zu den Öfen befördert, sondern mit Stickstoff unter Druck zu den Brennern gepreßt. Die Massenströme können dabei Werte von bis zu 400 kg/h erreichen.

Stützbrennstoff

Neben den Abgasen und Flüssigkeiten wird in den Öfen ein sog. Stützbrennstoff (Erdgas oder Wasserstoff) verfeuert, um die Verbrennung schlecht brennbarer Abfallstoffe zu ermöglichen. Da die Abgase auf Grund ihrer variablen Zusammensetzung (z.B. abwechselnd verunreinigtes Schutzgas und Acetylen) einen ausgeprägt stochastischen Einfluß auf den physikalischen Zustand der Öfen haben, führt die Zufeuerung von Stützbrennstoff auch zu einer gewollten Stabilisierung der Verbrennungsdynamik. Zusätzlich wird der Stützbrennstoff beim Anfahren eines Ofens dazu verwendet, die mindestens erforderliche Brennraumtemperatur von 900 °C zu erreichen. Als Stützbrennstoff wird sowohl Wasserstoff als auch Erdgas zugefeuert. Allerdings kann während des laufenden Betriebs nicht zwischen diesen beiden Gasen umgeschaltet werden.

7.1.2. Randbedingungen und Sicherheitsaspekte

Um die Sicherheit der Anlage zu gewährleisten und die Abgabe von Schadstoffen in den Ofenabgasen zu begrenzen, dürfen bestimmte Grenzwerte nicht unter- oder überschritten werden. Die Einhaltung dieser Grenzwerte wird überwacht, indem eine vom Technischen Überwachungsverein (TÜV) bereitgestellte speicherprogrammierbare Steuerung (SPS) laufend die Anlage kontrolliert. Wird eine der unten genauer spezifizierten Vorgaben verletzt, schaltet die SPS den entsprechenden Ofen sofort aus, indem automatisch alle SS-Ventile geschlossen werden. Da ein neues Anfahren des betroffenen Ofens ca. 10 Minuten in Anspruch nimmt, ist ein solches Über- bzw. Unterschreiten der Grenzwerte natürlich auch aus betrieblicher Sicht nicht erwünscht. Erschwerend kommt hinzu, daß bei Ausfall eines Ofens die beiden verbleibenden Öfen durch Überlastung ebenfalls ausfallen können, da dann dort diejenigen Abgase, die ungepuffert zur RV geleitet werden, zusätzlich verbrannt werden müssen. So kann eine regelrechte Kettenreaktion ablaufen, die schließlich zum Ausfall der gesamten RV führt und damit auch zum Stillstand von Produktionsbetrieben im Werk, die auf eine zuverlässige Abgasentsorgung angewiesen sind. Gerade wenn zu Wartungszwecken ein Ofen (und hier besonders der durchsatzstarke Ofen 3) abgeschaltet ist, wird die Wahrscheinlichkeit einer solchen Kettenreaktion groß. Die vom TÜV vorgegebenen Grenzwerte, die zu einer Ofenabschaltung führen, sind:

- Der Ofeninnendruck darf nur für einige Sekunden den Normaldruck um 1 mbar übersteigen. Die Verbrennung läuft in allen Öfen im leichten Unterdruck (ca. 5 mbar unter Normaldruck) ab, um ein Zurückschlagen der Flamme in die Brenner zu verhindern.

- Die Sauerstoffkonzentration darf nur für einige Sekunden 4 Volumenprozent unterschreiten. Dies verhindert die Bildung einer explosionsfähigen Gasmischung durch unvollständige Verbrennung.
- Der in einen Ofen eingesaugte Luftstrom muß einen Wert von 1700 m³/h überschreiten. Dies soll ebenfalls die Entstehung einer explosionsfähigen Gasmischung verhindern.
- Die Temperatur in der Nachbrennkammer eines Ofens muß stets einen Wert von 900 °C überschreiten. Eine hohe Temperatur ist notwendig, um organische Kohlenstoffe im ausreichenden Maße zu zerstören.

7.1.3. Prozeßleitsystem der Rückstandsverbrennung

Auf die meisten Sensoren und Aktoren der Anlage kann über das Prozeßleitsystem *Contronic P* der Fa. Hartmann & Braun zugegriffen werden. Ein PLS wird zur Steuerung, Regelung und Visualisierung von verfahrenstechnischen Prozessen verwendet. Solche Systeme sind meist als sehr zuverlässige Spezialcomputer⁴ ausgeführt, die mit einem Echtzeitbetriebssystem ausgestattet sind. Damit wird eine definierte Reaktionszeit auf wichtige Ereignisse garantiert. Ein PLS liest zyklisch (beim *Contronic P* alle 20 ms) Meßdaten der verschiedenen Sensoren ein, verrechnet diese und gibt anschließend Stellgrößen an die Aktoren aus. Die Konfiguration dieser Systeme (wie auch des *Contronic P*) erfolgt durch Verknüpfen vordefinierter Funktionsblöcke mit den Eingabe- und Ausgabekanälen. Eine Programmierung von selbstdefinierten Funktionen ist dagegen erst bei Systemen jüngster Bauart möglich. Neben der Visualisierung aller relevanten Meßdaten, hat das PLS in der RV die folgenden regelungstechnischen Aufgaben:

- **Regelung der Stellventile auf eine gewünschte Stellgröße.** Ein Stellventil ist im wesentlichen aus einem Ventil, einem Elektromotor und einem Winkelabnehmer aufgebaut. Daher kann die Einstellung einer bestimmten Ventilöffnung mit einem einfachen Regelkreis vorgenommen werden. Im folgenden werden diese Regelkreise jedoch stets vernachlässigt. Wenn von einer bestimmten Ventilstellgröße die Rede ist, wird angenommen, daß das PLS diese Stellgröße auch korrekt einregelt.
- **Regelung der Flüssigkeitsventile auf einen vorgegebenen Massenstrom.** Oben wurde erläutert, daß jede Flüssigkeitszuleitung mit einer Massenstrommessung ausgestattet ist. Die im PLS implementierten PID-Regler zur Einstellung vorgegebener Massenströme sind aber aus verschiedenen Gründen (siehe nächster Abschnitt) nur sehr bedingt brauchbar.

⁴Auf Grund der Leistungsfähigkeit der neuesten Computergeneration wird inzwischen eine Reihe moderner Prozeßleitsysteme angeboten, die auf *workstations* beruhen (z.B. *Simatic PCS 7* von Siemens oder *DeltaV* von Fisher-Rosemount).

Physikalischer Ofenzustand		
T_{FR}^σ	°C	Feuerraumtemperatur
T_{NK}^σ	°C	Nachbrennkammertemperatur
T_{AM}^σ	°C	Ausmauerungstemperatur
O^σ	Vol%	Sauerstoffkonzentration
P^σ	mbar	Ofeninnendruck

Zustand der Abgasleitungen		
p_j	mbar	Abgasdruck
v_j	m ³ /h	Abgasvolumenstrom
y_j^σ	%	Stellung des Abgasventils
s_j^σ	0/1	Zustand des SS-Ventils

Zustand der Flüssigkeitsleitungen		
M_j^σ	kg/h	Massenstrom der Flüssigkeit
Y_j^σ	%	Stellung des Flüssigkeitsventils
S_j^σ	0/1	Zustand des SS-Ventils

Führungsgrößen		
Ziel p_j	mbar	Abgasdruck
Ziel M_j^σ	kg/h	Massenstrom der Flüssigkeit

Stellgrößen		
Soll y_j^σ	%	Stellung des Abgasventils
Soll Y_j^σ	%	Stellung des Flüssigkeitsventils

Tabelle 7.1.: **Die für die Beschreibung der Öfen wichtigsten Größen.** Der hochgestellte Index $\sigma = 1, 2, 3$ kennzeichnet hierbei den jeweiligen Ofen. Generell gilt für alle folgenden Ausdrücke, daß ein griechischer Index die Öfen indiziert. Der tiefergestellte Index j kennzeichnet das entsprechende Ventil. Bei den Abgasen ist $j = 1 \dots 6, 8 \dots 13$, bei den Flüssigkeiten $j = 1 \dots 4$ für Ofen 1 und $j = 1 \dots 8$ für Ofen 2. Um die Ventile der Abgase und der Flüssigkeiten einfach unterscheiden zu können, werden die ersteren mit Kleinbuchstaben und die letzteren mit Großbuchstaben gekennzeichnet. Ein SS-Ventil im Zustand 0 ist geschlossen und im Zustand 1 geöffnet.

- **Regelung der Abgasdrücke auf einen konstanten Druck.** Um eines der Abgase auf die drei Öfen aufzuteilen, wird ein sog. *split-range*-Regler verwendet [Föl92]. Dabei berechnet zunächst ein PID-Regler aus dem aktuellen Abgasdruck und dem Sollwert eine Stellgröße zwischen 0% und 300%. Diese Stellgröße wird anschließend in einer fest vorgegebenen Reihenfolge an die drei Ventile weitergegeben. So wird bei einer Reihenfolge 2-3-1 zunächst nur das Abgasventil von Ofen 2 geöffnet. Wenn die Ausgabe des Reglers 100% überschreitet, wird das Abgasventil des dritten Ofens geöffnet und ab 200% sind alle Öfen an der Verbrennung des Abgases beteiligt.

Die oben skizzierte Regelung durch das PLS nimmt auf etwa 50 Ventilstellungen Einfluß, wobei der aktuelle Zustand der drei Öfen durch ca. 150 Größen beschrieben wird. Tabelle 7.1 faßt diese Größen in Gruppen zusammen und gibt eine Übersicht über die verwendeten Formelzeichen und Einheiten.

7.1.4. Probleme bei der Regelung der Rückstandsverbrennung

Wie oben schon an einigen Stellen angedeutet wurde, führt der Aufbau der RV zu einigen Regelungsproblemen, die nun genauer diskutiert werden sollen. Dabei wird im folgenden stets zwischen der Regelung der Flüssigkeiten und der Abgase unterschieden.

Flüssigkeiten

Da die chemische Zusammensetzung der angelieferten Flüssigkeiten nur eingeschränkt bekannt ist, kann ihre Verbrennungsenthalpie nur grob oder gar nicht geschätzt werden. Allerdings lassen die verschiedenen Flüssigkeitsbrennstrecken einen gewissen Rückschluß auf die Brennbarkeit zu. So werden über die Leitungen LOESM und ANFAHR meist Flüssigkeiten mit hohem Brennwert verfeuert. Eine erste Abschätzung über die Eigenschaften einer Flüssigkeit erhalten die Anlagenfahrer dadurch, daß sie beim ersten Zufeuern mittels der Veränderung der Temperatur und Sauerstoffkonzentration die Reaktion des Ofens beobachten. Entsprechend erhöhen oder verringern sie die Zufeuerung von Stützbrennstoff. Diese Einstellung muß aber immer wieder überprüft werden, da sich die Zusammensetzung einer Flüssigkeit langsam (d.h. innerhalb einiger Stunden) auf Grund von Dichtegradientenbildungen in den Tanks bzw. Tankwagen ändern kann.

Abbildung 7.3 illustriert an einem fünfstündigen Ausschnitt aus den archivierten Daten die Dosierung für eine typische Flüssigkeit. Aus diesem Beispiel wird klar deutlich, wie unberechenbar sich ein Flüssigkeitsventil verhalten kann und wie stark ein Massenstrom trotz konstanter Ventilstellung variieren kann.

Einige Flüssigkeiten (besonders die Leitungen CADMA und B202) sind nur schwer dosierbar. Die Gründe dafür sind eine hohe Zähflüssigkeit, schnell variierende Konsistenz durch Klümpchen-Bildung und möglicherweise Ablagerung von Feststoffen in den Ventilen. Als Folge sind die Kennlinien der Ventile stark nichtlinear. Teilweise ist sogar ein rapider Anstieg des Durchflusses bei nur minimaler Ventilverstellung zu beobachten. Vermutlich ist dieses Verhalten darin begründet, daß sich bei einem hohen Durchfluß Ablagerungen im Ventil ablösen. Allerdings ist aber auch die Massenstrommessung zeitweise äußerst unzuverlässig, so daß häufig nur durch eine eingehende Analyse die Ursachen von Schwankungen dingfest gemacht werden können. Auch die oben schon angesprochene Schichtung in den Tanks hat einen Einfluß auf die zeitliche Entwicklung des Durchflusses und der Ofendynamik. Eine weitere Schwierigkeit stellt die gelegentliche manuelle Verstellung der Handventile durch die Anlagenfahrer dar. Da die Stellungen der Handventile nicht gemessen werden, kann daher häufig aus den im PLS vorliegenden Daten keine korrekte Aussage über die effektive Ventilstellung einer Zuleitung gemacht werden.

Auf Grund der schwierigen Dosierbarkeit wird die im PLS vorgesehene Regelung der Flüssigkeiten nur bei den beiden Leitungen LOESM und ANFAHR häufiger angewandt. Bei den restlichen Leitungen besteht das übliche Vorgehen der Anlagenfahrer darin, die Flüssigkeitsventile nach Anschluß eines Tanks oder Tankwagens auf einen mittleren Massenstrom (ca. 200 kg/h) einzustellen. Diese Ventilstellung wird dann möglichst nicht mehr verändert, bis der Tank nach vielen Stunden entleert ist. Durch dieses Vorgehen wird die mögliche Kapazität eines Ofens bewußt nicht ausgenutzt, um sich durch eine relativ hohe Sauerstoffkonzentration (7-9 Vol%) genügend Spielraum für Belastungsspitzen (z.B. plötzliche, unvorhergesehene Zufuhr gut brennbarer Abgase) zu erhalten.

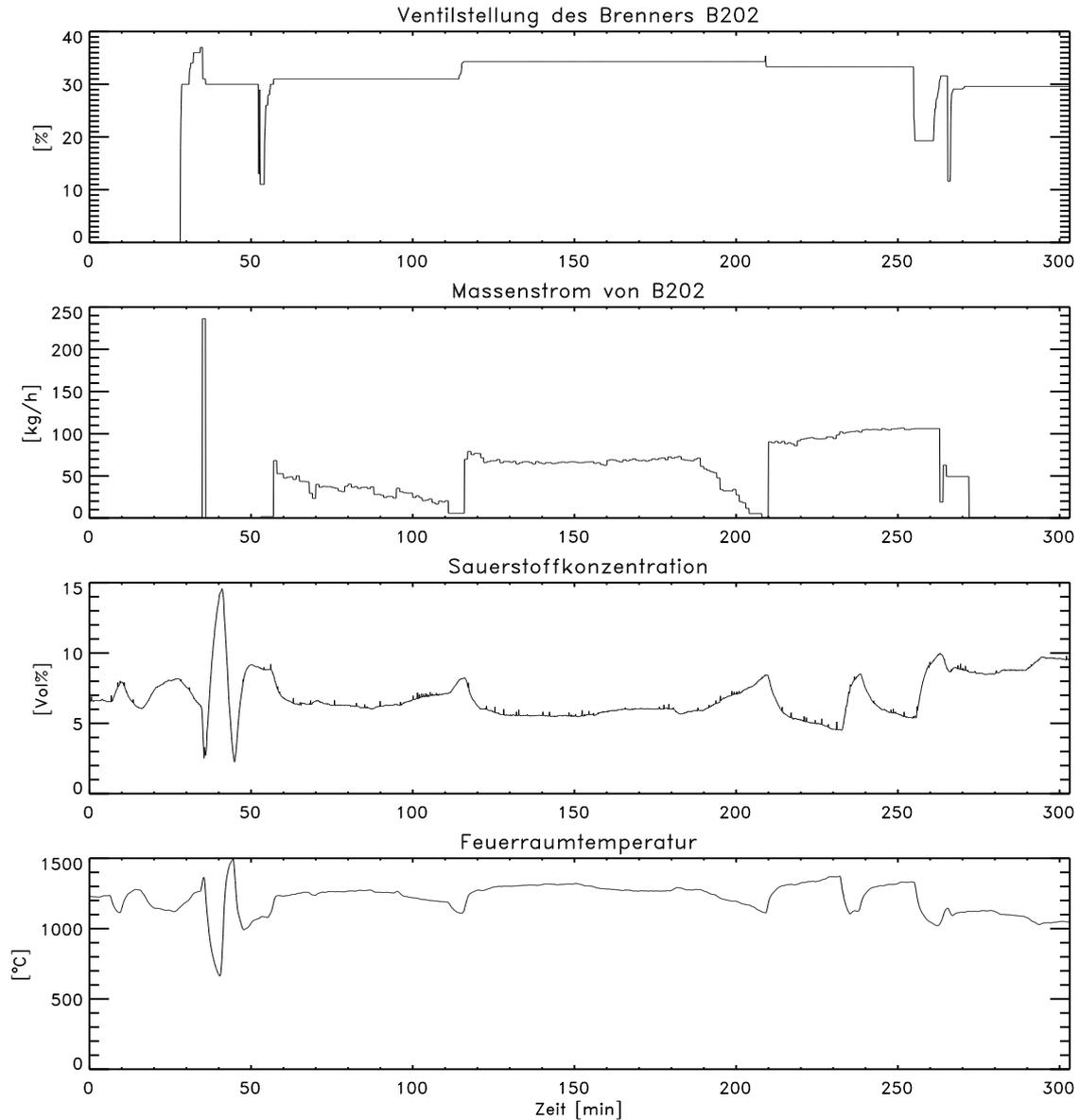


Abbildung 7.3.: **Typisches Verhalten einer Flüssigkeitsstrecke.** Das Beispiel zeigt die Leitung B202 von Ofen 1 am 27. April 1997. Die manuellen Ventilverstellungen innerhalb der ersten fünfzig Minuten zeigen – bis auf einen kurzen und starken Schwall – zunächst kaum eine Wirkung auf den Durchfluß. Erst nachdem das Ventil bis auf 12 % zu- und dann wieder aufgedreht wurde (bei 55 min), kommt ein einigermaßen konstanter Durchfluß zustande, der aber innerhalb der nächsten Stunde deutlich abfällt und erst durch eine weitere Ventilverstellung (bei 115 min) wieder erhöht werden kann. Dieses Verhalten wiederholt sich anschließend wieder. Deutlich ist der Einfluß des Durchflusses auf die Sauerstoffkonzentration zu erkennen. Insbesondere der erste Schwall (bei 35 min) führt zu einem starken Abfall der Sauerstoffkonzentration und damit fast zu einem Ausfall des Ofens. Die Temperatur ist zur Sauerstoffkonzentration antikorreliert.

Abgase

Im Gegensatz zu den Flüssigkeiten verursacht die Verbrennung der Abgase eine starke Kopplung zwischen den Öfen, da hohe Abgasvolumenströme nicht nur in einem Ofen verbrannt werden können. Vielmehr müssen mehrere Öfen zusammenarbeiten, um den von den Teilbetrieben geforderten konstanten Abgasunterdruck an den Leitungsenden in der RV einzuhalten. Diese Kopplung kann bei sehr hohen Belastungen auch zum Ausfall der gesamten Anlage führen. Eine für den laufenden Betrieb übliche Situation ist daher, daß durch die *split-range*-Regelung plötzlich Abgas an einen Ofen geleitet wird, um Drucksitzen in den Leitungen zu vermeiden.

Die meisten Abgasproduzenten bringen keinen konstanten Volumenstrom in die Leitungen ein, so daß Volumenschwankungen zu ständigen stochastischen Störungen der Ofenzustände führen. Da kein elektronischer Informationsaustausch zwischen Produzenten und dem PLS der RV existiert, sind auch keine Prognosen über zukünftige Belastungsspitzen möglich. Tritt ein hoher Volumenstrom auf, muß daher von den Anlagenfahrern innerhalb von einigen Minuten eine Flüssigkeit manuell angedrosselt werden, um einen kritischen Zustand zu vermeiden. Um die stoßartigen Belastungen, die durch die *split-range*-Regelung auf einen Ofen einwirken, etwas abzumildern, wird von den Anlagenfahrern häufig die Abgaszufuhr durch Verwendung der Handventile reduziert. Da – wie bei den Flüssigkeiten – die Stellungen der Handventile aber nicht aufgezeichnet werden, kann dadurch die nachträgliche Analyse der Ventilstellungen und Abgasdruckverläufe sehr erschwert werden. Hauptursache für die Notwendigkeit einer schnellen Reaktion durch die Anlagenfahrer sind die PID-Regler des PLS, die ständig versuchen den Solldruck $^{Ziel}p_j$ in den Abgasleitungen exakt einzuhalten. Hier ist kritisch anzumerken, daß diese Genauigkeit von den Abgasproduzenten im Grunde gar nicht gefordert wird.

