Problemlösung durch Komitees neuronaler Netze

Zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

der Fakultät für Informatik

der Universität Karlsruhe (Technische Hochschule)

genehmigte

D i s s e r t a t i o n

von

Thomas Ragg

aus Rottweil

Tag der mündlichen Prüfung:	27. Oktober 2000
Erster Gutachter:	Prof. Dr. W. Menzel
Koreferent:	Prof. Dr. A. Zell

Für Arian, Anja, April und Janée

Danksagung

Eine Arbeit wie die vorliegende entsteht selten ohne die Unterstützung und Mitwirkung anderer. Ich darf hier die Gelegenheit nutzen, allen Menschen zu danken, die mich in unterschiedlichster Weise während dieser Zeit begleitet, unterstützt und gefördert haben, ohne deren Hilfe ich mir das Fertigstellen diese Arbeit nicht vorstellen kann.

Zuerst gilt mein Dank meinen Referenten, Prof. Wolfram Menzel und Prof. Andreas Zell, für die Betreuung meiner Arbeit, für die sie mir zahlreiche Anregungen geliefert haben. Prof. Menzel ermöglichte mir nicht nur ein überaus angenehmes Arbeitsklima an seinem Lehrstuhl, sondern kümmerte sich auch in Belangen, die über die fachliche Betreuung hinausgingen.

Gefördert wurde diese Arbeit im Rahmen des Landesschwerpunktprogramms 'Neuroinformatik' des Landes Baden-Württemberg und durch die Deutsche Forschungsgemeinschaft im Rahmen des Projektes 'Integrierte Entwicklung von Komitees Neuronaler Netze', wodurch ich diesen Institutionen Dank schulde.

Danken möchte ich auch Mitarbeitern der Landesbank Hessen-Thüringen und des Springer-Verlages, die ich in Projekten als angenehme und interessante Gesprächspartner kennenlernte und durch die ich immer wieder auf die Notwendigkeit gestoßen wurde, Methoden zu entwickeln, die für einen unkomlizierten und vielfachen Einsatz in der Praxis tauglich sind.

Besonderen Dank möchte ich meinen Kollegen am Institut für Logik, Komplexität und Deduktionssysteme aussprechen, die mir ein überaus angenehmes Arbeitsklima ermöglichten und nicht nur für fachliche Diskussionen zu haben waren. Die fachliche Zusammenarbeit mit Dr. Heinrich Braun, Johannes Feulner und Dr. Rainer Malaka half mir die ersten Schritte in dem Forschungsgebiet der neuronalen Netze und evolutionären Algorithmen zu machen. Interessante und hilfreiche Diskussionen konnte ich in den letzten Jahren häufig mit Dr. Martin Riedmiller, Dr. Dominik Hörnel, Dr. Steffen Gutjahr, Karin Höthker, Ralf Schoknecht, Martin Lauer und Artur Merke führen. Besonderer Dank gebührt Dr. Steffen Gutjahr, mit dem ich nicht nur thematisch und in Projekten zusammenarbeitete, sondern auch das Arbeitzimmer teilte, wodurch sich immer wieder interessante und fruchtbare Diskussionen entwickeln konnten, bei denen ich oft von seinem mathematischen Sachverstand profitiert habe.

Herzlicher Dank gilt Frau Klara Ragg, die das Erscheinen dieser Arbeit noch erleben konnte. Durch ihre Unterstützung in einigen schwierigen Situationen hat sie ebenfalls wesentlich zur Fertigstellung dieser Arbeit beigetragen.

Besonders herzliche Dankbarkeit gilt meiner Frau Janée, die mit ihrem Verständnis, ihrer Treue und Liebe mich in dieser Zeit durch Höhen und Tiefen begleitete, und meinen drei Kindern, April, Anja und Arian, die oft mit meiner Abwesenheit leben mußten. Aus der gemeinsamen Zeit, die wir miteinander verbringen durften, schöpfte ich nicht nur Kraft für meine Arbeit, sondern gewann auch immer wieder den Blick für die Realität des Lebens.