Abbildung 7.4 zeigt an einem ca. einstündigen Beispiel die Arbeitsweise der *split-range*-Regelung. Diese Regelung erreicht durch ein massives Eingreifen, daß der Abgasdruck trotz des sehr schnellen Volumenstromanstiegs nur für kurze Zeit vom Sollwert abweicht. Ofen 2 reagiert auf diese Mehrbelastung durch einen Abfall der Sauerstoffkonzentration. Dagegen zeigt die Sauerstoffkonzentration in Ofen 3 keine Reaktion auf das voll geöffnete Abgasventil. Möglicherweise ist bei diesem Ofen das entsprechende Handventil angedrosselt worden oder das Stellventil hat bei einer Öffnung von ca. 35% schon seinen maximalen Volumenstrom erreicht.

Auch das plötzliche Ausbleiben eines gut brennenden Abgases führt zu Problemen, da dann die übliche Feuerraumtemperatur von etwa 1100 °C innerhalb einiger Minuten unterhalb der notwendigen 900 °C fallen kann. Daher wird stets mit einer “guten” Flüssigkeit zugefeuert oder – wenn dies nicht möglich ist – ein zusätzliches Stützgas verbrannt. Eine starke Stützflamme beruhigt so die Schwankungen der Temperatur und der Sauerstoffkonzentration.

Jede Abgasleitung ist – im Gegensatz zu den Flüssigkeiten – eindeutig einem Produzenten zugeordnet. Daher sind bestimmte Erfahrungswerte über die Brennbarkeit der Abgase bei

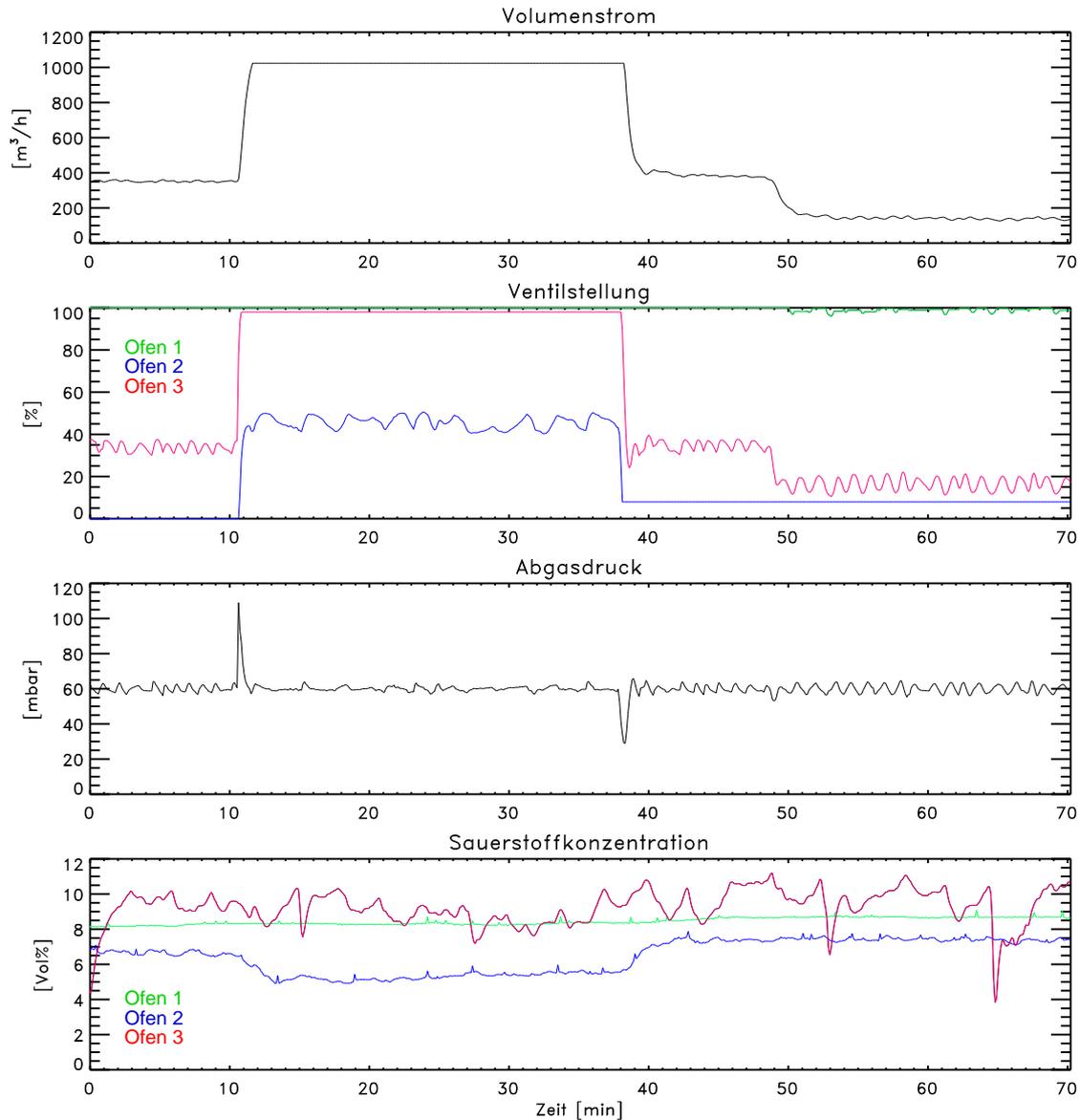


Abbildung 7.4.: **Typisches Verhalten der *split-range*-Regelung.** Dargestellt sind Messungen vom 25. April 1997, die die Größen von Abgas 8 betreffen. Der Volumenstrom steigt innerhalb einer Minute sehr stark an und erreicht mit $1020 \text{ m}^3/\text{h}$ sogar die Grenze des vom PLS darstellbaren Meßbereichs. Um die Grundlast von ca. $350 \text{ m}^3/\text{h}$ zu verbrennen, ist das Ventil von Ofen 1 zu 100 % und das Ventil von Ofen 3 zu etwa 35 % geöffnet. Auf Grund des Druckanstiegs durch den erhöhten Volumenstrom wird durch die Regelung das Ventil von Ofen 3 voll aufgedreht und das Ventil von Ofen 2 auf etwa 45 % geöffnet. Durch dieses drastische Eingreifen des Reglers weicht der Abgasdruck nur kurz vom geforderten Sollwert von 60 mbar ab. Diese hohe Regelgüte wird aber mit einem Abfall der Sauerstoffkonzentration von Ofen 2 auf fast 5 Vol% erkauft. Die Sauerstoffkonzentration des deutlich größeren Ofen 3 scheint aber auf die Mehrbelastung nicht zu reagieren.

den Anlagenfahrern vorhanden. Allerdings kann gelegentlich die Situation auftreten, daß von den Produzenten die Leitungen mit Stickstoff gespült werden. In diesem Fall scheint fälschlicherweise auf Grund des hohen Volumenstroms eine hohe Belastung vorzuliegen, die sich aber nicht im Verlauf von Temperatur und Sauerstoffkonzentration widerspiegelt.

Neben der Aufgabe einen konstanten Druck einzuhalten, besteht für einige Abgasleitungen (Abgas 1,3,5,6 und 11) die Auflage, einen bestimmten Unterdruck auf keinen Fall zu unterschreiten. Da die *split-range*-Regelung nicht den Zustand der SS-Ventile berücksichtigt, kann es durchaus vorkommen, daß ein Stellventil bei einem geschlossenem SS-Ventil zu 100% geöffnet ist. Wenn nun das SS-Ventil geöffnet wird, wird der Unterdruck in der entsprechenden Abgasleitung schlagartig weiter verringert. Dieses Problem tritt häufig auf, wenn ein Ofen nach einem Ausfall wieder hochgefahren wird.

Bevor im nächsten Abschnitt auf die neuartige Regelung der RV eingegangen wird, sollen nochmals die wichtigsten Probleme zusammengefaßt werden:

- unbekannte Brennwerte der Flüssigkeiten
- schlechte Dosierbarkeit der Flüssigkeiten durch nichtlineare Ventilkennlinien
- langsame zeitliche Variation der Eigenschaften einer Flüssigkeit durch Schichtbildung in den Tanks
- Abnahmezwang für die Abgase
- Kopplung zwischen den Öfen durch die *split-range*-Regelung für die Abgase
- Brennwert eines Abgases kann bei der Spülung mit Stickstoff sehr stark variieren
- Die Öffnung eines SS-Ventils kann zu einem Druckabfall in der Abgasleitung führen

7.2. Entwurf einer neuen Regelung

7.2.1. Zielsetzung und Konzeption

Die wesentliche Zielsetzung der Wacker-Chemie GmbH zu Beginn der Kooperation im Frühjahr 1995 war die Erhöhung des Stoffdurchsatzes durch die RV. Damals bestand ein dringender Handlungsbedarf, da die vorhandene Verbrennungskapazität voll ausgeschöpft werden mußte, um die im Werk anfallenden Abfallstoffe verbrennen zu können. Der offensichtlichste Weg den Durchsatz zu erhöhen, schien zunächst die Reduktion der Ofenausfälle zu sein. Zu diesem Zweck war ein System vorgesehen, das auf kritische Momente *lauscht* und im Falle eines drohenden Ofenausfalls eingreift. Schnell zeigte sich aber, daß für ein sinnvolles Eingreifen in die Anlage ein umfassendes Wissen über die Anlagendynamik notwendig ist. Somit verwandelte sich das ursprünglich geplante Notfallsystem immer

mehr zu einer Regelung für den ständigen Betrieb. Diese Regelung hat aber auch eine Reduktion der Ofenausfälle zum Ziel. Sie versucht dies aber nicht durch ein Eingreifen in *letzter Sekunde* zu erreichen, sondern durch die Prävention gefährlicher Situationen.

Grundlage dieser Präventionsstrategie ist eine an die momentane Belastung angepaßte Beheizung der Öfen. Dieses Ziel wird sowohl für die Abgase als auch für die Flüssigkeiten verfolgt. Bei den Abgasen ist hierzu eine dynamische Aufteilung der Abgase auf die Öfen notwendig, wobei – analog zur schon im PLS installierten *split-range*-Regelung – bestimmte Präferenzen vorgegeben werden können. Für die Flüssigkeiten wird ein Regler eingesetzt, der die Kennlinien abschätzt und den gewünschten Massenstrom nach dieser Schätzung einstellt. Der Massenstrom wird aus der aktuellen Belastung des jeweiligen Ofens berechnet. Neben diesen Hauptregelkreisen arbeitet eine Reihe von Notfallreglern, die nur in extremen Situationen in Aktion treten.

Im Gegensatz zu den verfahrenstechnischen Zielen der Wacker-Chemie GmbH lagen für mich zu Beginn der Kooperation andere Motive vor: (i) Zunächst bot die Kooperation eine gute Gelegenheit, einen großen chemischen Produktionsbetrieb aus der Nähe kennenzulernen. (ii) Innerhalb der Kooperation war es möglich, Regelungsverfahren unter Berücksichtigung der technischen Realisierbarkeit zu entwickeln. (iii) Die sehr großen Mengen archivierter Daten boten viele Möglichkeiten zur Datenanalyse und zur Erprobung neuer Verfahren. Letztendlich diente die Anlage der RV dazu, die Realisierbarkeit von Konzepten aus der Neuroinformatik und im Besonderen des Systems der lokalen Experten in einem technisch anspruchsvollen Umfeld zu überprüfen. Die im folgenden beschriebenen Arbeiten für eine neue Regelung der RV haben daher vor allem den Charakter einer Machbarkeitsstudie, die nicht den Anspruch hat *produktreife* Lösungen anzubieten.

Die Komplexität der Anlage und die Vielschichtigkeit der Regelungsaufgaben läßt ahnen, daß eine Lösung des vorliegenden Regelungsproblems ausschließlich mit datenbasierten Neuro-Reglern, wie sie in Abschnitt 5.4 beschrieben wurden, nicht praktikabel ist. Vielmehr ist eine ausgewogene Mischung von alten und neuen Verfahren zweckmäßiger und erfolgversprechender, um zu einer tatsächlichen Verbesserung zu kommen. Da das inzwischen implementierte Gesamtsystem zum einen sehr umfangreich ist und zum anderen in vielen Bereichen auch mit konventionellen Methoden arbeitet, soll hier keine umfassende Darstellung gegeben werden. Für eine vollständige Dokumentation sei daher auf das Konfigurationshandbuch verwiesen [Alb97b]. Der Schwerpunkt der folgenden Abschnitte wird eher auf der Konzeption der Regelung, der Beschreibung der neu entwickelten *software* PCS und der Realisierung eines Reglers für die Flüssigkeiten liegen.

Die zentrale Idee, die hinter dem Konzept des neuen Reglers steht, ist die Verwendung des in Kapitel 4 entwickelten Systems lokaler Experten. Da die Zahl der für die Regelung der RV notwendigen Experten bei etwa 20 liegt, wurden diese nochmals in sechs Blöcke eingeteilt, um den Überblick über das Gesamtsystem zu erleichtern. Abbildung 7.5 zeigt diese Blöcke und die dazugehörigen Experten. Die Aufteilung in Blöcke und Experten spiegelt die stark modulare Struktur des Gesamtsystems wieder. Diese hohe Modularität hat zum einen bei der Entwicklung von Regelungsstrategien und zum anderen bei der Erweiterung

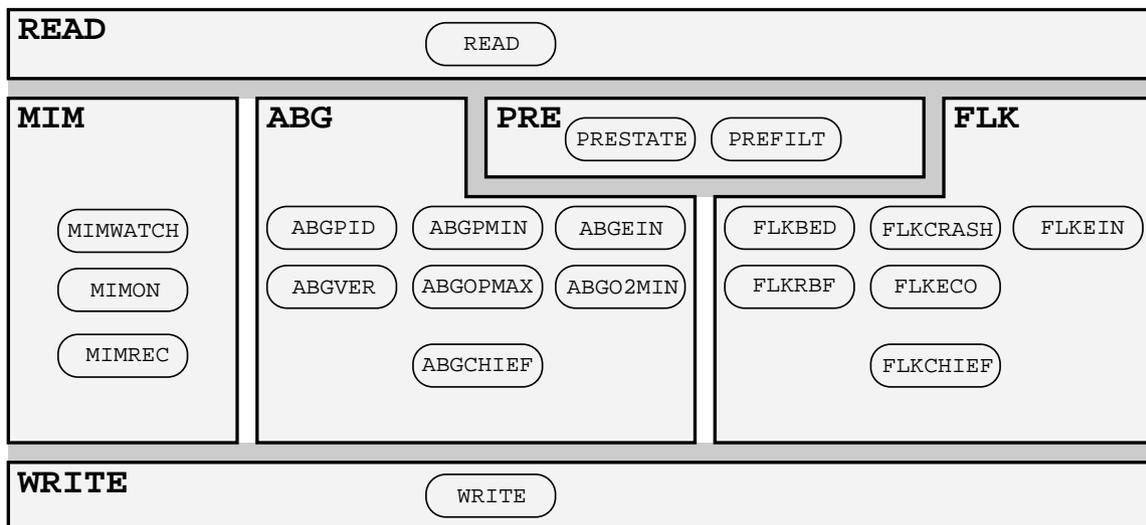


Abbildung 7.5.: **Blockstruktur des mit PCS implementierten Regelungssystems.** Die sechs Blöcke (hellgraue Gebiete) gliedern sich jeweils in eine gewisse Anzahl von Experten (abgerundete Rechtecke). Die dunkelgrauen Bereiche deuten an, daß zwischen den angrenzenden Blöcken Daten ausgetauscht werden.

des Systems um neue Funktionalitäten große Vorteile. Im folgenden werden die Aufgaben der verschiedenen Blöcke und Experten kurz charakterisiert:

- **Block READ und WRITE:** Einlesen von Daten vom PLS und Schreiben von Stellgrößen und binären Signalen an das PLS. Diese beiden Blöcke bestehen jeweils aus einem Experten, der die Kommunikation mit dem PLS abwickelt.
- **Block MIM:** Verschiedene Verwaltungsaufgaben innerhalb des Regelungssystems:
 - Experte MIMON: Mimik zum Einschalten der Regelung. Nur wenn das gesamte System durch einen Bediener am PLS freigeschaltet ist, können Stellgrößen an die Anlage weitergeben werden.
 - Experte MIMWATCH: Überprüfung der korrekten Kopplung zum PLS.
 - Experte MIMREC: Archivierung von Daten, um die Möglichkeit zur späteren Analyse zu haben.
- **Block PRE:** Bearbeitung von Meßgrößen, für die mehrere Messungen vorliegen. Da die Temperatur- und Sauerstoffmessungen aus Gründen der Redundanz jeweils doppelt vorliegen, ist es wünschenswert, jeweils nur einen Meßwert auszuwählen, mit dem die nachgeschalteten Experten arbeiten. Die Auswahl der momentan zuverlässigsten Messung beruht auf einer periodischen Analyse der Kreuzkorrelationsfunktionen einer Messung mit allen anderen Messungen. Wenn eine erwartete Korrelation, z.B. die starke Antikorrelation zwischen Feuerraumtemperatur und Sauerstoffkonzentration, nicht vorliegt, kann daraus auf eine defekte Messung geschlossen

werden. Eine Beschreibung dieses von A. Wiesmüller entwickelten Verfahrens ist in [Alb97b] nachzulesen.

- **Block FLK:** Regelung der Flüssigkeiten, Luftklappen und des Stützbrennstoffs:
 - Experte FLKEIN: Einschaltmimik für die Regelung der Flüssigkeiten.
 - Experte FLKBED, FLKRBF, FLKCRASH, FLKECO: Verschiedene Experten, die zur Flüssigkeitsregelung beitragen.
 - Experte FLKCHIEF: Verknüpfung der diversen Flüssigkeits-Experten.
- **Block ABG:** Regelung der Abgasdrücke auf die Solldrücke p_j^{Ziel} :
 - Experte ABGEIN: Einschaltmimik für die Abgasregelung.
 - Experten ABGPID, ABGVER, ABGPMIN, ABGOPMAX und ABGO2MIN: Verschiedene Experten, die zur Abgasregelung beitragen.
 - Experte ABGCHIEF: Verknüpfung der diversen Abgas-Experten.

Bevor die beiden großen Blöcke ABG und FLK genauer beschrieben werden, soll zunächst auf die Möglichkeiten und die Realisierung des neu entwickelten Regelungsprogramms PCS eingegangen werden.

7.2.2. Programm PCS zur Regelung industrieller Prozesse

PCS ist ein *software*-System zur Realisierung komplexer Regelungs- und Steuerungsaufgaben von großen technischen Anlagen auf konventionellen Microcomputern [Alb97a]. Viele Anlagen der chemischen Industrie sind mit Prozeßleitsystemen zur Steuerung, Regelung und Visualisierung der Produktionsprozesse ausgestattet. Solche Prozeßleitsysteme sind in aller Regel Spezialcomputer, die gleichzeitig sehr viele Aufgaben ausführen, wie z.B. Abfrage von Schnittstellen, Darstellung von diversen Größen auf dem Bildschirm, Interaktion mit dem Benutzer, Anwendung von Regelungsalgorithmen und Steuerlogiken. Wegen der sich daraus ergebenden vielfältigen Anforderungen sind solche Systeme aber meist nur bedingt geeignet, neuartige Algorithmen oder Konzepte zu implementieren und zu erproben. Diese Situation liegt auch mit dem PLS *Contronic P* bei der Rückstandsverbrennung vor.