Weingarten, im Februar 2001

Inhaltsverzeichnis

Kapitel 1

Einleitung

Zwei sind besser daran als einer, weil sie eine gute Belohnung für ihre Mühe haben; denn wenn sie fallen, so richtet der eine seinen Genossen auf. Wehe aber dem einzelnen, welcher fällt, ohne daß ein zweiter da ist, um ihn aufzurichten! (Jesus Sirach 4:9-10)

Um gut überleben zu können, muß ein Gen gut mit den anderen Genen in derselben Art - demselben Fluß - zusammenwirken können. Um langfristig erhalten zu bleiben, muß ein Gen ein guter Kamerad sein. (Richard Dawkins)

1.1 Motivation

Das Zusammenwirken unabhängiger Individuen zur Lösung von Problemen ist nicht nur ein technisches Konzept, das in dieser Arbeit von Interesse ist, sondern vielmehr allgegenwärtig in unserem Leben und unserer Gesellschaft. Um das Konzept der vorliegenden Arbeit zu motivieren, möchte ich mich deshalb der Analogie zu bestimmten Entscheidungsprozessen in unserer Demokratie bedienen, bei denen eine grundlegende Problematik dadurch gegeben ist, daß die Fakten und Zusammenhänge unübersichtlich und vielschichtig sind. Angenommen es tritt eine bisher unbekannte ethische Fragestellung auf, in der die Gesellschaft als Ganzes eine Position beziehen muß. Die Frage sei derart, daß eine Antwort nicht offensichtlich ist, z.B. dadurch daß viele Faktoren sich gegenseitig beeinflussen und manche Antworten auch negative Rückkopplungen auf die Gesellschaft haben können. Typische negative Rückkopplungen sind beispielsweise die Fehllenkung von eigentlich sinnvollen Fördermitteln aufgrund der unübersichtlichen Zusammenhänge. Ein Beispiel für eine ethische Fragestellung wäre, ob und wann menschliche Gene gezielt manipuliert werden dürfen.

Häufig wird in solchen Fällen eine Kommission gebildet, z.B. durch die Regierung, und mit Mitgliedern besetzt. Jedes dieser Mitglieder wird sinnvollerweise spezielle Kenntnisse in einem oder mehreren Themengebieten haben, die die Fragestellung tangieren. Seine Meinung bildet sich das Mitglied auf Basis seines Wissens zu der Thematik, durch Fakten und Zusammenhänge. Nimmt man zur Vereinfachung an, daß die Menge der Fakten und Zusammenhänge, die jemand kennen kann, etwa konstant ist, dann folgt daraus, daß mit zunehmender Zahl an gewählten Themen die Entscheidungsgrundlage für ein Mitglied durch die abnehmende Zahl an Fakten immer dünner, seine Entscheidung somit unsicherer wird. Ist andererseits die Zahl seiner Themen zu klein, dann fehlt ihm vielleicht wichtige Information, um den richtigen Zusammenhang herzustellen. Beim Lernen aus Daten nennt man die Themengebiete Merkmale oder Eingabekomponenten und die Fakten Datenpunkte. Bei konstanter Zahl an Datenpunkten und steigender Zahl an Merkmalen ist der Eingaberaum immer dünner mit Punkten besetzt. Es wird immer schwerer, daraus eine Funktion zu erlernen. Das Problem nennt man den *Fluch der Dimensionen*.

Enthalten die Quellen zu einem Themengebiet nun verschiedene Informationen, eventuell sogar widersprüchliche, dann wird die Meinung der mit diesem Thema befaßten Kommissionsmitglieder stark davon abhängen, aus welcher Quelle sie ihre Informationen beziehen. Ebenso wird man im allgemeinen verschiedenartige Funktionen lernen, wenn unterschiedliche Daten zum Training verwendet werden. Diese Abhängigkeit von der ganz speziellen Wahl der Trainingsmenge spiegelt sich in der Varianz der Funktionen wider und wird auch als das Bias-Varianz Dilemma bezeichnet. In beiden Fällen muß man in geeigneter Weise dieser Varianz begegnen. Die Bildung einer Kommission bzw. eines Komitees ist gerade so ein Instrumentarium, verschiedene Meinungen wieder zu integrieren, anstatt eine Einzelmeinung zu favorisieren, die auf falschen Daten basieren könnte.

Information mag übertrieben dargestellt oder sogar fehlerhaft sein. Das Kommissionsmitglied kann nun die Information im Lichte seines Allgemeinwissens prüfen und bewerten. Es nutzt damit sein a priori Wissen, um zu entscheiden, ob die Information damit konsistent ist. Ist sie es nicht, dann wird es beide Quellen gegeneinander gewichten. Diese Vorgehensweise ähnelt der Bayes'schen Regel, ein Verfahren aus der Statistik, mit dem man a priori 'Wissen' nach Erhalt neuer Information in a posteriori 'Wissen' überführt. 'Wissen' wird dabei jeweils als Wahrscheinlichkeitsverteilung formuliert. Auf Basis der Bayes'schen Regel läßt sich unter Umständen ein Algorithmus formulieren, der es ermöglicht, die 'optimale' Gewichtung der beiden Informationsquellen zu berechnen. Die Gewichtung beider Quellen bezeichnet man beim Lernen aus Daten als Regularisierung.

Eine weitere Verbesserung in der Entscheidungsfindung könnte erreicht werden, wenn jedes Kommissionsmitglied angibt, wie sicher es sich selbst bei seiner Entscheidung einschätzt und diese Information dann nutzt, um die Meinungen zu gewichten. Das oben erwähnte Bayes'sche Verfahren ermöglicht das dadurch, daß man sein 'Wissen' mit Wahrscheinlichkeitsverteilungen beschreibt. Neben der wahrscheinlichsten Ausgabe, der eigentlichen Entscheidung, liefert es auch die Varianz der Verteilung und somit eine Aussage darüber ob die Wahrscheinlichkeitsmasse um die eigentliche Ausgabe konzentriert oder breit verteilt ist. Je breiter die Masse verteilt ist, als desto 'unsicherer' wird man die Entscheidung bewerten.

Einerseits soll jedes Kommissionsmitglied so gut wie möglich sein, andererseits sollen sie sich aber auch weitestgehend unabhängig ihre Meinung bilden. Ist eine Komission mit Mitgliedern besetzt, die sich in vielen Faktoren sehr ähnlich sind, dann deckt sich die Ent-

scheidung der Kommission stark mit der Meinung jedes Einzelnen. Haben die einzelnen Mitglieder dagegen einen unterschiedlichen Hintergrund und bedienen sie sich verschiedener Informationsquellen, dann begünstigt das den Entscheidungsprozeß in dem Sinne, daß ein bestmöglichster Konsens gefunden wird.

Diese Unabhängigkeit in der Entscheidung läßt sich bei Menschen schwerlich messen. Anders beim Lernen aus empirischen Daten. Die Abhängigkeit potentieller Komiteemitglieder ist hier einer mathematischen Beschreibung zugänglich. Damit sind die beiden prinzipiellen Grundgedanken formuliert, die in dieser Arbeit in ein Optimierungskonzept integriert werden sollen: Jedes Komiteemitglied soll im Rahmen seiner Möglichkeiten optimiert werden. Gleichzeitig soll ein übergeordneter Prozeß dafür sorgen, daß systematisch eine weitestgehende Unabhängigkeit der Komiteemitglieder erreicht wird. Eine Regierung kann das beipielsweise dadurch zu erreichen versuchen, daß sie darauf achtet, daß sich möglichst viele gesellschaftliche Gruppen in der Besetzung einer Komission widerspiegeln.