Um die geplanten Regelungsalgorithmen umsetzen zu können, wurde das *Process Control System* (PCS) als Erweiterung des bestehenden PLS entwickelt. Das PLS ist dabei weiterhin für elementare Funktionen wie Ansteuerung und Abfrage von Aktoren und Sensoren zuständig. Es übernimmt nun aber zusätzlich die Aufgabe den Datentransfer zum PCS zu bedienen.

Die wesentlichen Kriterien, die bei der Entwicklung des PCS eine Rolle spielten, sind :

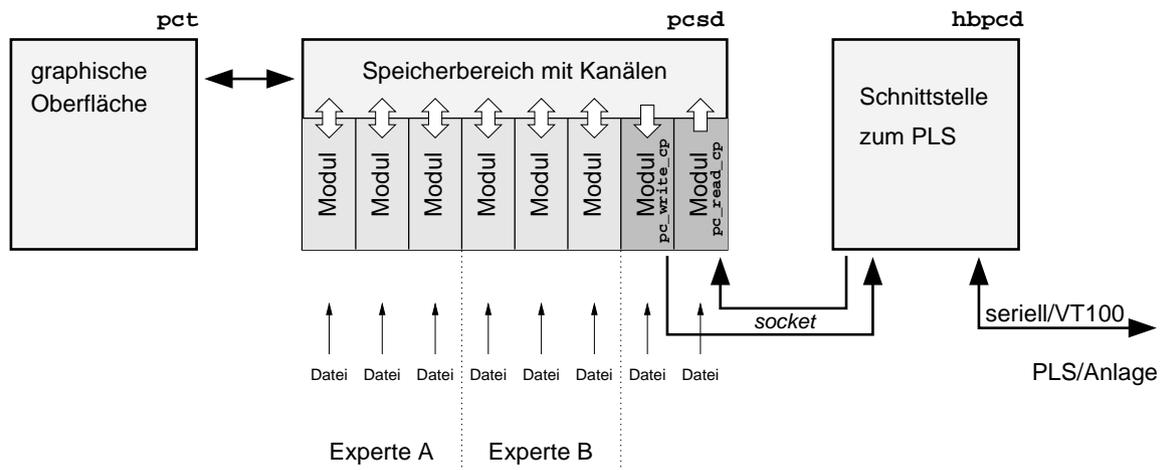


Abbildung 7.6.: **Aufbau des PCS.** Das Gesamtsystem gliedert sich in Kommunikation mit dem PLS (hbpcd), Datenverarbeitung (pcsd) und graphischer Oberfläche zur Visualisierung und Interaktion (pct).

- **Modularität:** Das System gliedert sich in viele unabhängige und selbständige *Module*. Diese werden zentral gestartet und führen anschließend die ihnen zugewiesenen Operationen aus.
- **Erweiterbarkeit:** Bei der Definition der Schnittstelle der Einzelmodule zum restlichen PCS wurde auf größtmögliche programmtechnische Einfachheit geachtet. Es sind nur wenige Programmzeilen erforderlich, um ein Modul in das System einzugliedern. Entsprechend ist es leicht möglich, das System um selbstentwickelte Module zu erweitern.

Abbildung 7.6 zeigt die wichtigsten Elemente des PCS. Die drei im folgenden beschriebenen Programme pcsd, hbpcd und pct laufen vor Ort auf einer *workstation* DEC AXP 255 mit 233 MHz Taktfrequenz und 96 MB Hauptspeicher.

pcsd

Die zentrale Stellung im PCS nimmt das Programm pcsd ein. Dieses startet nacheinander die Module⁵, die anschließend jeweils aus eigenen Dateien ihre spezielle Konfiguration einlesen. Es können also durchaus viele identische Module gleichzeitig im System arbeiten, die aber – durch ihre Konfiguration spezifiziert – verschiedene Aufgaben haben. Wenn die Initialisierung aller Module erfolgreich war, kommt pcsd in die Arbeitsphase. In dieser werden die Module periodisch und in einer vorgegebenen Phasenbeziehung von pcsd

⁵ Jedes Modul arbeitet als sog. *thread* [Her96]. Bei pcsd handelt es sich daher um ein *multi-threading*-Programm. Bei dieser Technik führt ein Programm nicht nur eine Befehlssequenz, sondern viele Sequenzen nebeneinander aus.

aufgefordert, die ihnen zugeordneten Operationen auszuführen. Somit kann für die eingelesenen Daten ein beliebiger Ablauf von Operationen konfiguriert werden. Die Perioden der verschiedenen Module müssen hierbei nicht übereinstimmen, so daß auch asynchrone Operationen, wie z.B. Archivierungsaufgaben oder Adaptionsvorgänge, vorgenommen werden können.

Die Kommunikation zwischen den einzelnen Modulen erfolgt über sog. *Kanäle*. Diese werden während der Startphase von `pcsd` angelegt und sind für alle Module beschreibbare und lesbare Speicherbereiche. Über solche Kanäle wird der Austausch der Meß- und Stellgrößen der Anlage und der intern berechneten Größen abgewickelt. An jeden Kanal angeheftet sind verschiedene Informationen, wie z.B. der Zeitpunkt des letzten Schreibens, Größeneinheit und Kommentare. Zusätzlich ist es möglich, die Inhalte eines Kanals über einen bestimmten Zeitbereich zu puffern. In gewisser Weise stellt das PCS somit eine Echtzeit-Datenbank dar, welche im starken Maße fähig ist, verschiedene eingelesene Daten zu verrechnen und nach außen weiterzugeben.

hbpcd

Für die Kommunikation mit dem PLS sind die beiden `pcsd`-Module `pc_read_cp` und `pc_write_cp` zuständig, die *socket*-Verbindungen⁶ zum Programm `hbpcd` aufbauen. `hbpcd` ist wiederum über eine serielle Schnittstelle mit dem PLS verbunden. Da das PLS nur über eine VT100-Schnittstelle mit einer Bandbreite von 9600 Bit/s verfügt, stellt der Datentransfer zwischen PLS und PCS den Flaschenhals in Bezug auf die erreichbare Antwortzeit des PCS dar. Zusammen mit dem ineffektiven Übertragungsprotokoll *ConLink V*, über welches mit dem PLS kommuniziert werden muß, führt dies dazu, daß für das Lesen und Schreiben von Daten ca. 2.0 s benötigt werden. Werden 0.5 s für die Berechnung eines Regelschritts und 0.5 s als Reserve eingerechnet, dann ergibt sich eine Abtastperiode T_s von 3.0 s.

pct

Die Kontrolle und Manipulation des PCS erfolgt über die graphische Oberfläche *Process Control Tool* (`pct`), die weitestgehend unabhängig von `pcsd` läuft, um die Ausfallsicherheit der eigentlichen Regelung zu erhöhen. Dieses interaktive und frei konfigurierbare Programm aktualisiert *online* die alphanumerische Darstellung der Kanalinhalt. Daneben erlaubt es, manuell Kanäle zu verändern, Informationen zu Kanälen abzurufen und den Zeitverlauf von Kanälen graphisch darzustellen.

⁶Ein *socket* ist eine unter Unix standardisierte Methode, Daten zwischen Programmen auszutauschen, die nicht zwingend auf einem Rechner laufen müssen [Her96].

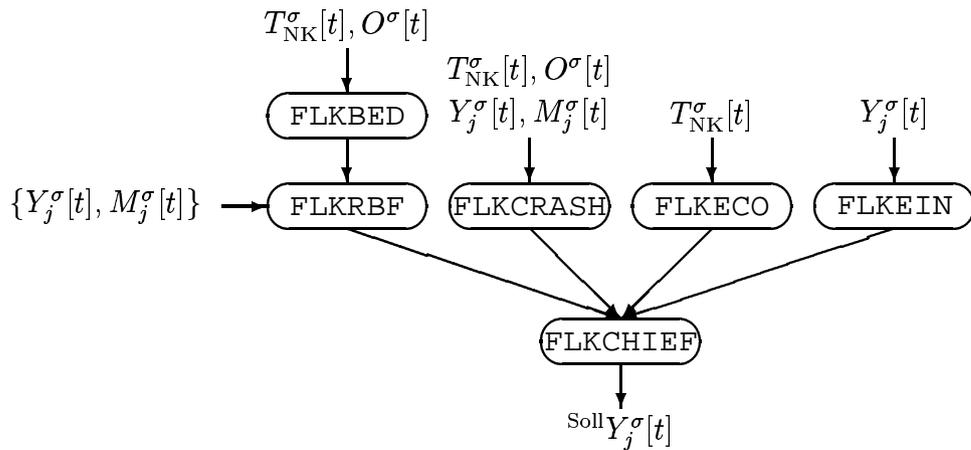


Abbildung 7.7.: Die Experten des Blocks FLK.

7.2.3. Experten zur Regelung der Flüssigkeitszufuhr

Bei der Entwicklung des Blocks FLK stand die Implementierung einer bedarfsgerechten Regelung für die Flüssigkeiten im Vordergrund. Der Aufbau der beiden dafür verantwortlichen Experten FLKBED und FLKVER wird im folgenden Abschnitt noch ausführlich dargestellt. Da die verbleibenden Experten (siehe Abbildung 7.7) jeweils speziellere Ziele verfolgen, soll deren Realisierung hier nur cursorisch skizziert werden:

- **FLKCRASH**: Dieser Experte versucht durch kurzfristige Eingriffe, die mit einer hohen Kompetenz erfolgen, den Ausfall eines Ofens zu vermeiden. Dabei werden drei bedrohliche Situationen berücksichtigt: (i) Bei niedriger, bzw. sehr niedriger und fallender Temperatur wird das Stützgasventil geöffnet. (ii) Wenn auf Grund eines sehr großen Gradienten dM_j^σ/dY_j^σ ein schlagartiger Anstieg des Massenstroms detektiert wird, dann wird das entsprechende Flüssigkeitsventil gedrosselt. (iii) Bei einem starken Abfall der Sauerstoffkonzentration werden alle Flüssigkeitsventile geschlossen. Die obigen Vorschriften sind durch einen Satz von *Fuzzy*-Regeln formuliert.
- **FLKECO**: Dieser Experte hat zum Ziel, den Verbrauch an Stützbrennstoff zu reduzieren. Dies wird dadurch erreicht, daß das entsprechende Ventil sehr langsam zugedreht wird, wenn ein günstiger Ofenzustand vorliegt.
- **FLKEIN**: Wenn die Flüssigkeitsregelung nicht vom Bedienpersonal freigeschaltet ist, gibt dieser Experte die tatsächlich in der Anlage vorliegenden Ventilstellungen an **FLKCHIEF** weiter. Wenn die PCS-Regelung vom Bedienpersonal eingeschaltet wird, dann nimmt die lokale Kompetenz dieses Experten langsam ab. Da die übergeordnete Kompetenz von **FLKEIN** sehr groß ist, wird daher nach Einschalten der Regelung langsam zwischen den Vorgaben des PLS und der Regelung des PCS umgeschaltet. Die Freigabe der Flüssigkeitsregelung bezieht sich immer auf

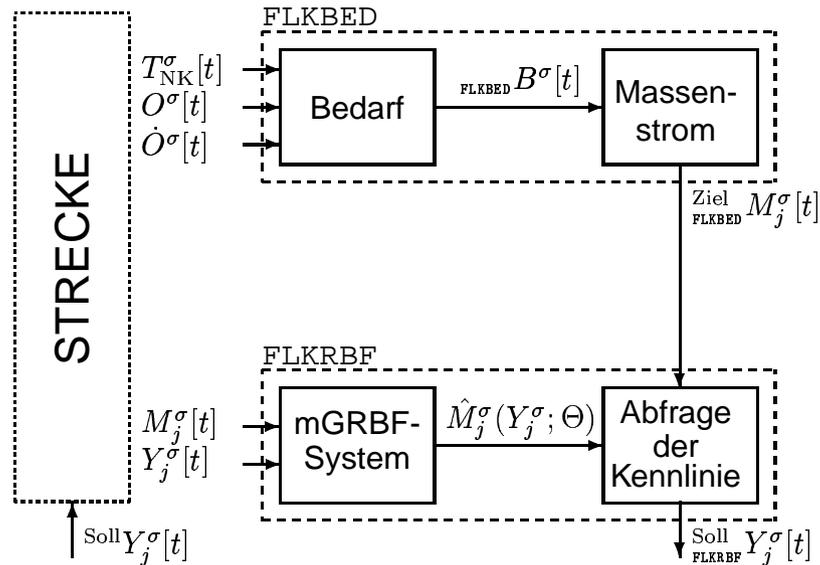


Abbildung 7.8.: **Struktur der Regelung für die Flüssigkeiten.** Die neuen Ventilstellungen $\text{Soll}_{\text{FLKRBF}} Y_j^\sigma[t]$ werden in einem mehrstufigen Prozeß aus dem aktuellen Ofenzustand ($T_{\text{NK}}^\sigma[t]$, $O^\sigma[t]$, $\dot{O}^\sigma[t]$), dem Durchfluß $M_j^\sigma[t]$ und der geschätzten Kennlinie $\hat{M}_j^\sigma(Y_j^\sigma; \Theta)$ gewonnen.

ein bestimmtes Ventil. Üblicherweise wurde nur eine Flüssigkeit pro Ofen durch die PCS-Regelung kontrolliert.

- **FLKCHIEF:** Hier werden die Ausgaben und lokalen Kompetenzen der vorgeschalteten Experten nach (4.1) verknüpft. Die so resultierenden Ventilstellungen werden über den Experten WRITE an die Anlage weitergegeben.

Die oben genannten Experten sind im wesentlichen für spezielle Aufgaben zuständig. Die kontinuierliche und bedarfsgerechte Dosierung einer Flüssigkeit wird von zwei Experten übernommen, die nun – als Beispiel für den hybriden Aufbau der Gesamtregelung – detaillierter beschrieben werden.

7.2.4. Zwei spezielle, bedarfsorientierte Experten

Abbildung 7.8 zeigt das Schema dieses Regelkreises, der sich in zwei große Teile gliedert. Dieser Regler hat nicht – wie der schon im PLS implementierte Regler – zum Ziel, einen manuell vorgegebenen Massenstrom einzuregeln, sondern ermittelt zunächst einen bedarfsgerechten Massenstrom $\text{Ziel}_{\text{FLKBED}} M_j^\sigma[t]$ aus dem Ofenzustand. Anschließend wird eine Ventilstellung $\text{Soll}_{\text{FLKRBF}} Y_j^\sigma[t]$ auf Grundlage des Massenstroms $\text{Ziel}_{\text{FLKBED}} M_j^\sigma[t]$ und einer Schätzung $\hat{M}_j^\sigma(Y_j^\sigma; \Theta^{\text{mGRBF}})$ mit einem mGRBF-System bestimmt.

wenn	$(T_{\text{NK}}^\sigma[t] \in T_{\text{sk}} \vee T_{\text{NK}}^\sigma[t] \in T_{\text{k}}) \wedge (O^\sigma[t] \in O_{\text{g}} \vee O^\sigma[t] \in O_{\text{sg}})$
dann	$B^\sigma[t] = 1.0$ sonst $B^\sigma[t] = 0.0$
wenn	$(T_{\text{NK}}^\sigma[t] \in T_{\text{sk}} \vee T_{\text{NK}}^\sigma[t] \in T_{\text{k}}) \wedge O^\sigma[t] \in O_{\text{m}}$
dann	$B^\sigma[t] = 0.5$ sonst $B^\sigma[t] = 0.0$
wenn	$T_{\text{NK}}^\sigma[t] \in T_{\text{m}} \wedge (O^\sigma[t] \in O_{\text{g}} \vee O^\sigma[t] \in O_{\text{sg}})$
dann	$B^\sigma[t] = 0.5$ sonst $B^\sigma[t] = 0.0$
wenn	$(T_{\text{NK}}^\sigma[t] \in T_{\text{sg}} \vee T_{\text{NK}}^\sigma[t] \in T_{\text{g}}) \wedge O^\sigma[t] \in O_{\text{k}}$
dann	$B^\sigma[t] = -0.5$ sonst $B^\sigma[t] = 0.0$
wenn	$(O^\sigma[t] \in O_{\text{sk}} \vee O^\sigma[t] \in O_{\text{k}} \vee O^\sigma[t] \in O_{\text{m}}) \wedge \dot{O}^\sigma[t] \in \dot{O}_{\text{sf}}$
dann	$B^\sigma[t] = -1.0$ sonst $B^\sigma[t] = 0.0$
wenn	$O^\sigma[t] \in O_{\text{sk}}$
dann	$B^\sigma[t] = -1.0$ sonst $B^\sigma[t] = 0.0$
wenn	$O^\sigma[t] \in O_{\text{m}} \wedge \dot{O}^\sigma[t] \in \dot{O}_{\text{f}}$
dann	$B^\sigma[t] = -0.5$ sonst $B^\sigma[t] = 0.0$
wenn	$O^\sigma[t] \in O_{\text{k}} \wedge \dot{O}^\sigma[t] \in \dot{O}_{\text{f}}$
dann	$B^\sigma[t] = -0.5$ sonst $B^\sigma[t] = 0.0$
wenn	$O^\sigma[t] \in O_{\text{sg}} \wedge (\dot{O}^\sigma[t] \in \dot{O}_{\text{ss}} \vee \dot{O}^\sigma[t] \in \dot{O}_{\text{s}})$
dann	$B^\sigma[t] = 1.0$ sonst $B^\sigma[t] = 0.0$
wenn	$O^\sigma[t] \in O_{\text{g}} \wedge (\dot{O}^\sigma[t] \in \dot{O}_{\text{ss}} \vee \dot{O}^\sigma[t] \in \dot{O}_{\text{s}})$
dann	$B^\sigma[t] = 0.5$ sonst $B^\sigma[t] = 0.0$
wenn	$O^\sigma[t] \in O_{\text{m}} \wedge \dot{O}^\sigma[t] \in \dot{O}_{\text{ss}}$
dann	$B^\sigma[t] = -0.5$ sonst $B^\sigma[t] = 0.0$

Tabelle 7.2.: *Fuzzy* Regeln für die Bestimmung des Flüssigkeitsbedarfs.

Bestimmung der Massenströme M_j^{σ} [t]

Für die Bestimmung eines der Massenströme M_j^{σ} [t] wird vom zugeordneten Experten FLKBED zunächst ein "Bedarf" $B^\sigma[t]$ bestimmt, der den momentanen Zustand eines Ofens σ widerspiegelt. Der Bedarf ist klein, wenn der Ofen durch zusätzliche Abfallstoffe in Gefahr gerät auszufallen und groß, wenn sich der Ofen in einem Zustand befindet, der weitere Verbrennungskapazitäten erlaubt. Eine zweckmäßige Bestimmung des Bedarfs ist über die Aufstellung von *Fuzzy*-Regeln möglich, da so die gewünschten Eigenschaften einfach und intuitiv formuliert werden können. In Zusammenarbeit mit den Anlagenfahrern wurden daher *Fuzzy*-Regeln für einen situationsangepaßten Bedarf $B^\sigma[t]$ formuliert. In Tabelle 7.2 sind diese Regeln zusammengefaßt. Abbildung 7.9 zeigt die dafür notwendigen *membership*-Funktionen. Mit dem dadurch definierten *Fuzzy*-Regelsatz wird aus der Temperatur und der Sauerstoffkonzentration nach der Sugeno-Regel (5.16) ein Bedarf $B^\sigma[t]$ berechnet.