Das Ergebnis des Prozesses ist eine Auswahl von Netzen, die in sinnvoller Weise ein Komitee ergibt. Die Wahl der Komiteemitglieder muß also nicht der Entwickler treffen, sondern wird durch den Prozeß bestimmt.

Nachdem mit dieser Analogie eine grundlegende Motivation für das hier angestrebte Optimierungskonzept gegeben ist, werde ich in die bereits angerissenen Probleme, die beim Lernen aus Daten auftreten, genauer einführen, jeweils auf die möglichen Synergieeffekte durch Komiteebildung hinweisen und dann auf die Zielsetzung dieser Arbeit überleiten.

1.2 Lernen aus Daten

Eine immer wiederkehrende Aufgabe in Naturwissenschaften, Technik, Wirtschaft oder auch Medizin ist das Problem, aus empirisch ermittelten Daten den *funktionalen Zusammenhang* zwischen diesen Daten abzuleiten oder ein *Modell* dieser Daten anzugeben. Manchmal ist man in der glücklichen Lage, den Zusammenhang durch eine mathematische Beschreibung einer *Modellklasse* anzugeben, so daß man mittels der Daten 'nur noch' genau spezifizierte Parameter des Modells bestimmen muß, z.B. Mittelwert und Varianz einer Normalverteilung. Man bezeichnet diese Modelle entsprechend als parametrische Modelle.

Oft ist das Problemwissen aber nicht ausreichend, um eine genauere Beschreibung des mathematischen Zusammenhangs aufzustellen. In diesem Fall kann man auf nicht-parametrische oder semi-parametrische Verfahren, zu denen auch neuronale Netze gehören, zurückgreifen, die eine reichhaltigere Klasse von Modellen zulassen. Diese Reichhaltigkeit ist allerdings Segen und Fluch zugleich. Einerseits erlaubt die größere Flexibilität, die Daten besser zu approximieren, andererseits wird es aber auch wesentlich mehr verschiedenartige Modelle geben, die die Daten gut erklären. Die mögliche Verschiedenartigkeit der Modelle ist eine Folge der größeren Anzahl von freien Parametern. Diese Zahl steht in einem zunehmend schlechteren Verhältnis zur Zahl der Datenpunkte.

Erst seitdem allgemein erhältliche und erschwingliche Computersysteme die Erfassung, Speicherung und Aufbereitung von Daten erlauben, sind viele nicht-parametrische und semiparametrische Verfahren von praktischem Interesse, die aus diesen Daten einen funktionalen Zusammenhang ermitteln. Leistungsfähige Rechner lassen viele Algorithmen und Methoden

für reale Probleme erstmalig einsatzfähig werden. In dem Buch *Functional Data Analysis* schreiben die Autoren J. Ramsay und B.W. Silverman deswegen treffend, daß die Analyse von Daten sich in den letzten Jahren stürmisch weiterentwickelt hat (Ramsay & Silverman, 1997). Erst recht gilt das Gesagte für die Prognose mittels dieser Daten.

Ein junger Ast dieser Entwicklung sind Methoden der Komiteebildung, meist basierend auf sogenannten *Bootstrap-Verfahren* (Efron & Tibshirani, 1993), die den Datenraum geeignet *sampeln*. Das bedeutet das Folgende. Ist man an einer Eigenschaft $\rho(F)$ einer (Verteilungs-) Funktion F interessiert ist, dann kann man diese schätzen, in dem man viele Stichproben von F generiert. Beim Lernen aus Daten ist die Verteilungsfunktion F unbekannt. Bootstrapping bedeutet nun, $\rho(F)$ durch $\rho(F_n)$ zu schätzen, indem F durch die empirische Verteilungsfunktion F_n ersetzt wird, die aus den gegebenen Daten durch Ziehen mit Zurücklegen gewonnen wird (Silverman, 1986, Büning, 1991). Durch die Auswahl vieler Stichproben aus der eigentlichen Trainingsmenge steigt der Rechenaufwand gegenüber der Entwicklung eines einzelnen Modells entsprechend an. Dieser Mehraufwand ist aber notwendig, wenn man die Unsicherheiten bei der Modellentwicklung minimieren möchte (Büning, 1991). Mit der Zunahme an Rechenleistung wurden in den letzten Jahren etliche Methoden der Komiteebildung vorgeschlagen. Eine Übersicht bietet beispielsweise (Sharkey, 1999).

Im folgenden werde ich zuerst den Begriff der Generalisierung konkretisieren, um dann auf verschiedenen Stufen der Modellentwicklung zu fragen, wie eine gute Generalisierung gewährleistet werden kann. Das betrifft zuerst die Ebene der Parameteroptimierung und die Verwendung von Regularisierungstermen, dann die Ebene der geeigneten Eingabestruktur und eine damit zusammenhängende Merkmalsauswahl, und letztendlich die Frage nach der Abhängigkeit der Modelle von der speziellen Wahl der Trainingsdaten und der dadurch implizierten Komiteebildung. Eine wesentliche Motivation für meine Arbeit ist die Feststellung, daß eine Optimierung der Generalisierungsleistung eines Komitees sicherstellen muß, daß die Methoden auf den einzelnen Ebenen die dortigen Anforderungen für eine geeignete Modellentwicklung berücksichtigen. Zum Beispiel verbessert eine Komiteebildung über Modelle, die ohne Regularisierung trainiert wurden, mit hoher Wahrscheinlichkeit die Generalisierung im Vergleich zu den einzelnen Modellen, gilt das aber auch bezüglich optimal regularisierter Modelle? Es ist eine wesentliche Zielsetzung dieser Arbeit, in genau diese Lücke vorzustoßen. Dabei sollen die Defizite bisheriger Ansätze durch Einbeziehung aller Ebenen vermieden und durch die auf die Komiteebildung ausgerichtete integrierte Optimierung zusätzliche Synergieeffekte freigesetzt werden.

Für die hier gewählte Vorgehensweise werde ich eine mathematische Begründung herleiten, aus der sich auch ein Kriterium entwickeln läßt, welche und wieviele der Modelle aus einer gegebenen Menge von trainierten Modellen sinnvollerweise für ein gutes Komitee auszuwählen sind, denn Komiteemitglieder sollten *gute Kameraden sein*. Damit befreit man sich aus der unglücklichen Lage, die Komiteezusammensetzung heuristisch zu optimieren, wie das bisher noch häufig der Fall ist (vgl. dazu die Beiträge in (Sharkey, 1999)). Somit steht der gesamte Modellentwicklungsprozeß auf einem soliden theoretischen Fundament.

Die Abbildung 1.1 zeigt ein anschauliches Beispiel, das in der Arbeit immer wieder zur Verdeutlichung einiger wichtiger Aspekte dienen wird. Der zugrundeliegende Prozeß ist eine Sinusfunktion, die jeweils mit unterschiedlich starkem Rauschen behaftet ist. Im einen Fall ist die zugrundeliegende Struktur noch gut zu erkennen, im anderen fällt es, obwohl der