Um die Zufeuerung der Flüssigkeiten nicht zu stark an die stochastischen Schwankungen der Temperatur und der Sauerstoffkonzentration zu koppeln, wird $B^\sigma[t]$ nach Auswer-

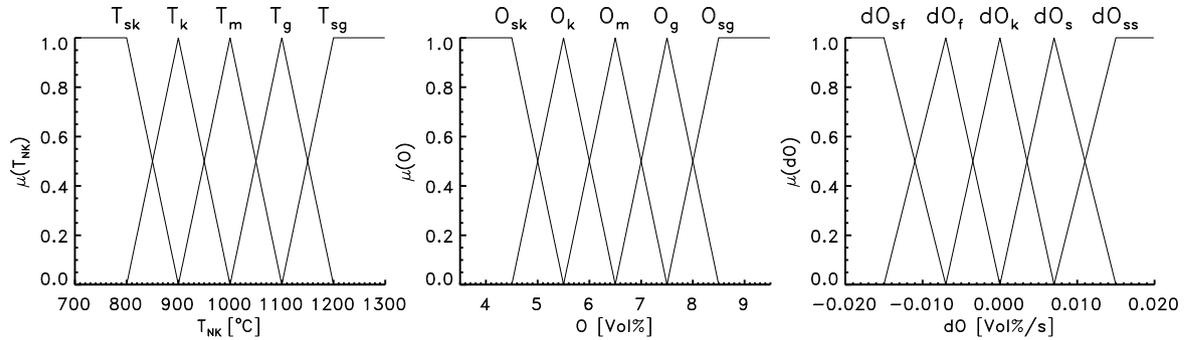


Abbildung 7.9.: Die drei Gruppen von *membership*-Funktionen zur *fuzzy* Berechnung des Bedarfs ${}_{\text{FLKBED}}B^\sigma[t]$.

tion der *Fuzzy*-Regeln mit einer Zeitkonstante von 20 Sekunden zeitlich geglättet.

Der Bedarf ${}_{\text{FLKBED}}B^\sigma[t]$ bietet nun die Möglichkeit einen geeigneten Massenstrom ${}_{\text{FLKBED}}M_j^\sigma[t]$ zu bestimmen. Dies erfolgt durch eine zum Bedarf proportionale Anpassung des Massenstroms in jedem Regelungsschritt durch

$${}_{\text{FLKBED}}M_j^\sigma[t + 1] = {}_{\text{FLKBED}}M_j^\sigma[t] + K({}_{\text{FLKBED}}B^\sigma[t]){}_{\text{FLKBED}}B^\sigma[t] \quad (7.1)$$

$$K({}_{\text{FLKBED}}B^\sigma[t]) = \begin{cases} 9.0 \text{ kg/h} & : \quad {}_{\text{FLKBED}}B^\sigma[t] \leq 0 \\ 3.0 \text{ kg/h} & : \quad {}_{\text{FLKBED}}B^\sigma[t] > 0 \end{cases} \quad (7.2)$$

Die Nichtlinearität der Proportionalitätsgröße $K({}_{\text{FLKBED}}B^\sigma[t])$ führt dazu, daß der Massenstrom langsam erhöht wird, während in kritischen Situationen (${}_{\text{FLKBED}}B^\sigma[t] < 0$) die Flüssigkeitszufuhr dreimal schneller reduziert wird. Bei einem vollständig geschlossenen Ventil benötigt der Regler daher mindestens 200 s, um einen Massenstrom von 200 kg/h einzustellen. Das Schließen dieses Ventils dauert dagegen nur etwa 70 s.

Schätzung der Ventilkennlinie mit einem mGRBF-System

Wie kann nun der Massenstrom ${}_{\text{FLKBED}}M_j^\sigma[t]$ am Ventil eingestellt werden? Oben wurde schon mehrfach erwähnt (vgl. auch Abbildung 7.3), daß einige Flüssigkeitsventile ein überraschendes Verhalten zeigen, welches stark von der zu dosierenden Flüssigkeit abhängt. Die im PLS konfigurierte PID-Regelung muß zwangsläufig für ein solch nichtlineares System versagen. In Kapitel 6 hatten wir aber bei der Regelung eines Bioreaktors, der durch ein Ventil mit einer variablen Kennlinie kontrolliert werden muß, gesehen, daß ein Regler, der auf einer Funktionenapproximation einer multifunktionalen Datenfolge beruht, zu sehr guten Ergebnissen führt. Da bei den Flüssigkeitsventilen der RV eine äquivalente Situation vorliegt, soll dieses Konzept nun auf die Schätzung der Kennlinien der Flüssigkeitsventile übertragen werden. Weil damals (Oktober 1997) der für die Regelung des Bioreaktors

verwendete LLM-Approximator noch nicht zur Verfügung stand, wurde die Funktionenapproximation mit dem im Abschnitt 4.3.1 dargestellten mGRBF-System realisiert.

Die Schätzung der Ventilkennlinien $M_j^\sigma(Y_j^\sigma)$ wird vom Experten FLKRBF mit Hilfe eines mGRBF-System vorgenommen. Dieses System lokaler GRBF-Netzwerke lernt aus dem Trainingsdatensatz $Z_j^\sigma = \{Y_j^\sigma[t], M_j^\sigma[t + \Delta T_{\text{FLK}}] \mid t = 1 \dots N\}$ die Funktionenapproximation

$$\hat{M}_j^\sigma : Y_j^\sigma[t] \rightarrow \hat{M}_j^\sigma[t + \Delta T_{\text{FLK}}]. \quad (7.3)$$

Die Totzeit ΔT_{FLK} beschreibt die Verzögerung auf Grund der Messung und beträgt nach einer Analyse der Kreuzkorrelation zwischen dem Durchfluß und der Ventilstellung ca. 6 s.

Aus Gründen der Übersichtlichkeit werden im folgenden die Indizes für die Öfen und die Ventile weggelassen. Ebenso wird zunächst das Training nur *eines* GRBF-Netzwerks für die Schätzung $\hat{M}(Y; \Theta^{\text{GRBF}})$ einer Ventilkennlinie betrachtet. Entsprechend dem Vorgehen von M. Sonner [Son97] wird das Intervall $[0, 100]$ der möglichen Ventilstellungen in M äquidistante Stützstellen diskretisiert. In Abbildung 7.10, links, ist ein typischer Trainingsdatensatz zusammen mit einer geschätzten Kennlinie $\hat{M}(Y; \Theta^{\text{GRBF}}) = \sum_r f_r a_r(Y; \Theta^{\text{GRBF}})$ dargestellt. Offensichtlich ist die in der linken Abbildung gezeigte Schätzung $\hat{M}(Y; \Theta^{\text{GRBF}})$ wenig sinnvoll, da sie nicht die von einer Ventilkennlinie erwartete Monotonie erfüllt. Für eine vernünftige Schätzung $\hat{M}(Y; \Theta^{\text{GRBF}})$ muß vielmehr $d\hat{M}(Y; \Theta^{\text{GRBF}})/dY \geq 0$ gelten. Außerdem wird verlangt, daß ein geschlossenes Ventil keinen Durchfluß erlaubt, also $\hat{M}(0; \Theta^{\text{GRBF}}) = 0$ gilt. Im folgenden werden die Größen, die eine monotone Kennlinie betreffen, mit einem Querstrich versehen.

Um die beiden oben erwähnten Randbedingungen zu erfüllen, wird eine *Monotonisierung* der Kennlinienschätzung vorgenommen. Hierzu wird die Lernregel (2.64) für die Bestimmung der Ausgabebewichte f_r beim GRBF-Netzwerk durch die Anwendung zweier lokaler Experten erweitert: Da die Kennlinie von links betrachtet monoton steigend, von rechts betrachtet aber monoton fallend sein muß, werden für jedes Zentrum c_r des GRBF-Netzwerks zwei lokale Experten \hat{M}_r^+ und \hat{M}_r^- eingeführt, die die Gewichte f_r so anpassen sollen, daß die jeweilige Eigenschaft erfüllt ist. In Abbildung 7.10, rechts, wird die Arbeitsweise dieser Experten illustriert. Der Experte \hat{M}_r^+ erzwingt die Monotonie, indem er aus den Gewichten f_r neue Gewichte f_r^+ berechnet, für die $f_{r-1}^+ \leq f_r^+$, $r = 2 \dots M$, gilt:

$$f_0^+ = 0 \quad (7.4)$$

$$f_r^+ = \begin{cases} f_r & : f_r > f_{r-1}^+ \\ f_{r-1}^+ & : \text{sonst} \end{cases} \quad r = 2 \dots M. \quad (7.5)$$

Analog geht der Experte \hat{M}_r^- vor, der die Gewichte f_r so modifiziert, daß sie von rechts gesehen immer kleiner werden:

$$f_M^- = f_M \quad (7.6)$$

$$f_r^- = \begin{cases} f_r & : f_r < f_{r+1}^- \\ f_{r+1}^- & : \text{sonst} \end{cases} \quad r = M - 1 \dots 1. \quad (7.7)$$

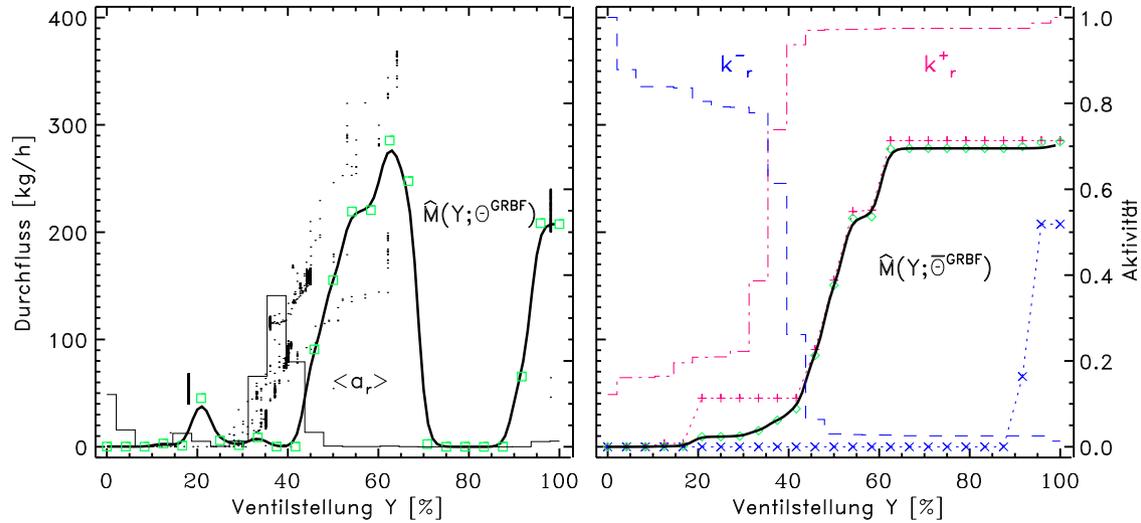


Abbildung 7.10.: **Training einer Ventilkennlinie.** Links sind exemplarisch die Punkte eines Trainingsdatensatzes $\mathcal{Z} = \{Y[t], M[t + \Delta T_{FLK}] \mid t = 1 \dots 17938\}$ eines Flüssigkeitsventils abgebildet. Diese Datenverteilung wurde mit einem GRBF-Netzwerk, welches auf einem äquidistantem Gitter basiert, approximiert. Die Gewichte $f_r, r = 1 \dots 25$, sind als grüne \square und die mittlere Aktivität $\langle a_r \rangle_{\mathcal{Z}}$ als dünne Linie eingezeichnet. Die daraus resultierende Modellfunktion $\hat{M}(Y; \Theta^{GRBF})$ des GRBF-Netzwerks ist als dicke Linie eingetragen. Rechts sind die Ausgaben der Experten \hat{M}_r^+ und \hat{M}_r^- dargestellt. Die roten + (blauen \times) kennzeichnen die Ausgabegewichte f_r^+ (f_r^-) und die gestrichelte rote bzw. blaue Linie zeigt die lokale Kompetenz k_r^+ (k_r^-). Als Ergebnis der beiden Experten erhält man die Gewichte \bar{f}_r (grüne \diamond). Die nun monotone Modellfunktion $\hat{M}(Y; \bar{\Theta}^{GRBF})$ des GRBF-Netzwerks ist wieder als dicke durchgezogene Linie eingetragen.

Diese Gewichte sind in Abbildung 7.10, rechts, als rote + (f_r^+) und blaue \times (f_r^-) eingetragen. Um die Ausgaben f_r^+ und f_r^- der beiden Experten durch einen übergeordneten Experten verknüpfen zu können, ist die Bestimmung lokaler Kompetenzen k_r^+ und k_r^- notwendig. Da die mittlere Aktivität $\langle a_r \rangle_{\mathcal{Z}}$ ein Maß für die Zuverlässigkeit eines Gewichts f_r ist, beschreiben die rekursiv definierten lokalen Kompetenzen

$$k_r^+ = k_{r-1}^+ + \langle a_r \rangle_{\mathcal{Z}} \quad r = 2 \dots M \quad (7.8)$$

$$k_r^- = k_{r+1}^- + \langle a_r \rangle_{\mathcal{Z}} \quad r = M - 1 \dots 1 \quad (7.9)$$

wie repräsentativ die Gewichte f_r^+ und f_r^- sind. Die Verknüpfung von \hat{M}_r^+ und \hat{M}_r^- erfolgt schließlich entsprechend (4.1) durch

$$\bar{f}_r = \frac{f_r^+ k_r^+ + f_r^- k_r^-}{k_r^+ + k_r^-}, \quad r = 1 \dots M, \quad (7.10)$$

wobei hier die übergeordneten Kompetenzen K_r^+ und K_r^- auf Eins gesetzt wurden. Auf diese Weise approximiert das GRBF-Netzwerk die Ventilkennlinie mit der nun monotonen

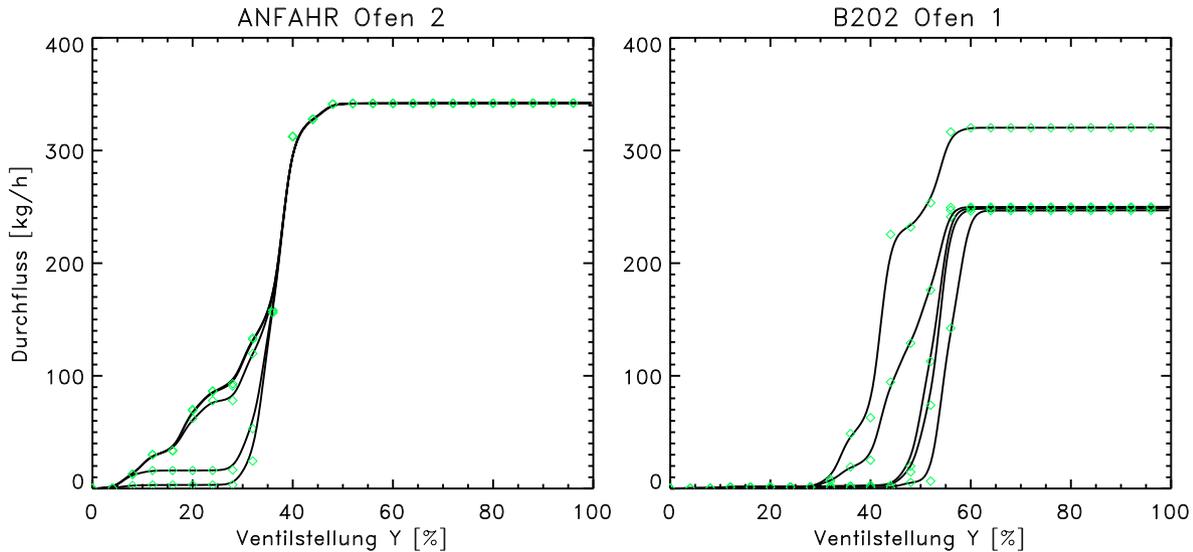


Abbildung 7.11.: **Schätzung der Ventilkennlinien mit einem mGRBF-System.** Links ist die monotonisierte Schätzung $\hat{M}(Y; \bar{\Theta}^{mGRBF})$ der Ventilkennlinie für die Flüssigkeit ANFAHR an Ofen 2 mit einem mGRBF-System aus fünf Modellen dargestellt (durchgezogenen Linien). Die Ausgabegewichte der fünf GRBF-Netzwerke, die zu dieser Schätzung führen, sind als grüne \diamond eingetragen. Rechts ist die entsprechende Schätzung für B202 an Ofen 1 dargestellt.

Schätzung

$$\hat{M}(Y; \bar{\Theta}^{GRBF}) = \sum_{r=1}^M \bar{f}_r a_r(Y; \bar{\Theta}^{GRBF}) \quad (7.11)$$

Abbildung 7.10 illustriert, wie durch eine solche Monotonisierung die oben verlangten Randbedingungen erfüllt werden.

Das gerade dargestellte Vorgehen läßt sich ohne Schwierigkeiten auf die Funktionenapproximation $\hat{M}(Y; \bar{\Theta}^{mGRBF})$ mit einem mGRBF-System übertragen. Für die Schätzung der Ventilkennlinie mit dem Experten FLKRBFB wurde ein mGRBF-System aus 5 Modellen mit jeweils $M = 25$ Stützpunkten verwendet. Abbildung 7.11 illustriert exemplarisch an zwei Flüssigkeitsventilen das Ergebnis der multifunktionalen Funktionenapproximation mit einem mGRBF-System, welches durch die oben geschilderte Monotonisierung modifiziert wurde. Das linke Teilbild zeigt die mit fünf GRBF-Netzwerken geschätzten Kennlinien der erfahrungsgemäß relativ gut dosierbaren Brennerstrecke ANFAHR und das rechte Teilbild die Kennlinien der meist problematischen Brennerstrecke B202. Da die fünf Modelle für ANFAHR sich nicht so stark unterscheiden wie für B202, spiegelt sich das Erfahrungswissen der Anlagenfahrer auch in der Funktionenapproximation durch das mGRBF-System wider.

Bestimmung einer Ventilstellung $\overset{\text{Soll}}{\text{FLKRBF}} Y[t]$ zu einem Massenstrom $\overset{\text{Ziel}}{\text{FLKBED}} M[t]$

Um den in Abbildung 7.8 skizzierten Regelkreis zu schließen, steht noch die Bestimmung einer Ventilstellung $\overset{\text{Soll}}{\text{FLKRBF}} Y[t]$ aus. Diese wird durch den Experten FLKRBF durch eine Nullstellenbestimmung von

$$\hat{M}(\overset{\text{Soll}}{\text{FLKRBF}} Y[t]; \bar{\Theta}^{mGRBF}) - \overset{\text{Ziel}}{\text{FLKBED}} M[t] = 0 \quad (7.12)$$

berechnet. Die lokale Kompetenz $\overset{\text{Soll}}{\text{FLKRBF}} k[t]$ des Experten FLKRBF wird durch die Gesamtaktivität

$$A(\overset{\text{Soll}}{\text{FLKRBF}} Y[t]; \bar{\Theta}^{mGRBF}) = \sum_{l=1}^L \bar{z}_l(\overset{\text{Soll}}{\text{FLKRBF}} Y[t]) A(\overset{\text{Soll}}{\text{FLKRBF}} Y[t]; \bar{\Theta}_l^{GRBF}) \quad (7.13)$$

bestimmt, die durch die Ventilstellung $\overset{\text{Soll}}{\text{FLKRBF}} Y[t]$ auf der Zwischenschicht des mGRBF-Netzwerks ausgelöst wird. Diese lokale Kompetenz wird zusammen mit der Ventilstellung $\overset{\text{Soll}}{\text{FLKRBF}} Y[t]$ an den Experten FLKCHIEF weitergegeben, der diese mit den Eingaben der anderen beteiligten Experten verrechnet und schließlich eine Ventilstellung $\overset{\text{Soll}}{Y}[t]$ an die Anlage weitergibt (vgl. auch Abbildung 7.8).