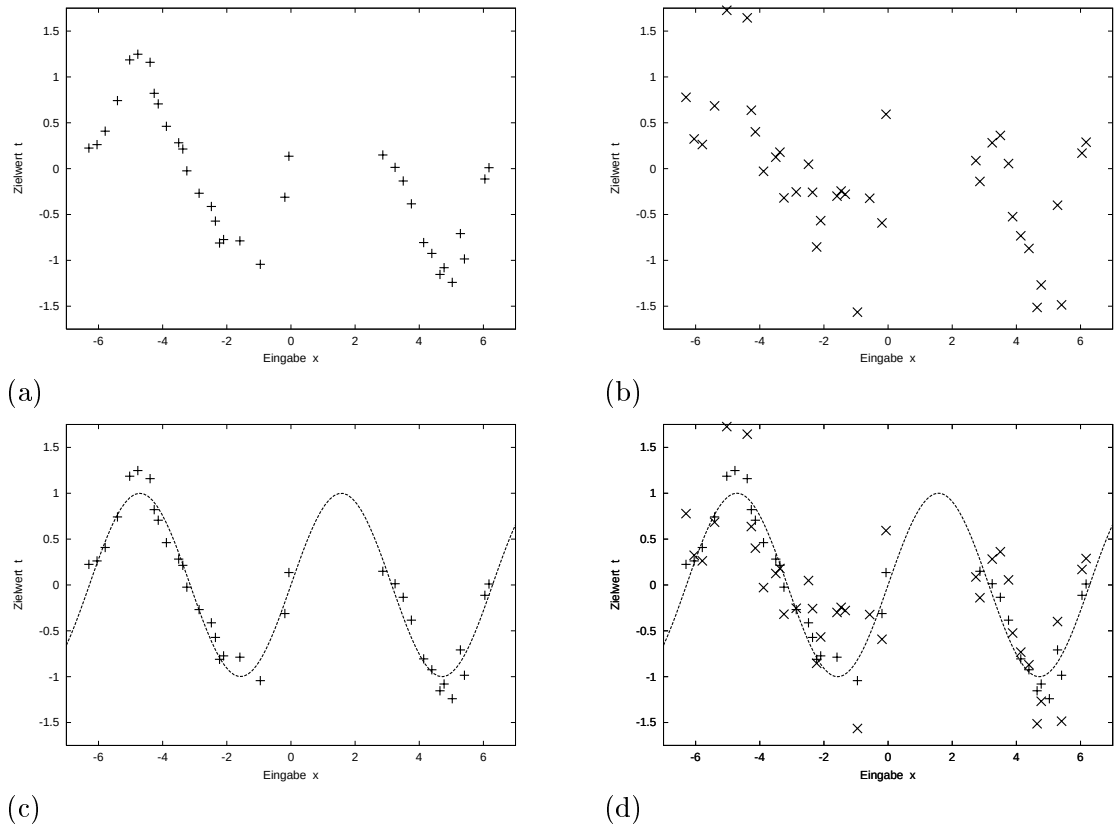


Abbildung 1.1: Ein Beispiel eines möglichen Datensatzes. Die Ausgabe wird durch einen sinus-artigen Prozeß generiert, der zusätzlich noch mit normalverteiltem Rauschen (mit Mittelwert 0) behaftet ist: $t = sin(x) + \epsilon$ Man beachte, daß die Datenpunkte nicht gleichmäßig über den Eingaberaum verteilt sind. Zwischen $x = 0$ und $x = 2$ muß die Funktion auf Basis der sonstigen Daten geschätzt werden, was die Aufgabe zusätzlich erschwert. a) Varianz des Rauschens ist 0.04. b) Varianz des Rauschens ist 0.4. In c) und d) ist zusätzlich zur besseren Visualisierung der erzeugende Prozeß mit eingezeichnet.

Graph nur zweidimensional ist, auch dem Menschen schwer, aus den gegebenen Daten auf den generierenden Prozeß zu schließen.

Ist nun ein Datensatz vorgegeben, dann ist es das Ziel des Lernprozesses, eine Wahrscheinlichkeit zu schätzen: Für jeden nicht in der Trainingsmenge liegenden Eingabewert $\mathbf{x}_i$ möchte man den wahrscheinlichsten zugehörigen Ausgabewert t_i schätzen. Gesucht ist also ein Modell der bedingten Wahrscheinlichkeit $p(t|\mathbf{x})$, um bei Beobachtung eines neuen Meßwertes $\mathbf{x}_i$ den Zielwert optimal zu prognostizieren, d.h. die Prognose sollte mit dem geringsten Risiko behaftet sein. In Abbildung 1.2 ist dieser Sachverhalt für den Datensatz aus Abbildung 1.1a verdeutlicht.

Für jeden Punkt im Eingaberaum wird also eine Wahrscheinlichkeitsverteilung geschätzt. In der Abbildung ist diese Verteilung für t quer zur Richtung von x eingezeichnet. Das erwartete Risiko wird nun minimiert, indem zu jedem neuen x_i immer das wahrscheinlichste t_i ausgegeben wird.

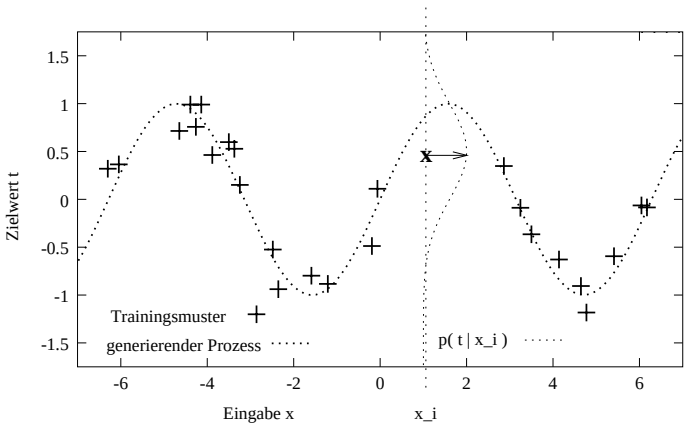


Abbildung 1.2: Neuronales Netz als Modell einer bedingten Wahrscheinlichkeit. Die Abbildung zeigt das Ziel des Lernens am Beispiel der Daten aus Abbildung 1.1a. Das Ziel ist es, ein Modell der bedingten Wahrscheinlichkeit $p(t|x)$ zu schätzen. Die Schätzung dieser Wahrscheinlichkeit für ein gegebenes x_i ist in der Abbildung eingetragen. Für jedes t_i erhält man dessen Wahrscheinlichkeit, der zugehörige Ausgabewert zu sein. Die Ausgabe des Modells ist das Maximum der Wahrscheinlichkeitsverteilung, im Bild mit einem Pfeil markiert. Die wahrscheinlichste Ausgabe entspricht dem Fußpunkt des Pfeils und liegt etwas unterhalb der Sinuskurve.

1.2.1 Generalisierung

In diesem und dem folgenden Abschnitt wird dargelegt, warum es nicht genügt, gute Generalisierung als minimalen Fehler auf einer unabhängigen Testmenge zu definieren, sondern daß es notwendig ist, den Fehler einer regularisierten Fehlerfunktion zu minimieren.

Versucht man mittels neuronaler Netze einen funktionalen Zusammenhang aus Beispielen zu erlernen, dann ist das Ziel dieses Lernvorgangs, daß eine Struktur der Daten erkannt wird und das trainierte Netz somit sein 'Wissen' auf unbekannte Fälle übertragen kann. Die Struktur in den Daten muß dabei von zufälligem Rauschen unterschieden werden. Dies stellt man sich in analoger Weise zum Lernen beim Menschen vor. Hat man einmal das Wesentliche des Buchstabens 'A' gelernt, dann erkennt man den Buchstaben in vielen Variationen wieder: $\mathcal{A}$, A, A oder auch $\forall$. Diese Übertragung von Wissen bezeichnet man oft als Generalisierung. Es bleibt dabei erstmal unklar, wie man die Güte der Generalisierung messen kann, bzw. wie man sie mathematisch beschreiben könnte.

Im Sprachgebrauch in der Literatur gibt es durchaus Unterschiede, was mit Generalisierung eigentlich gemeint ist, folglich auch verschiedene Auffassungen über die Zielsetzung beim Lernen aus Daten. Das schlägt sich auch in den Methoden nieder, die angewendet werden, um gute Generalisierung zu erreichen. In nicht wenigen Lehrbüchern über neuronale Netze wird über Generalisierung, das eigentliche Ziel des Lernens, überhaupt keine Aussage gemacht. Die gängigste Definition von Generalisierung wird meist als