7.2.5. Experten zur Regelung der Abgaszufuhr

Die Regelung der Abgase ist konzeptionell ähnlich der Flüssigkeitsregelung aufgebaut. Auch hier werden viele – sich zum Teil widersprechende – Zielsetzungen verfolgt, die an verschiedene lokale Experten delegiert werden. Abbildung 7.12 zeigt die beteiligten Experten. Da mit den Experten FLKBED und FLKRBF das Vorgehen beim Reglerentwurf schon exemplarisch demonstriert wurde, wird im folgenden nur kurz die Zielsetzung und die Umsetzung der verschiedenen Experten beschrieben. Die genaue Konfiguration kann in [Alb97b] nachgelesen werden.

- **ABGPID** und **ABGVER**: Diese beiden Experten übernehmen die Hauptarbeit bei der Regelung des Abgasdrucks $\overset{\text{Ziel}}{p}_j[t]$. **ABGPID** ist ein konventioneller PID-Regler, welcher einen Gesamtvolumenstrom $\overset{\text{Ziel}}{v}_j[t]$ als “fiktive” Stellgröße ausgibt. Dieser Volumenstrom, der von den zur Verfügung stehenden Öfen verbrannt werden muß, wird durch den Experten **ABGVER** auf die Öfen aufgeteilt. Dabei wird der Bedarf $\overset{\text{Soll}}{\text{FLKBED}} B^\sigma[t]$ eines Ofens, die Stellung der Schnellschlußventile $s_j^\sigma[t]$ und die durch den Anlagenbetreiber vorgegebene Abnahmereihenfolge berücksichtigt. Hieraus berechnet **ABGVER** einen Volumenstrom $\overset{\text{Soll}}{\text{ABGVER}} v_j^\sigma[t]$, aus dem durch eine Schätzung⁷ der Ventilkennlinien eine Ventilstellung $\overset{\text{Soll}}{\text{ABGVER}} y_j^\sigma[t]$ bestimmt wird.

⁷ Diese Schätzung ist rein erfahrungsbasiert. Auf Grund der gemeinsamen Volumenmessung $v_j^\sigma[t]$ ist eine datenbasierte Schätzung nur schwer möglich.

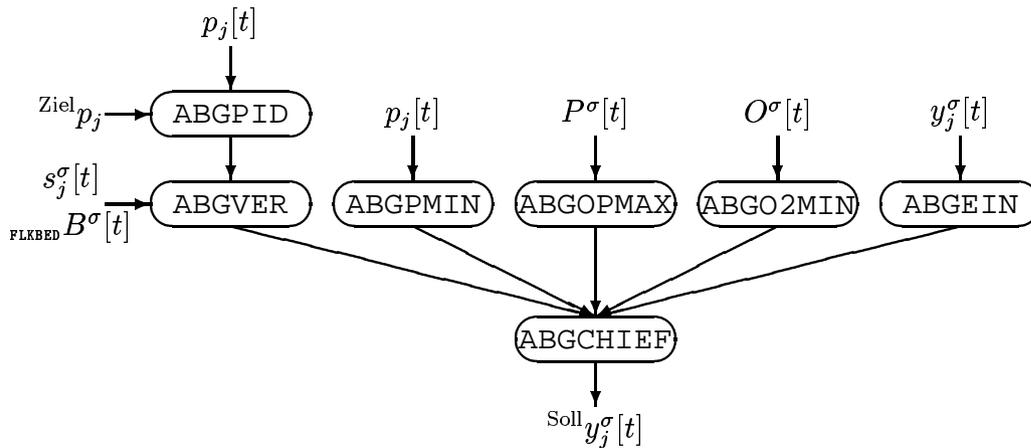


Abbildung 7.12.: Die Experten des Blocks ABG.

- **ABGPMIN**: In einigen Abgasleitungen (1,3,5,6 und 11) darf der Druck nur möglichst selten unterhalb eines vorgegebenen Minimaldrucks fallen, da die Produktionsbetriebe, die diese Abgase erzeugen, auf die Einhaltung eines bestimmten Drucks angewiesen sind. Bei Gefahr eines zu starken Druckabfalls drosselt ABGPMIN daher die entsprechenden Ventile und unterstützt so APBPID und ABGVER.
- **ABGOPMAX**: Dieser Experte tritt nur dann in Aktion, wenn der Ofeninnendruck über den Außendruck anzusteigen droht. In diesem Fall werden alle Abgasventile mit einer hohen Kompetenz geschlossen.
- **ABGO2MIN**: Bei einer kleinen und fallenden Sauerstoffkonzentration kann ein Ofen wegen Unterschreitung des Sauerstoffgrenzwerts ausfallen. Bei Auftreten einer solchen Situation werden die Abgasventile kurzfristig geschlossen und anschließend langsam wieder geöffnet.
- **ABGEIN**: Analog zum Experten FLKEIN sorgt ABGEIN dafür, daß das Umschalten zwischen PLS- und PCS-Regelung langsam erfolgt.

7.3. Beispiele und Diskussion

Zunächst sollen zwei Ausschnitte aus den archivierten Daten betrachtet werden, um die Arbeitsweise des neuen Regelungssystems der RV zu illustrieren.

Abbildung 7.13 zeigt einen Ausschnitt vom 11. November 1997, der die Regelung der Flüssigkeit B202 am Ofen 1 betrifft⁸. Das Regelungssystem wurde zum Zeitpunkt 24 min

⁸Im folgenden werden die Indizes für den Ofen 1 und die Flüssigkeit B202 weggelassen.

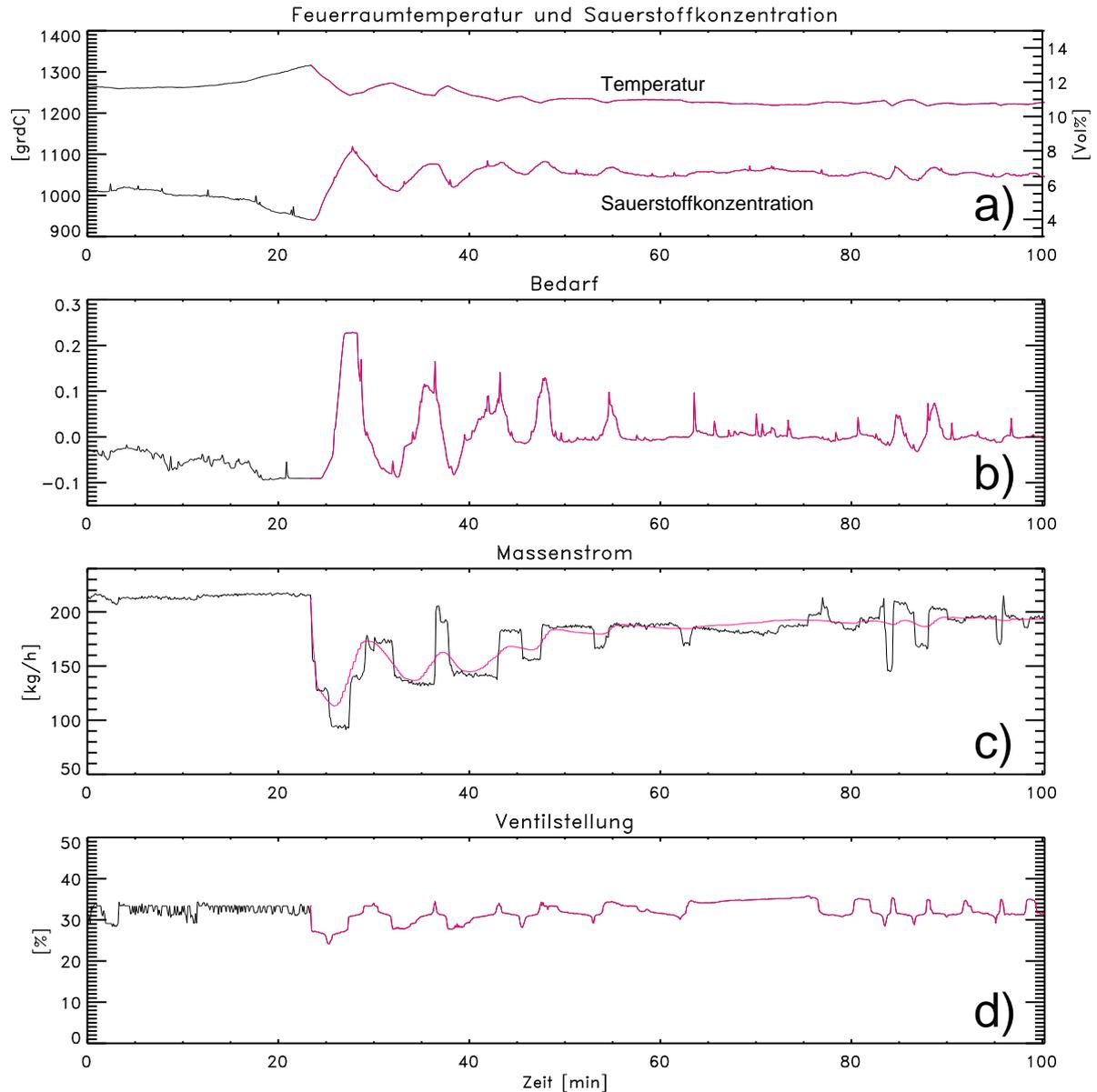
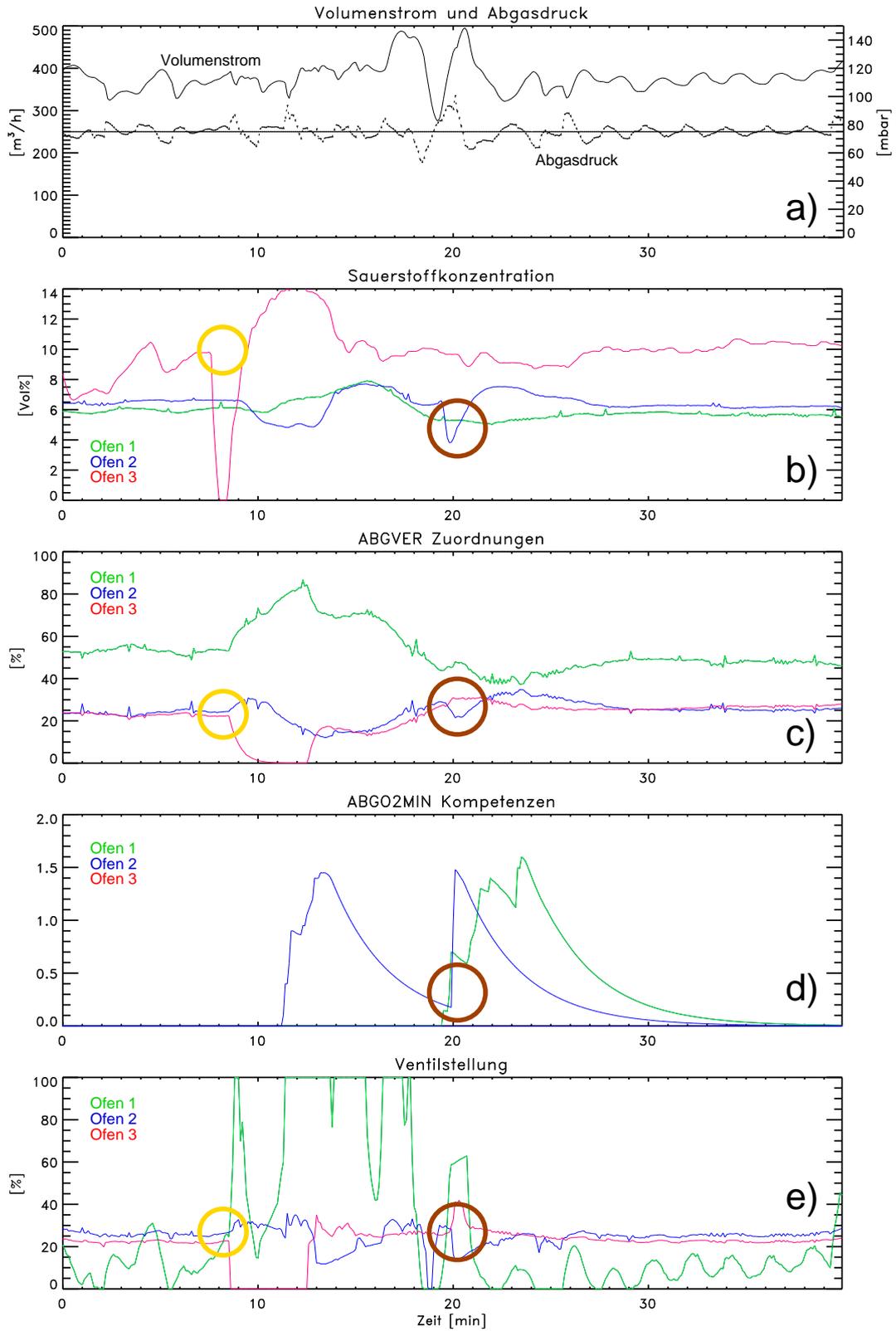


Abbildung 7.13.: **Typisches Verhalten der Flüssigkeitsregelung durch das PCS.** Für eine Diskussion siehe Text.

eingeschaltet. Die roten Kurven markieren den Zeitbereich, in dem das Regelungssystem in die Flüssigkeitendosierung eingreifen konnte. Teilbild a) zeigt oben den Verlauf der Feuerraumtemperatur und unten den dazu antikorrelierten Verlauf der Sauerstoffkonzentration. Man erkennt, wie ca. 10 min vor Einschalten der Regelung ein Abfallen der Sauerstoffkon-

Abbildung 7.14.: **Typisches Verhalten der Abgasregelung durch das PCS.** Für eine Diskussion der Abbildung auf Seite 161 siehe Text.



zentration einsetzt. Der in Teilbild b) dargestellte Bedarf ${}_{\text{FLKBED}}B^\sigma[t]$ ist daher negativ, was bei Einschalten der Regelung zu einer sofortigen Absenkung des Sollmassenstroms ${}_{\text{FLKBED}}^{\text{Ziel}}M^\sigma[t]$ führt. In Teilbild c) ist der Sollmassenstrom ${}_{\text{FLKBED}}^{\text{Ziel}}M^\sigma[t]$ (rot) zusammen mit dem gemessenen Massenstrom M^σ (schwarz) dargestellt. Teilbild d) zeigt schließlich die Ventilstellung ${}_{\text{FLKRBF}}^{\text{Soll}}Y^\sigma[t]$, die aus der Schätzung der Ventilkennlinie mit einem mGRBF-System resultiert. Das dargestellte Beispiel zeigt deutlich, wie die Regelung in die Dosierung eingreift, und es letztendlich auch schafft, einen drohenden Ofenausfall zu vermeiden. Allerdings führt dieser Eingriff auch zu einigen Schwingungen, die erst nach etwa 40 min gedämpft sind.

In Abbildung 7.14 wird die Arbeitsweise der neuen Abgasregelung am Beispiel von Abgas 6 für einen Ausschnitt vom 1. Oktober 1997 illustriert. Das oberste Teilbild zeigt den aktuellen Volumenstrom $v_6[t]$ und Druck $p_6[t]$ des Abgases zusammen mit dem Sollwert ${}_{\text{Ziel}}p_6$. In den Teilbildern b)–e) ist das Verhalten von Meßgrößen, bzw. lokalen Experten dargestellt. Dabei werden die drei Öfen jeweils farblich kodiert. In dem gewählten Datenausschnitt sollen zwei Situationen genauer betrachtet werden:

- Der gelb markierte Bereich kennzeichnet einen Ausfall von Ofen 3. Dies wird durch den verfahrenstechnisch bedingten Abfall der Sauerstoffkonzentration auf Null ersichtlich. Durch den Ausfall verändern sich die vom Experten ABGVER berechneten Zuordnungen, durch welche die Abgasabnahme der einzelnen Öfen bestimmt wird. Im Teilbild e) ist schließlich zu erkennen, wie die Regelung das Abgasventil von Ofen 1 öffnet und so den Abgasdruck $p_6[t]$ trotz Ausfall von Ofen 3 nahezu konstant halten kann. Zusätzlich schließt die Regelung das Abgasventil von Ofen 3, um ein Abfallen des Abgasdrucks durch das schlagartige Öffnen des Schnellschlußventils zu vermeiden.
- Der braun markierte Bereich kennzeichnet ein schnelles Abfallen der Sauerstoffkonzentration von Ofen 2, das vermutlich in einer starken Schwankung eines Flüssigkeitsstroms begründet ist. Experte ABGVER reagiert darauf, indem er langsam die Zuordnung von Ofen 2 verringert. Wesentlich schneller agiert der Experte ABGO2MIN, der immer einen Ventilwert von Null ausgibt. ABGVER erreicht durch eine kräftige Anhebung seiner lokalen Kompetenz, daß die Abgasventilstellung (unterstes Teilbild) angedrosselt wird. Durch das Eingreifen von ABGO2MIN werden alle Abgasventile von Ofen 2 angedrosselt, wodurch nach einigen Minuten die Sauerstoffkonzentration wieder ansteigt.

Die objektive Untersuchung der Leistungsfähigkeit der beiden neuen Regelungen für die Flüssigkeiten und Abgase gestaltete sich schwierig, da ein quantifizierbares Maß für die Qualität der Ofenregelung nur mit einem großen Fehler zu ermitteln ist. Die Ausführungen von Abschnitt 7.1.4 haben deutlich gemacht, daß nicht die Einhaltung eines bestimmten Sollwerts – wie bei einem einfachen SISO-Reglerkreis – sondern die Berücksichtigung einer Reihe von Randbedingungen, die zu einer Erhöhung der Ausfallsicherheit und des Stoffdurchsatzes führen sollen, von Bedeutung ist. Mögliche Qualitätsmaße wären daher

Meßwerterfassung: 163.3 Tage								
PCS-Abgasregelung in Betrieb: 36.0 %								
Ofen 1 in Betrieb: 91.7 %			Ofen 2 in Betrieb: 94.3 %			Ofen 3 in Betrieb: 85.3 %		
	ABG _{an}	ABG _{aus}		ABG _{an}	ABG _{aus}		ABG _{an}	ABG _{aus}
R [1/d]	2.69	2.70	R [1/d]	3.82	3.55	R [1/d]	3.35	3.36
$\Delta R/R$ [%]	8.0	6.0	$\Delta R/R$ [%]	6.7	5.2	$\Delta R/R$ [%]	7.1	5.3

Tabelle 7.3.: **Ausfallraten der Öfen.** Ein Ofenausfall liegt vor, wenn alle Schnellschlußventile eines Ofens geschlossen werden. Die Tabelle vergleicht die Ausfallraten R (in Ereignissen pro Tag), für den Fall, daß die PCS- und die PLS-Regelung der Abgase in Betrieb war. Der statistische Fehler $\Delta R/R$ berechnet sich aus der absoluten Anzahl N der Ausfälle, also $\Delta R/R = 1/\sqrt{N}$.

die mittlere Ausfallrate und der mittlere Stoffdurchsatz eines Ofens. Da die Regelung mit Block FLK nur über kurze Zeiträume (einige Tage) für *eine* Zuleitung in Betrieb war, ist es aber nicht zuverlässig möglich, den mittleren Massenstrom mit und ohne PCS-Regelung zu vergleichen. Ebenso ist ein Vergleich des mittleren Abgasvolumenstroms nicht sinnvoll, da für die Abgase ein Abnahmezwang besteht und Schwankungen daher ausschließlich auf die Produktionsbetriebe zurückzuführen sind. Eine weitere Möglichkeit besteht in der Gegenüberstellung der mittleren Ausfallraten für die PLS- und die PCS-Regelung. Tabelle 7.3 zeigt diese Ausfallraten zusammen mit den statistischen Fehlern. Offensichtlich sind hier aber keine signifikanten Unterschiede festzustellen.

Die generelle Schwierigkeit einer Beurteilung der implementierten PCS-Regelung ist in drei Punkten begründet:

- Der alltägliche Betriebsablauf der RV ist von sehr vielen externen, nicht quantifizierbaren Einflüssen abhängig. Die angelieferten Abgase, Flüssigkeiten und Feststoffe spiegeln die Produktionsverhältnisse innerhalb des Werks wider, die wiederum starken saisonalen Schwankungen unterworfen sind. Daher ist es unmöglich, bestimmte Betriebssituationen zu reproduzieren, um eine vergleichende Evaluation verschiedener Regelungen zu erarbeiten.
- Die Anlage befindet sich ständig im Umbau. So wurden während der Projektlaufzeit sukzessive alle Flüssigkeitsventile ausgetauscht, eine große Pufferungsmöglichkeit für das durchsatzstarke Abgas 10 eingerichtet und ein Katalysator für die Gesamtanlage errichtet. Die Öfen fallen daher immer wieder wegen Wartungs- und Umbauarbeiten tage- bis wochenlang aus. Diese betriebsbedingten Ausfälle können in den Daten nicht eindeutig von sicherheitsbedingten Ausfällen unterschieden werden.
- Auch die Konfiguration der PCS-Regelung wurde ständig verändert, um Schwächen und Schwierigkeiten auszumerzen. Vor allem für den Block FLK wurden auf Grund der schwer beherrschbaren Flüssigkeitsventile verschiedene Regelungskonzepte er-

probt. Die oben dargestellt Konfiguration spiegelt daher nur den Zustand der PCS-Regelung zum Abschluß der Projekts wider.

Man muß also letztendlich festhalten, daß eine objektive Aussage darüber, ob das neue Regelungssystem für die RV zu den von der Wacker-Chemie GmbH anvisierten Zielen geführt hat, nur schwer möglich ist. Es sei aber angemerkt, daß etwa von der Hälfte der Anlagenfahrer das neue Regelungssystem aktiv eingeschaltet wurde⁹. Diese subjektive Einschätzung der Anlagenfahrer scheint zumindest auf eine gewisse Stabilisierung der RV durch die neue Regelung hinzuweisen.

Das wichtigste Ergebnis aus der Kooperation mit der Wacker-Chemie GmbH ist, daß es gelungen ist mit einem relativ geringen Gesamteinsatz an Personal (ca. 2 Mannjahre) ein funktionsfähiges und alltagstaugliches Regelungssystem für eine industrielle Anlage zu entwickeln. Einen wesentlichen Anteil daran hat das dem Regelungssystem zugrunde liegende Konzept der lokalen Experten. Nur so war es möglich, neue Algorithmen schnell in das Gesamtsystem zu integrieren und zu erproben.

⁹Dieses Verhalten kann gut in den archivierten Daten kurz nach dem Schichtwechsel beobachtet werden.

A. Anhang

A.1. Der LLM-Approximator

A.1.1. Herleitung des LLM-Approximators

In diesem Anhang sollen die Integrale aus (3.34) ausgewertet werden. Wenn die Mischung (3.30) in (3.34) eingesetzt und die Integration mit der Summation vertauscht wird, erhält man

$$\hat{\mathbf{f}}(\mathbf{x}; \Theta^{LLM}) = \frac{\sum_{r=1}^M \mathbf{I}_r(\mathbf{x})}{\sum_{r'=1}^M J_{r'}(\mathbf{x})} \quad (\text{A.1})$$

mit

$$\mathbf{I}_r(\mathbf{x}) \equiv \int_{\mathbb{R}^K} \mathbf{y} \hat{p}(\mathbf{x}, \mathbf{y} | r, \Theta_r^{PCA}) d\mathbf{y} \quad (\text{A.2})$$

und

$$J_r(\mathbf{x}) \equiv \int_{\mathbb{R}^K} \hat{p}(\mathbf{x}, \mathbf{y} | r, \Theta_r^{PCA}) d\mathbf{y} \quad . \quad (\text{A.3})$$

Zunächst soll das Integral (A.2) berechnet werden. Dazu definieren wir

$$\mathbf{b}_r \equiv \begin{pmatrix} \mathbf{b}_{r,x} \\ \mathbf{b}_{r,y} \end{pmatrix} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} - \begin{pmatrix} \mathbf{c}_r \\ \mathbf{f}_r \end{pmatrix}. \quad (\text{A.4})$$

Ferner stellen wir fest, daß die inverse Kovarianzmatrix Σ_r^{-1} , wie in (3.37)-(3.40) angegeben, in Blockmatrizen zerlegt werden kann. Damit kann das Argument $\mathbf{b}_r^T \Sigma_r^{-1} \mathbf{b}_r$ der Exponentialfunktion in (3.31) als

$$\mathbf{b}_r^T \Sigma_r^{-1} \mathbf{b}_r = \mathbf{b}_{r,x}^T \boldsymbol{\chi}_r^{-1} \mathbf{b}_{r,x} + 2\mathbf{b}_{r,x}^T \boldsymbol{\Omega}_r^{-1} \mathbf{b}_{r,y} + \mathbf{b}_{r,y}^T \boldsymbol{\Phi}_r^{-1} \mathbf{b}_{r,y} \quad (\text{A.5})$$

geschrieben werden. Nun lässt sich die Exponentialfunktion faktorisieren und man erhält mit der Substitution $\mathbf{y} = \mathbf{f}_r + \mathbf{b}_{r,y}$ für (A.2)

$$\mathbf{I}_r(\mathbf{x}) = \frac{1}{N_r} \exp \left[-\frac{1}{2} \mathbf{b}_{r,x}^T \boldsymbol{\chi}_r^{-1} \mathbf{b}_{r,x} \right] \mathbf{K}_r(\mathbf{b}_{r,x}) \quad (\text{A.6})$$

mit

$$\mathbf{K}_r(\mathbf{b}_{r,x}) \equiv \int (\mathbf{f}_r + \mathbf{b}_{r,y}) \exp \left[-\mathbf{b}_{r,x}^T \boldsymbol{\Omega}_r^{-1} \mathbf{b}_{r,y} - \frac{1}{2} \mathbf{b}_{r,y}^T \boldsymbol{\Phi}_r^{-1} \mathbf{b}_{r,y} \right] d\mathbf{b}_{r,y}. \quad (\text{A.7})$$

Der Normierungsfaktor in (A.6) ist dabei durch

$$N_r \equiv (2\pi)^{D/2} (\det \boldsymbol{\Sigma}_r)^{1/2} \quad (\text{A.8})$$

gegeben. Das verbleibende Integral (A.7) wird durch eine quadratische Erweiterung des Arguments der Exponentialfunktion ausgewertet. Mit der Definition

$$\mathbf{v}_r \equiv \mathbf{b}_{r,y} + \boldsymbol{\Phi}_r (\boldsymbol{\Omega}_r^{-1})^T \mathbf{b}_{r,x} \quad (\text{A.9})$$

kann dieses Argument in zwei Summanden zerlegt werden, wobei nur einer der Summanden von \mathbf{v}_r abhängig ist:

$$-\mathbf{b}_{r,x}^T \boldsymbol{\Omega}_r^{-1} \mathbf{b}_{r,y} - \frac{1}{2} \mathbf{b}_{r,y}^T \boldsymbol{\Phi}_r^{-1} \mathbf{b}_{r,y} = -\frac{1}{2} \mathbf{v}_r^T \boldsymbol{\Phi}_r^{-1} \mathbf{v}_r + \frac{1}{2} \mathbf{b}_{r,x}^T \boldsymbol{\Omega}_r^{-1} \boldsymbol{\Phi}_r (\boldsymbol{\Omega}_r^{-1})^T \mathbf{b}_{r,x}. \quad (\text{A.10})$$

Die Existenz der zu $\boldsymbol{\Phi}_r$ inversen Matrix $\boldsymbol{\Phi}_r^{-1}$ aus (A.9) wird weiter unten noch bewiesen. Da (A.9) nur eine von \mathbf{y} unabhängige Verschiebung des Vektors $\mathbf{b}_{r,y}$ bewirkt, gilt $d\mathbf{b}_{r,y} = d\mathbf{v}_r$. So erhält man für $\mathbf{K}_r(\mathbf{b}_{r,x})$ aus (A.7) zunächst

$$\begin{aligned} \mathbf{K}_r(\mathbf{b}_{r,x}) &= \quad (\text{A.11}) \\ &\exp \left[\frac{1}{2} \mathbf{b}_{r,x}^T \boldsymbol{\Omega}_r^{-1} \boldsymbol{\Phi}_r (\boldsymbol{\Omega}_r^{-1})^T \mathbf{b}_{r,x} \right] \int (\mathbf{f}_r + \mathbf{v}_r - \boldsymbol{\Phi}_r (\boldsymbol{\Omega}_r^{-1})^T \mathbf{b}_{r,x}) \exp \left[-\frac{1}{2} \mathbf{v}_r^T \boldsymbol{\Phi}_r^{-1} \mathbf{v}_r \right] d\mathbf{v}_r \end{aligned}$$

und durch Integration

$$\mathbf{K}_r(\mathbf{b}_{r,x}) = N_{r,y} \exp \left[\frac{1}{2} \mathbf{b}_{r,x}^T \boldsymbol{\Omega}_r^{-1} \boldsymbol{\Phi}_r (\boldsymbol{\Omega}_r^{-1})^T \mathbf{b}_{r,x} \right] (\mathbf{f}_r - \boldsymbol{\Phi}_r (\boldsymbol{\Omega}_r^{-1})^T \mathbf{b}_{r,x}) \quad (\text{A.12})$$

mit dem Normierungsfaktor

$$N_{r,y} \equiv (2\pi)^{K/2} (\det \boldsymbol{\Phi}_r^{-1})^{-1/2} \quad (\text{A.13})$$

Werden nun die Hilfsgrößen \mathbf{b}_r und \mathbf{v}_r resubstituiert bekommt man für (A.2) den Ausdruck

$$\begin{aligned} \mathbf{I}_r(\mathbf{x}) &= \frac{N_{r,y}}{N_r} \exp \left[-\frac{1}{2} (\mathbf{x} - \mathbf{c}_r)^T \tilde{\boldsymbol{\chi}}_r^{-1} (\mathbf{x} - \mathbf{c}_r) \right] (\mathbf{f}_r - \boldsymbol{\Phi}_r (\boldsymbol{\Omega}_r^{-1})^T (\mathbf{x} - \mathbf{c}_r)) \\ &= \frac{N_{r,x} N_{r,y}}{N_r} (\mathbf{f}_r + \tilde{\boldsymbol{\Phi}}_r (\mathbf{x} - \mathbf{c}_r)) \tilde{p}(\mathbf{x} | r, \Theta_r^{PCA}). \end{aligned} \quad (\text{A.14})$$

Die Matrix

$$\tilde{\Phi}_r \equiv -\Phi_r(\Omega_r^{-1})^T \quad (\text{A.15})$$

projiziert dabei den Differenzvektor $\mathbf{x} - \mathbf{c}_r \in \mathbb{R}^D$ in den Ausgaberaum \mathbb{R}^K . Die Matrix

$$\tilde{\chi}_r^{-1} \equiv \chi_r^{-1} - \Omega_r^{-1}\Phi_r(\Omega_r^{-1})^T \quad (\text{A.16})$$

ist nach (3.38) und (3.39) symmetrisch und definiert eine neue Metrik im Unterraum \mathbb{R}^D , die von der lokalen Verteilung der Daten im $\mathbb{R}^K \times \mathbb{R}^D$ abhängig ist. Mittels dieser neuen Metrik wird eine neue lokale Normalverteilung

$$\tilde{p}(\mathbf{x}|r, \Theta_r^{LLM}) \equiv \frac{1}{N_{r,x}} \exp \left[-\frac{1}{2}(\mathbf{x} - \mathbf{c}_r)^T \tilde{\chi}_r^{-1}(\mathbf{x} - \mathbf{c}_r) \right] \quad (\text{A.17})$$

auf dem Unterraum \mathbb{R}^D definiert, deren Parametersatz durch $\Theta_r^{LLM} = \{\mathbf{c}_r, \tilde{\chi}_r^{-1} | r = 1 \dots M\}$ gegeben ist und vermöge (A.16) aus Θ_r^{PCA} zu bestimmen ist. Dazu ist eine Inversion der durch (3.39) gegebenen Matrix Φ_r^{-1} nötig. Für den Normierungsfaktor gilt wie üblich

$$N_{r,x} \equiv (2\pi)^{D/2} (\det \tilde{\chi}_r^{-1})^{-1/2}. \quad (\text{A.18})$$

In (A.1) steht noch die Berechnung des Integrals $J_r(\mathbf{x})$ aus. Mit der zum obigen Vorgehen analogen Rechnung erhält man für dieses Integral

$$J_r(\mathbf{x}) = \frac{N_{r,x}N_{r,y}}{N_r} \tilde{p}(\mathbf{x}|r, \Theta_r^{LLM}), \quad (\text{A.19})$$

so daß abschließend lediglich die Berechnung der Konstanten $N_{r,x}N_{r,y}/N_r$ verbleibt. Mit den Definitionen (A.8), (A.13) und (A.18) berechnen sich diese zu

$$\left(\frac{N_{r,x}N_{r,y}}{N_r} \right)^2 = \frac{\det(\Sigma_r^{-1})}{\det(\Phi_r^{-1}) \det(\chi_r^{-1} - \Omega_r^{-1}\Phi_r(\Omega_r^{-1})^T)}. \quad (\text{A.20})$$

Aus dem Determinantenentwicklungssatz folgt für die Determinante von Matrizen in Blockform der Ausdruck [Ait69, Fis86]

$$\det \Sigma_r^{-1} = \det \begin{pmatrix} \chi_r^{-1} & \Omega_r^{-1} \\ (\Omega_r^{-1})^T & \Phi_r^{-1} \end{pmatrix} = \det(\Phi_r^{-1}) \det(\chi_r^{-1} - \Omega_r^{-1}\Phi_r(\Omega_r^{-1})^T) \quad (\text{A.21})$$

und somit, wie zu erwarten war,

$$\frac{N_{r,x}N_{r,y}}{N_r} = 1. \quad (\text{A.22})$$

Da $\det(\Sigma_r^{-1}) = \det(\hat{\Sigma}_r^{-1}) = \prod_r \sigma_{ir}^{-2} \neq 0$, folgt aus (A.21), daß immer $\det(\Phi_r^{-1}) \neq 0$ gilt. Damit ist Φ_r^{-1} regulär und die Existenz der inversen Matrix Φ_r gesichert.

Als Ergebnis für die Funktionenapproximation (A.1) erhält man schließlich aus (A.14) und (A.19) und mit den Definitionen (3.42) und (3.43) die Gleichung (3.41).

A.1.2. Klassifikation mit einem LLM-Approximator

Für die Berechnung des Ansatzes (3.2) müssen die Integrale (A.2) und (A.3) ausgewertet werden. Dazu wollen wir zunächst für den gegebenen Fall des Klassifikationsproblems die Eigenschaften der zugehörigen Verbundwahrscheinlichkeitsdichte $p(\mathbf{x}, \mathbf{y})$ der Daten \mathcal{Z} genauer untersuchen. Eine solche Dichte kann immer als eine Summe

$$p(\mathbf{x}, \mathbf{y}) = \sum_{\lambda=1}^K P_\lambda p(\mathbf{x}, \mathbf{y}|\lambda) \quad (\text{A.23})$$

aus den K Produkten der *a priori*-Wahrscheinlichkeiten P_λ und der klassenlokalen Dichten $p(\mathbf{x}, \mathbf{y}|\lambda)$ geschrieben werden. Diese können jeweils gemäß

$$p(\mathbf{x}, \mathbf{y}|\lambda) = p(\mathbf{y}|\lambda)p(\mathbf{x}|\mathbf{y}, \lambda) \quad (\text{A.24})$$

weiter zerlegt werden. Auf Grund unserer speziellen Kodierung der Klassenzugehörigkeit sind alle Datenpunkte \mathbf{x} einer Klasse λ gerade durch $\mathbf{y} = \mathbf{e}_\lambda$ markiert. Folglich gilt $p(\mathbf{y}|\lambda) = \delta(\mathbf{y} - \mathbf{e}_\lambda)$ und $p(\mathbf{x}|\mathbf{y}, \lambda) = p(\mathbf{x}|\lambda)$. Wir können daher die klassenlokalen Verbunddichten aus (A.24) auch als

$$p(\mathbf{x}, \mathbf{y}|\lambda) = \delta(\mathbf{y} - \mathbf{e}_\lambda)p(\mathbf{x}|\lambda) \quad (\text{A.25})$$

schreiben. Diese Dichten haben also in \mathbf{y} -Richtung verschwindende Varianzen und sind im \mathbf{y} -Unterraum um die Punkte \mathbf{e}_λ zentriert.

Die Komponenten $\hat{p}(\mathbf{x}, \mathbf{y}|r, \Theta_r^{PCA})$ einer ML-Schätzung der Verbunddichte $p(\mathbf{x}, \mathbf{y})$ müssen nun ebenfalls diese Eigenschaften haben. Dies ist gerade der Fall für die Zerlegung

$$\hat{p}(\mathbf{x}, \mathbf{y}|r, \Theta_r^{PCA}) = \delta(\mathbf{y} - \mathbf{f}_r)\hat{p}(\mathbf{x}|\mathbf{y}, r, \Theta_r^{PCA}) \quad (\text{A.26})$$

mit

$$\mathbf{f}_r \in \{\mathbf{e}_\lambda | \lambda = 1 \dots K\}, \quad (\text{A.27})$$

so daß jede multivariate Normalverteilung r eindeutig einer Klasse λ zugeordnet ist. Mit der Eigenschaft (A.26) der in die ML-Mischungsdichte $\hat{p}(\mathbf{x}, \mathbf{y}|\Theta^M)$ eingehenden Komponenten $\hat{p}(\mathbf{x}, \mathbf{y}|r, \Theta_r^{PCA})$ können die Integrale (A.2) und (A.3) leicht berechnet werden und man erhält für den LLM-Approximator

$$\hat{f}(\mathbf{x}; \Theta^{LLM}) = \frac{\sum_{r=1}^M \mathbf{f}_r \hat{p}(\mathbf{x}|\mathbf{f}_r, r, \Theta_r^{PCA})}{\sum_{r'=1}^M \hat{p}(\mathbf{x}|\mathbf{f}_{r'}, r', \Theta_{r'}^{PCA})}. \quad (\text{A.28})$$

Wegen (A.27) können alle Normalverteilungen $r = 1 \dots M$ mittels der Zerlegung

$$\{r | r = 1 \dots M\} = \bigcup_{\lambda=1}^K \{r_\lambda | r_\lambda = 1 \dots M_\lambda\} \quad (\text{A.29})$$

auf die K Klassen aufgeteilt werden, wobei für die Mächtigkeiten M_λ

$$\sum_{\lambda}^K M_\lambda = M \quad (\text{A.30})$$

gilt. Auf Grund der Eigenschaft der *load balance*, welche der *multivar*-Algorithmus garantiert, teilen sich die Zentren der Normalverteilungen entsprechend der *a priori*-Wahrscheinlichkeiten P_λ der Klassen auf die K disjunkten Untermengen $\{r_\lambda\}$ auf [Der95]. Für große M gilt daher

$$\frac{M_\lambda}{M} \approx P_\lambda. \quad (\text{A.31})$$

Mit (A.27) und (A.29)–(A.31) können nun die Summen in (A.28) in Teilsommen zerlegt werden, die jeweils nur eine Klasse λ betreffen. Man erhält somit (3.58).

A.2. Dynamik des Bioreaktors

A.2.1. Fixpunkte des Bioreaktors

Die Gleichungen (6.5) und (6.6) definieren für einen konstanten Durchfluß u an jedem Zustandspunkt $\mathbf{C} \equiv (C_1, C_2)$ eine Richtung, die angibt wie sich der Zustand des Bioreaktors in einer infinitesimal kleinen Zeitspanne entwickeln wird. Diese Richtungen sind Tangentialvektoren an die Trajektorien des Systems. Werden diese Tangentialvektoren als kleine Pfeile in den Zustandsraum, welcher durch C_1 und C_2 aufgespannt wird, eingetragen, erhält man ein Abbild des Phasenraums. Abbildung A.1 zeigt Phasenräume, die sich für vier verschiedene Durchflüsse u ergeben. Der Phasenraum läßt sich in Gebiete unterteilen, in denen $dC_1/dt > 0$ oder $dC_1/dt < 0$ bzw. $dC_2/dt > 0$ oder $dC_2/dt < 0$ gilt. Diese Gebiete sind durch Kurven getrennt, auf denen die Konzentrationen von C_1 bzw. C_2 konstant bleiben. Für die Gleichungen (6.5) und (6.6) ist dies – trivialerweise – für

$$f_{\dot{C}_1=0}(C_1, C_2) : C_1 = 0 \quad (\text{A.32})$$

erfüllt. Als nichttriviale Lösung ergibt sich

$$g_{\dot{C}_1=0}(C_1, C_2) : u = \mu(C_2) \quad (\text{A.33})$$

$$g_{\dot{C}_2=0}(C_1, C_2) : C_1 = C_2 \frac{1 + \beta - C_2}{1 + \beta}. \quad (\text{A.34})$$

Offensichtlich ist $g_{\dot{C}_1=0}(C_1, C_2)$ lediglich numerisch lösbar, da für einen gegebenen Durchfluß u die Konzentration C_2 in (A.33) nur implizit gegeben ist. Die Kurve $g_{\dot{C}_2=0}(C_1, C_2)$ ist dagegen explizit gegeben. Ein Blick auf Abbildung 6.2 zeigt, daß $u = \mu(C_2)$ für $u < 1$ eine

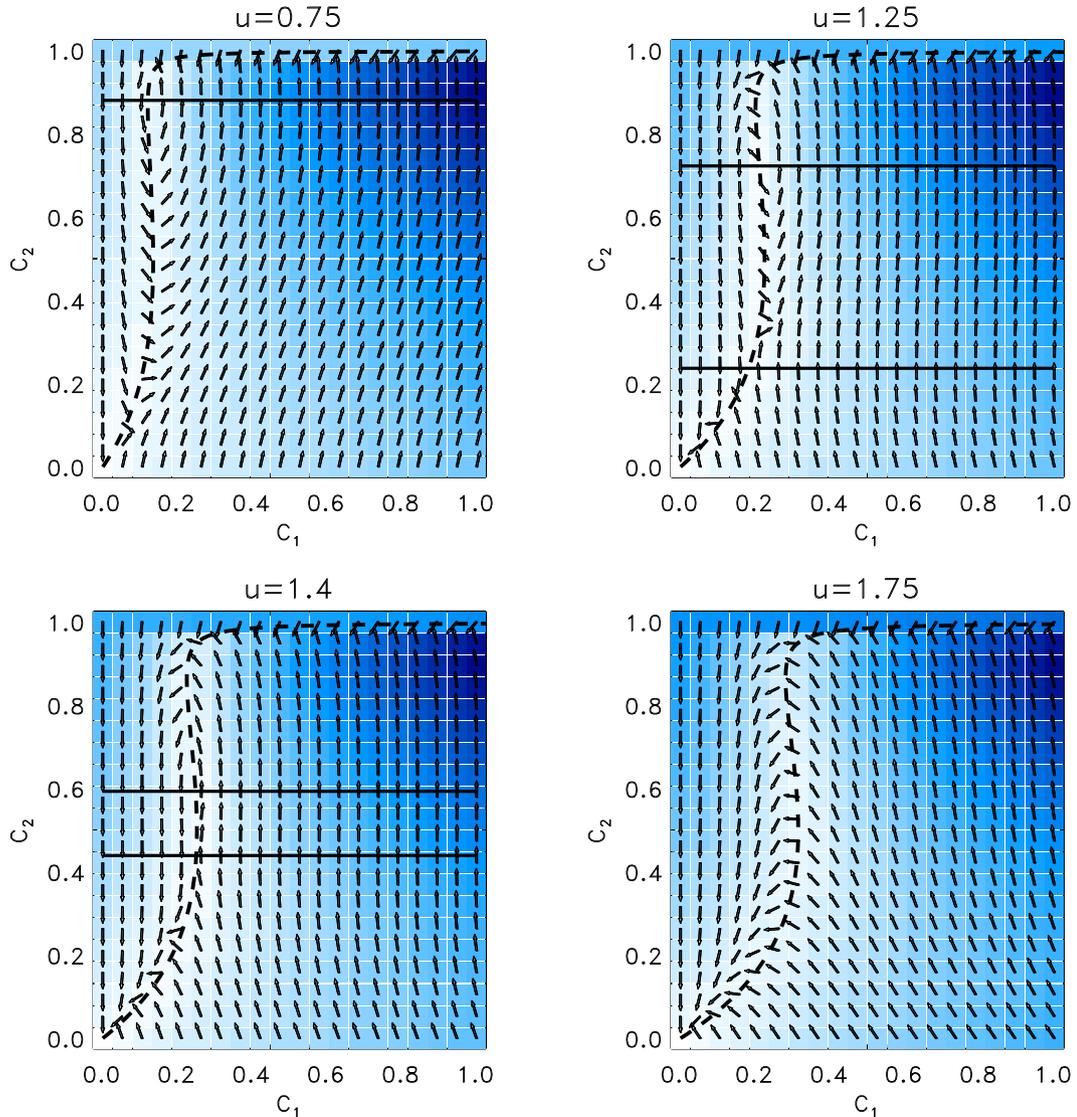


Abbildung A.1: **Phasenraum des Bioreaktors.** Die Teilbilder zeigen durch Pfeile der Steigung dC_2/dC_1 an, in welche Richtung sich die Lösungen des Differentialgleichungssystems (6.5) und (6.6) für konstante Durchflüsse $u = 0.75, 1.25, 1.4$ und 1.75 entwickeln. Der farbige Hintergrund kodiert den Betrag $|(dC_1/dt, dC_2/dt)|$ (weiß symbolisiert einen kleinen Betrag und blau einen großen). Für die eingezeichneten Kurven $g_{\dot{C}_1=0}(C_1, C_2)$ (durchgezogen) bzw. $g_{\dot{C}_2=0}(C_1, C_2)$ (gepunktet) gilt $dC_1/dt = 0$ bzw. $dC_2/dt = 0$. Die Kurven trennen also Gebiete, in denen eine Richtungskomponente dC_1/dt bzw. dC_2/dt das Vorzeichen wechselt.

Lösung, für $1 \leq u \leq \mu^{\max}$ genau zwei Lösungen und für $u > \mu^{\max}$ keine Lösung hat. Die Gerade $g_{\dot{C}_1=0}(C_1, C_2)$, die sich durch die numerische Lösung der Gleichung $u = \mu(C_2)$ nach C_2 ergibt und die Kurve $g_{\dot{C}_2=0}(C_1, C_2)$, sind in Abbildung A.1 eingetragen.

	$\text{Im}(\lambda_i) = 0$	$\text{Im}(\lambda_i) \neq 0$
$\text{Re}(\lambda_i) < 0$	stabiler Knoten 	stabiler Focus 
$\text{Re}(\lambda_i) > 0$	instabiler Knoten 	instabiler Focus 
$\text{Re}(\lambda_i) = 0$		Zyklus 
$\text{Re}(\lambda_1) > 0, \text{Re}(\lambda_2) < 0$	Sattelpunkt 	

Tabelle A.1.: **Stabilität von Fixpunkten.** Fixpunkte werden nach dem Vorzeichen des Real- und Imaginäranteile der Eigenwerte λ_i der Jacobimatrix klassifiziert, da diese das Verhalten der Näherungslösung (A.38) bestimmen.

An den Schnittpunkten $\mathbf{C}^S = (C_1^S, C_2^S)$ der in Abbildung A.1 eingetragenen Kurven gilt gleichzeitig $dC_1/dt = 0$ und $dC_2/dt = 0$. Dies sind die sog. *Fixpunkte* des Differentialgleichungssystems, an denen die Dynamik des Bioreaktors stationär wird. Man erkennt, daß es drei verschiedene Arten von Fixpunkten gibt:

- Triviale Lösung $\mathbf{C}^{S,0} = (0, 0)$. Dieser auch als *washout* bezeichnete Zustand charakterisiert eine Situation, bei der trotz maximaler Nährstoffkonzentration ($C_2(t) = 0$) alle Zellen aus dem Tank gespült wurden ($C_1(t) = 0$) [ALLR82].
- Fixpunkt $\mathbf{C}^{S,1}$, der nur für $u \leq \mu^{\max}$ existiert. In Abbildung A.1 ist dies der obere Schnittpunkt.
- Fixpunkt $\mathbf{C}^{S,2}$, der nur für $1 \leq u \leq \mu^{\max}$ existiert. In Abbildung A.1 ist dies der untere Schnittpunkt.

Abbildung 6.3 zeigt die Lage der Fixpunkte in Abhängigkeit vom Durchfluß u . In dieser Abbildung ist zusätzlich auch das Verhalten der Reaktordynamik in der Nähe der Fixpunkte eingetragen. Die dafür notwendige Stabilitätsanalyse wird im nächsten Abschnitt vorgenommen.

A.2.2. Stabilitätsanalyse der Fixpunkte

Die Stabilitätsanalyse [Hak82] beruht auf der Untersuchung einer genäherten Lösung in der Umgebung eines Fixpunkts \mathbf{C}^S . Dazu wird das Differentialgleichungssystem (6.5) und (6.6) um einen Fixpunkt \mathbf{C}^S bis zur 1. Ordnung entwickelt:

$$f_1(\mathbf{C}) \approx f_1(\mathbf{C}^S) + \left. \frac{\partial f_1}{\partial C_1} \right|_{\mathbf{C}^S} (C_1 - C_1^S) + \left. \frac{\partial f_1}{\partial C_2} \right|_{\mathbf{C}^S} (C_2 - C_2^S) \quad (\text{A.35})$$

$$f_2(\mathbf{C}) \approx f_2(\mathbf{C}^S) + \left. \frac{\partial f_2}{\partial C_1} \right|_{\mathbf{C}^S} (C_1 - C_1^S) + \left. \frac{\partial f_2}{\partial C_2} \right|_{\mathbf{C}^S} (C_2 - C_2^S). \quad (\text{A.36})$$

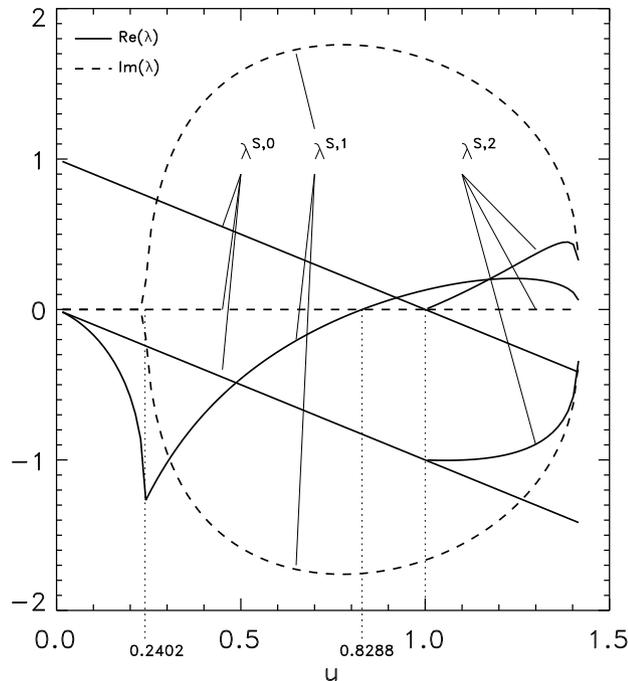


Abbildung A.2.: **Eigenwerte in Abhängigkeit vom Durchfluß u .** $\lambda^{S,0}$, $\lambda^{S,1}$ und $\lambda^{S,2}$ sind die Eigenwerte der Fixpunkte $C^{S,0}$, $C^{S,1}$ und $C^{S,2}$. Die Abbildung zeigt die Real- und Imaginärteile dieser Eigenwerte in Abhängigkeit vom Durchfluß u . Beispielsweise sind die beiden Eigenwerte $\lambda^{S,0}$ immer reell und für $u \geq 1$ negativ. Daher handelt es sich um den Fixpunkt $C^{S,0}$ für $u < 1$ um einen Sattelpunkt und sonst um einen stabilen Knoten (vgl. auch Tabelle A.1).

In vektorieller Notation kann dies einfacher als

$$\frac{d}{dt} \mathbf{C} = \mathbf{J}(\mathbf{C}^S) \cdot (\mathbf{C}(t) - \mathbf{C}^S) \quad (\text{A.37})$$

geschrieben werden, wobei $\mathbf{J}(\mathbf{C}^S)$ die Jacobimatrix des Differentialgleichungssystems (6.5) und (6.6) ist. Mit dem Lösungsansatz

$$\begin{aligned} \mathbf{C}(t) &= \mathbf{C}^S + \mathbf{C}_0 e^{\lambda t} \\ &= \mathbf{C}^S + \mathbf{C}_0 e^{\text{Re}(\lambda)t} e^{i\text{Im}(\lambda)t} \end{aligned} \quad (\text{A.38})$$

erhält man schließlich das Eigenwertproblem

$$\lambda \mathbf{C}_0 = \mathbf{J}(\mathbf{C}^S) \cdot \mathbf{C}_0. \quad (\text{A.39})$$

Die Analyse der beiden Eigenwerte λ_1 und λ_2 von (A.39) ermöglicht eine Beurteilung der Stabilität eines Fixpunkts \mathbf{C}^S . Aus (A.38) ist zu ersehen, daß ein negativer Realteil der beiden Eigenwerte dämpfend auf die Lösung wirkt und somit zu einem stabilen Fixpunkt, ein

positiver Realteil dagegen zu einem instabilen Fixpunkt führt. Ein von Null verschiedener Imaginärteil liefert noch zusätzliche periodische Anteile. Tabelle A.1 faßt die verschiedenen Eigenschaften von Fixpunkten zusammen [Hak82]. Zusätzlich ist in dieser Tabelle das dynamische Verhalten in der Umgebung eines Fixpunkts skizziert.

Wird der Formalismus der Stabilitätsanalyse auf die Gleichungen des Bioreaktors angewendet, so erhält man für die Jacobimatrix den Ausdruck

$$\mathbf{J}(\mathbf{C}^S) = \begin{pmatrix} 0 & C_1^S \frac{d\mu(C_2^S)}{dC_2} \\ \sigma(C_2^S) & -u + C_1^S \frac{d\sigma(C_2^S)}{dC_2} \end{pmatrix}. \quad (\text{A.40})$$

Eine weitere analytische Analyse von (A.40) gestaltet sich aber als schwierig, da die Größe C_2^S über (A.34) nur implizit gegeben ist. Daher werden die Eigenwerte $\lambda^{S,0}$, $\lambda^{S,1}$ und $\lambda^{S,2}$ der drei verschiedenen Fixpunkte von (A.40) numerisch in Abhängigkeit von u bestimmt. Das Ergebnis dieser Berechnung zeigt Abbildung A.2, in der die verschiedenen Eigenwerte in Abhängigkeit von u dargestellt sind. Mit Hilfe der Tabelle A.1 kann aus den Verläufen dieser Kurven die Stabilität der Fixpunkte des Bioreaktors abgelesen werden. Diese Eigenschaften der verschiedenen Fixpunkte sind nochmals in Tabelle 6.1 aufgeführt und in Abbildung 6.3 zusammen mit der Lage der Fixpunkte im Phasenraum dargestellt.

Literaturverzeichnis

- [ABK⁺99] S. Albrecht, J. Busch, M. Kloppenburg, F. Metze und P.Tavan. Generalized radial basis function networks for classification and novelty detection: Self-organization of optimal Bayesian decision. *Neural Networks*, 1999. submitted.
- [Ado92] U. Adolphs. Selbstorganisierende Perzeptronstrukturen mit radialen Basisfunktionen. Diplomarbeit, Physik-Department, TU München, April 1992.
- [Ait69] A.C. Aitken. *Determinanten und Matrizen*. Bibliographisches Institut, 1969.
- [AJ96] E. Alpaydin und M. Jordan. Local linear perceptrons for classification. *IEEE Transactions on Neural Networks*, 7(3):788–792, 1996.
- [Alb71] J.S. Albus. A theory of cerebellar function. *Mathematical Biosciences*, 10:25–61, 1971.
- [Alb95] S. Albrecht. Ein neuronaler Klassifikator für Phoneme. Diplomarbeit, Inst. für Med. Optik, LMU München, 1995.
- [Alb97a] S. Albrecht. *Process Control System 1.4, Handbuch*. Inst. für Med. Optik, LMU München, 1997.
- [Alb97b] S. Albrecht. *Regelung der Rückstandsverbrennung mit dem Process Control System 1.4*. Inst. für Med. Optik, LMU München, 1997.
- [ALLR82] P. Agrawal, C. Lee, H. Lim und D. Ramkrishna. Theoretical investigations of dynamic behavior of isothermal continuous stirred tank biological reactors. *Chemical Engineering Science*, 37(3):453–462, 1982.
- [And97] J.A. Anderson. *An Introduction to Neural Networks*. MIT Press, Cambridge, MA, 1997.
- [AW73] K.J. Astrom und B. Wittenmark. On self-tuning regulators. *Automatica*, 9:185–199, 73.
- [Bäc96] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.

- [Bar90] G. Barto. Connectionist learning for control: An overview. In W. Miller, R.S. Sutton und P.J. Werbos, Herausgeber, *Neural Networks for Control*. MIT Press, Cambridge, MA, 1990.
- [BF95] S. Billings und C. Fung. Recurrent radial basis function networks for adaptive noise cancellation. *Neural Networks*, 8(2):273–290, 1995.
- [BFG95] M. Bianchini, P. Frasconi und M. Gori. Learning without local minima in radial basis function networks. *IEEE Transactions on Neural Networks*, 6:749–756, 1995.
- [Bir99] F. Birkhahn. Prozeßleittechnik im Spannungsfeld neuer Standards und Technologien. *Automatisierungstechnische Praxis*, 41(1):17–26, 1999.
- [Bis95] C.M. Bishop. *Neural networks for pattern recognition*. Clarendon Press, Oxford, 1995.
- [BS91] I.N. Bronstein und K.A. Semendjajew. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Thun und Frankfurt/Main, 1991.
- [BS93] T. Bäck und H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.
- [Bus97] J. Busch. Neuronale Netze als approximative Bayessche Klassifikatoren. Diplomarbeit, Inst. für Med. Optik, LMU München, 1997.
- [CCG91] S. Chen, F. Cowan und P. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2:302–309, 1991.
- [Cri89] F. Crick. The recent excitement about neural networks. *Nature*, 337:129–132, 1989.
- [Cyb89] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, (2):303–314, 1989.
- [Czi89] H. Czichos. *Hütte*. Springer, Berlin, 1989.
- [Der95] D. Dersch. *Eigenschaften neuronaler Vektorquantisierer und ihre Anwendung in der Sprachverarbeitung*. Verlag Harri Deutsch, 1995.
- [DeS95] C.W. DeSilva. *Intelligent control*. CRC Press, 1995.
- [DH73] R. O. Duda und P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.
- [DL89] F. Dörrscheidt und W. Latzel. *Grundlagen der Regelungstechnik*. B.G. Teubner Stuttgart, 1989.

-
- [DLR77] A. P. Dempster, N. M. Laird und D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. Ser. B*, 39:1–38, 1977.
- [DLT95] R. Der, T. List und P. Tavan. Verbundprojekt REMÜVÖ: Regelung von Chemiemüll-Verbrennungsöfen, 1995. BMBF-Antrag.
- [DP80] D. Dubois und H. Prade. *Fuzzy Sets and Systems*. Academic Press, 1980.
- [DT94a] D. R. Dersch und P. Tavan. Control of annealing in minimal free energy vector quantization. In *Proceedings of the IEEE International Conference on Neural Networks*, Seiten 698–703, 1994.
- [DT94b] D. R. Dersch und P. Tavan. Load balanced vector quantization. In *Proceedings of the International Conference on Artificial Neural Networks*, Seiten 1067–1070. Springer, Berlin, 1994.
- [Ecc73] J.C. Eccles. The cerebellum as computer. *Journal of Physiology (London)*, 229, 1973.
- [EIS67] J.C. Eccles, M. Ito und J. Szentágothai. *The cerebellum as a neural machine*. Springer, Berlin, 1967.
- [EKS94] A. Engel, P. König und W. Singer. *Gehirn und Nervensystem*, Kapitel Bildung repräsentationaler Zustände im Gehirn, Seiten 42–48. Spektrum der Wissenschaft Verlagsgesellschaft, 1994.
- [Fis86] G. Fischer. *Lineare Algebra*. Vieweg, 1986.
- [Föll92] O. Föllinger. *Regelungstechnik*. Hüthig, 1992.
- [Fun89] K.-I. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192, 1989.
- [Gar97] C.W. Gardiner. *Handbook of stochastic methods for physics, chemistry and the natural sciences*. Springer, Berlin, 1997.
- [Gee94] H.P. Geering. *Regelungstechnik*. Springer, Berlin, 1994.
- [GJ79] M.R. Garey und D.S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. 1979.
- [Gol23] C. Golgi. *Opera omnia*. Mailand, 1923.
- [Gra95] A. Grauel. *Fuzzy-Logik*. BI Wissenschaftsverlag, Mannheim, 1995.
- [Gro74] S. Grossberg. *Progress in Theoretical Biology*, 51(3), 1974.

- [Gro95] V. Groll. Selektion und Gewichtung selbstorganisierender Merkmalsdetektoren zur Phonemerkennung. Diplomarbeit, Inst. für Med. Optik, LMU München, 1995.
- [GS84] G.C. Goodwin und K.S. Sin. *Adaptive filtering prediction and control*. Prentice-Hall, 1984.
- [Hak82] H. Haken. *Synergetik*. Springer, Berlin, 1982.
- [Hay92] I. Hayashi. Construction of fuzzy inference rules by neural network driven fuzzy reasoning with learning functions. *International Journal of Approximate Reasoning*, 6:241, 1992.
- [Heb49] D. Hebb. *Organisation of Behavior*. Wiley, New York, 1949.
- [Her96] H. Herold. *UNIX-Systemprogrammierung*. Addison-Weseley, Bonn, 1996.
- [HH52] A.L. Hodgkin und A.F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology (London)*, (117):500–544, 1952.
- [Hin94] G. Hinton. *Gehirn und Bewußsein*, Kapitel Wie neuronale Netze aus Erfahrung lernen. Spektrum Akademischer Verlag, 1994.
- [HK92] K. Hornik und C.-M. Kuan. Convergence analysis of local feature extraction algorithms. *Neural Networks*, 5:229–240, 1992.
- [HKRT94] C. Hillermeier, N. Kunstmann, B. Rabus und P. Tavan. Topological feature maps with self-organized lateral connections: A population-coded one-layer model of associative memory. *Biological Cybernetics*, 72:103–117, 1994.
- [Hop82] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79:2554–2558, 5 1982.
- [Hor91] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257, 1991.
- [HSW89] K. Hornik, M. Stinchcombe und H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.
- [Hub90] D.H. Hubel. *Auge und Gehirn, Neurobiologie des Sehens*. Spektrum der Wissenschaft, 1990.
- [JJ92] M. Jordan und R. Jacobs. Hierarchies of adaptive experts. In J. Moody, S. Hanson und R. Lippmann, Herausgeber, *Advances in Neural Information Processing Systems*, Nummer 4, Seiten 985–993. Morgan Kaufmann, 1992.

-
- [JJ93a] R. Jacobs und M. Jordan. Learning piecewise control strategies in a modular neural network architecture. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(2), 1993.
- [JJ93b] M. Jordan und R. Jacobs. Hierarchical mixtures of experts and the EM algorithm. A.I. Memo 1440, Massachusetts Institute of Technology, <ftp://publicatios.ai.mit.edu>, 1993.
- [JJ94] M. Jordan und R. Jacobs. Hierarchical mixtures of experts and the EM algorithm. 6:181–214, 1994.
- [JJNH91] R. Jacobs, M. Jordan, S.J. Nowlan und G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.
- [Jou92] C.O. Jou. On the mapping ability of fuzzy inference-systems. In *Proceedings of International Joint Conference on Neural Networks*, Seiten 708–713, 1992.
- [JX93] M. Jordan und L. Xu. Convergence results for the EM approach to mixtures of experts architectures. A.I. Memo 1458, Massachusetts Institute of Technology, <ftp://publicatios.ai.mit.edu>, 1993.
- [Kal60] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transacion of the ASME – Journal of Basic Engineering*, (3):35–45, 1960.
- [KGB86] S.R. Kelso, A.H. Ganong und T.H. Brown. Hebbian synapses in hippocampus. *Proceedings of the National Academy of Sciences*, 83:5326–5330, 1986.
- [KL98] A. Krzyżak und T. Linder. Radial basis function networks and complexity regularization in function learning. *IEEE Transactions on Neural Networks*, 9:247–256, 1998.
- [Klo96] M. Kloppenburg. Lokale Hauptkomponentenanalyse in hochdimensionalen Merkmalsräumen. Diplomarbeit, Inst. für Med. Optik, LMU München, 1996.
- [KLP⁺94] S. Kim, M. Lee, S. Park, S.-Y. Lee und C. Park. Neural linearizing control with radial basis function network for chemical processes. In *World Congress on Neural Networks-San Diego. 1994 International Neural Network Society Annual Meeting*, Band II, Seiten 94–99, 1994.
- [Koh82] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.
- [Koh90] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.
- [Kos92] B. Kosko. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice-Hall, 1992.

- [KS97] F. Kursawe und H.-P. Schwefel. Optimierung mit evolutionären Algorithmen. *Automatisierungstechnische Praxis*, 39(9):10–17, 1997.
- [KT97] M. Kloppenburg und P. Tavan. Deterministic annealing for density estimation by multivariate normal mixtures. *Physical Review E*, 55:2089–2092, 1997.
- [Küh94] J. Kühnel. *Datenrepräsentation durch Prototypen: Erweiterte Konzepte und ihre Anwendung in der Bild- und Spracherkennung*. Dissertation, Inst. für Med. Optik, LMU München, 1994.
- [LBD⁺90] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.W. Howard, W. Hubbard und L.D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, (1):541–551, 1990.
- [LBG80] Y. Linde, A. Buzo und R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–95, 1980.
- [LG88] R. Lenk und W. Gellert, Herausgeber. *Fachlexikon Physik*. Verlag Harri Deutsch, 1988.
- [Llo82] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28:129–137, 1982.
- [Lud89] G. Ludyk. *Theoretische Regelungstechnik I*. Springer, Berlin, 1989.
- [LVB⁺93] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. v. d. Malsburg, R. P. Würtz und W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993.
- [Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L.M. LeCam und J. Neyman, Herausgeber, *Proceeding of the Fifth Berkely Symposium on Mathematics, Statistics, and Probability*, Berkely, 1967. U. California Press.
- [MAC⁺95] M. Musavi, W. Ahmed, K. Chan, K. Fabris und D. Hummels. On the training of radial basis function classifiers. *Neural Networks*, 5:595–603, 1995.
- [Mal73] C.v.d. Malsburg. Self-organization of orientation sensitive cells in striate cortex. *Kybernetik*, 14:85–100, 1973.
- [Mal81] C.v.d. Malsburg. The correlation theory of brain function. Technical report, MPI for Biophysical Chemistry, 1981. Internal Report 81-2.
- [Mar69] D. Marr. A theory of cerebellar cortex. *Journal of Physiology (London)*, 202:437–470, 1969.
- [Mas84] I. Masao. *The cerebellum and neural control*. Raven Press, New York, 1984.

-
- [Mat99] The MathWorks, Inc., www.mathworks.com. *MATLAB Neural Network Toolbox*, 1999.
- [MBS93] T. Martinez, S. Berkovich und K. Schulten. Neural-gas network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4:558–569, 1993.
- [MD88] J. Moody und C. Darken. Learning with localized receptive fields. In D. Touretzky, G. Hinton und T. Sejnowski, Herausgeber, *Proceedings of the Connectionist Models Summer School*, Seiten 133–143, San Mateo, 1988. (Pittsburg 1988), Morgan Kaufmann Publishers.
- [MD89] J. Moody und C. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–294, 1989.
- [Met98] F. Metze. Integration neuronaler Methoden zur Phonemerkennung aus fließend gesprochener Sprache. Diplomarbeit, Inst. für Med. Optik, LMU München, 1998.
- [MG87] E.H. Mamdani und B.R. Gaines. *Fuzzy reasoning and its applications*. Academic Press, London, 1987.
- [MGPS96] T. Martinez, O. Gramckow, P. Protzel und G. Sörgel. Neuronale Netze zur Steuerung von Walzstraßen. *Automatisierungstechnische Praxis*, 10, 1996.
- [MKB94] K.V. Mardia, J.T. Kent und J.M. Bibby. *Multivariate Analysis*. Academic Press, London, 9 edition, 1994.
- [MP43] W.S. McCulloch und W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
- [MP69] M. Minsky und S. Papert. *Perceptrons*. MIT Press, Cambridge Massachusetts, 1969.
- [MS91] T. Martinez und K. Schulten. A 'neural-gas' network learns topologies. *Artificial Neural Networks*, Seiten 397–402, 1991.
- [MS94] R. Murray-Smith. *A Local Model Network Approach to Nonlinear Modelling*. Dissertation, Department of Computer Science, Univ. of Strathclyde, 1994.
- [MSE97] R. Murray-Smith und T. A. Johansen (Eds.), Herausgeber. Taylor and Francis, London, 1997.
- [MSW92] T. Miller, R. Sutton und P. Werbos, Herausgeber. *Neural Networks for Control*. MIT Press, 3 edition, 1992.

- [NK95] D. Nauck und R. Kruse. NEFCLASS – a neuro-fuzzy approach for the classification of data. In *Proceedings of the ACM Symposium on Applied Computing*. ACM Press, 1995.
- [NM65] J. Nelder und R. Mead. *Computer Journal*, 7:308–313, 1965.
- [PG89] T. Poggio und F. Girosi. A theorem of networks for approximation and learning. A. I. Memo 1140, Massachusetts Institute of Technology, 1989.
- [PG90a] T. Poggio und F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78:978–982, 1990.
- [PG90b] T. Poggio und F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.
- [PJ93] M. Pottman und P. Jörgl. Radial basis function networks for nonlinear process control - an internal model control approach. *Elektrotechnik und Informationstechnik (e&i)*, 110(7/8):336–341, 1993.
- [Pow87] M.J.D. Powell. Radial basis functions for multivariable interpolation: a review. In J.C. Mason und M.H. Cox, Herausgeber, *Algorithms for Approximation*, Band 2, Seiten 143–167. Clarendon Press, Oxford, 1987.
- [PTVF92] W. Press, S. Teukolsky, W. Vetterling und B. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.
- [RGF90] K. Rose, E. Gurewitz und G. C. Fox. Statistical mechanics and phase transitions in clustering. *Physical Review Letters*, 65:945–948, 1990.
- [RGF93] K. Rose, E. Gurewitz und G. C. Fox. Constrained clustering as an optimization method. Dept. of Electr. and Comp. Eng., Univ. of California, Santa Barbara, CA 93106, 1993.
- [RHW86] D.E. Rumelhart, G.E. Hinton und R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart und J.L. McClelland, Herausgeber, *Parallel Distributed Processing*, Band 1, Kapitel 8, Seiten 318–362. MIT Press, Cambridge, 1986.
- [Rip96] B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.
- [RMS92] H. Ritter, T. Martinetz und K. Schulten. *Neural Computation and Self-organizing Maps*. Addison-Wesley, New York, 1992.
- [Roj93] R. Rojas. *Theorie der neuronalen Netze*. Springer, Berlin, 1993.
- [Ros58] F. Rosenblatt. The perceptron: The probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.

-
- [RS90] J. Rubner und K. Schulten. Development of feature detectors by self-organization. *Biological Cybernetics*, 62:193–199, 1990.
- [RT89] J. Rubner und P. Tavan. A self-organizing network for principal-component analysis. *Europhysics Letters*, 10:693–698, 1989.
- [Rub89] J. Rubner. *Modelle zur Farberkennung*. Dissertation, Fakultät für Physik, Technische Universität München, 1989.
- [Sch86] P. Scheffe. *Künstliche Intelligenz – Überblick und Grundlagen*. Bibliographisches Institut, Mannheim, 1986.
- [Sch96] J.S. Schwegler. *Der Mensch – Anatomie und Physiologie*. Thieme, 1996.
- [Sem85] Sempé. *Radfahrer*. Diogenes, 1985.
- [Son97] M. Sonner. Selbstorganisation paralleler Experten zur Identifikation und Repräsentation wechselnder Systemzustände. Diplomarbeit, Inst. für Med. Optik, LMU München, 1997.
- [Sor70] H.W. Sorenson. Least-squares estimation: from gauss to kalman. *IEEE Spectrum*, 7(7):63–68, 1970.
- [SR87] T.J. Sejnowski und C.R. Rosenberg. Parallel networks that learn to pronounce English text. *Complex Systems*, 1:145–168, 1987.
- [ST85] R.f. Schmidt und G. Thews. *Physiologie des Menschen*. Springer, Berlin, 1985.
- [Ste88] C.F. Stevens. *Gehirn und Nervensystem*, Kapitel Die Nervenzelle, Seiten 3–31. Spektrum der Wissenschaft Verlagsgesellschaft, 1988.
- [Sug88] M. Sugeno. *Fuzzy Control*. Nikkan Kogyo, Tokyo, 1988.
- [TCA71] Y. Takahashi, C. Chan und D. Auslander. Parametereinstellung bei linearen DDC-algorithmen. *Regelungstechnik*, 19:237–244, 1971.
- [TGK90] P. Tavan, H. Grubmüller und H. Kühnel. Self-organization of assoziative memory and pattern classification: Recurrent signal processing on topological feature maps. *Biological Cybernetics*, 64:95–105, 1990.
- [Tho90] R.F. Thompson. *Das Gehirn*. Spektrum der Wissenschaft Verlagsgesellschaft, 1990.
- [Uni99] Universität Tübingen, www-ra.informatik.uni-tuebingen.de/SNNS. *Stuttgarter Neural Network Simulator*, 1999.
- [WCG97] Wacker-Chemie GmbH. Geschäftsbericht. Broschüre, 1997.

- [WCG99] Wacker-Chemie GmbH. Bilder, Daten und Fakten. Broschüre, 1999.
- [Wer74] P. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Dissertation, Harvard University, 1974.
- [WH60] B. Widrow und M.E. Hoff. Adaptive switching circuits. In *1960 IRE WESCON Convention Record*, Band 4, Seiten 96–104. IRE, New York, 1960.
- [Wie63] N. Wiener. *Kybernetik*. Econ-Verlag, Düsseldorf, 1963.
- [Wil98] A. Wilden. Mustererkennung, Restmengenklassifikation und deren Anwendung in der Spracherkennung. Diplomarbeit, Inst. für Med. Optik, LMU München, 1998.
- [WMS91] J. Walter, T. Martinetz und K. Schulten. Industrial robot learns visuo-motor coordination by means of 'neural-gas' network. *Artificial Neural Networks*, Seiten 357–364, 1991.
- [WS64] B. Widrow und F.W. Smith. Pattern-recognizing control systems. In *Computer and Information Sciences (COINS) Proceedings*. Spartan, Washington, 1964.
- [WS85] B. Widrow und S.D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, New York, 1985.
- [yC52] Ramón y Cajal. *Histologie du Système Nerveux de l'Homme et des Vertèbres*. Aus dem Spanischen übersetzt von L. Azoulay, Madrid, 1952.
- [YC98] Z. Yang und S. Chen. Robust maximum likelihood training of heteroscedastic probabilistic neural networks. *Neural Networks*, 11:739–747, 1998.
- [YSU94] A. L. Yuille, P. Stolorz und J. Utans. Statistical physics, mixtures of distributions, and the EM algorithm. *Neural Computation*, 6:334–340, 1994.
- [Zad73] L.A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(1):28–44, 1973.
- [Zim93] H.-J. Zimmermann. *Fuzzy-Technologien*. VDI-Verlag, 1993.

Lebenslauf

Name: Sebastian Albrecht
Geburtstag: 25. April 1966
Geburtsort: Hamburg

Schule

8/73 – 12/75 Grundschule in Kristinehamn/Schweden
1/76 – 7/78 Grundschule in München
9/78 – 6/86 Gymnasium in Haar/München

Studium und Beruf

11/88 – 12/94 Studium der Physik an der Technischen Universität München
2/95 Diplom im Fach Physik
2/95 – 9/99 wissenschaftlicher Mitarbeiter am Institut für Medizinische Optik
seit 1/98 Promotion an der Ludwig-Maximilians-Universität München

Danke !

Diese Arbeit verdankt ihr Entstehen der engagierten Unterstützung durch Prof. Dr. Paul Tavan, dem Leiter der Theoriegruppe am Institut für Medizinische Optik. Seine stete Diskussionsbereitschaft und seine direkte Kritik sind dieser Arbeit sehr zugute gekommen. Für diese – zeitweise sehr anstrengende – aber sehr fruchtbare Zusammenarbeit möchte ich ihm sehr danken. Besonderer Dank gebührt ihm auch für das hervorragend ausgestattete Arbeitsumfeld und sein stetes Engagement für meine finanzielle Absicherung.

Paul Tavan stellte 1995 den Kontakt mit der Wacker-Chemie GmbH her. Die daraus entstandene Zusammenarbeit lieferte für die vorliegende Arbeit wesentliche Impulse. Als entscheidender Motor dieser Zusammenarbeit erwies sich Thomas List. Erst seine Offenheit und Experimentierfreude machte die – manchmal sehr spannende¹ – Realisierung der in dieser Arbeit beschriebenen Verfahren in die Praxis möglich. Die “Wacker-Tage” und die vielen Diskussionen mit ihm werde ich in bester Erinnerung behalten.

Der graue Theoretikeralltag wurde mir durch die vielen Kollegen, mit denen ich in den letzten Jahren zusammenarbeiten durfte, erheblich erleichtert. Dies waren, ohne Anspruch auf Vollständigkeit, namentlich: Erik Baigar, Jan Busch, Dominik Dersch, Klaus Döhring, Bernhard Egwolf, Markus Eichinger, Basam Elia, Volker Groll, Helmut Grubmüller, Ingmar Hartl, Helmut Heller, Berthold Heymann, Martin Kloppenburg, Niels Kunstmann, Hans Kühnel, Alain Laederach, Gerald Mathias, Florian Metze, Christoph Niedermeier, Marco Nonella, Brita Schulze-Fiehn, Marcus Sonner, Paul Strodel, Andreas Wilden und Alexander Wörl. Gerade im letzten Jahr habe ich die vielen Gespräche mit Paul Strodel und die – auch sportliche – Gewitztheit von Alexander Wörl als sehr angenehm und erholsam empfunden.

Besonders in der Schlußphase war für mich die ideelle und praktische Unterstützung durch Basam Elia, Florian Wiest, Karen Meinberg und meine Geschwister Christoph, Dorothee und Thomas sehr wichtig. Dafür bin ich ihnen sehr dankbar. Meine Freundin Bettina Gaiser hat es immer wieder verstanden, mich von einer gewissen Geistesabwesenheit zu befreien, und hat mich unermüdlich aus jedem noch so tiefen Motivationsloch herausgeholt. Ohne ihre Hilfe könnte ich diese Zeilen letztendlich nicht schreiben.

Der größte Dank gilt jedoch meinen Eltern. Ihre Unterstützung und ihr Vertrauen machten erst mein Studium und damit diese Arbeit möglich.

¹ So führte der Start des von uns entwickelten Regelunssystems nicht nur einmal zu werksweiten Lautsprecherdurchsagen über den Ausfall großer Teile der Produktion