> *das Prüfen der Leistung auf einer unabhängigen Testmenge*

gegeben (Schürmann, 1996), siehe auch (Nauck *et al.*, 1994, Hecht-Nielsen, 1991, Deco & Obradovic, 1996). In ähnlicher Weise charakterisiert (Bishop, 1995) das Problem der Generalisierung bei einem einführenden Beispiel als

> *die Aufgabe, einen Klassifikator zu entwickeln, der vorher unbekannte Muster richtig klassifiziert.*

Im Kontext von 'Lernen und Generalisierung' schreibt er (siehe dazu auch (Ripley., 1996, Hertz *et al.*, 1991)),

> *daß es nicht das Ziel des Trainings ist, eine exakte Repräsentation der Trainingsdaten zu finden, sondern ein statistisches Modell des Prozesses, der die Daten generiert (hat). Dies ist wichtig, wenn das Modell gut generalisieren soll.*

Damit ist die Frage, was man eigentlich lernen möchte, sehr viel genauer beantwortet. Approximiert man den datenerzeugenden Prozeß, dann wird der Fehler über alle Muster, die man zukünftig seinem Modell vorlegt, minimal sein. Das heißt, die Generalisierung ist maximal, wenn der erwartete Fehler auf einem beliebig gezogenen Muster klein ist. Es wird aber sicherlich endliche Teilmengen geben, auf denen der Testfehler trotzdem groß ist. Dies ist insbesondere dann der Fall, wenn die Daten mit einem großen Rauschanteil behaftet sind (vgl. Abbildung 1.1). Die Leistung auf einer Testmenge alleine ist deswegen noch kein geeignetes Maß für die Generalisierungsfähigkeit. Aufgrund des Rauschanteils bedeutet ein quantitativ großer Testfehler nicht unbedingt, daß das Modell schlecht ist. Auch wäre es in dem Fall, daß man Modelle anhand ihres Testfehlers vergleicht, vorteilhaft, mit einem statistischen Test zu überprüfen, ob die Trainings- und Testdaten zur selben Verteilung gehören (vgl. Kapitel A.1.4).

Andererseits bedeutet ein quantitativ kleiner Testfehler auch nicht automatisch, daß das zugehörige Modell gut ist. Im gegebenen Beispiel erreicht auch die Funktion $k \sin \alpha x$ einen kleinen Fehler auf einer endlichen Testmenge, wenn α nur groß genug gewählt wird (Abbildung 1.3.

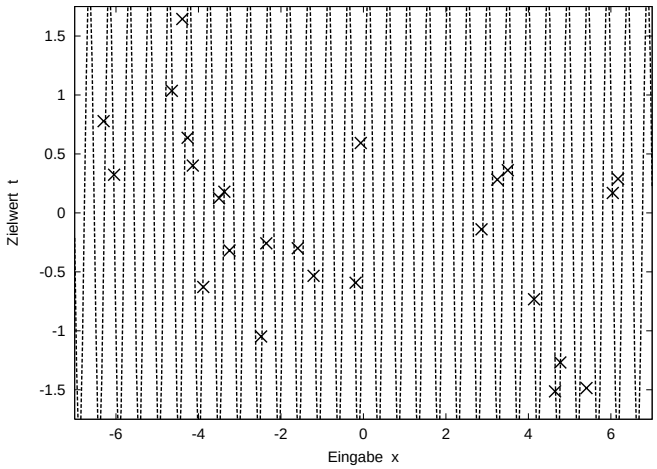


Abbildung 1.3: Die Funktion $k \sin \alpha x$ erzeugt auf endlichen Testmengen immer einen kleinen Fehler, solange α nur groß genug gewählt wird. Dies verdeutlicht die Notwendigkeit der Regularisierung, d.h. der Gewichtung von Komplexität des Modells zum Trainigsfehler.

Offensichtlich steht zur Messung der Generalisierungsfähigkeit kein absolutes Kriterium zur Verfügung. Sonst wäre es nicht weiter schwierig, dieses mittels einer evolutionären Strategie direkt zu optimieren. Einen brauchbaren Schätzwert wird man aber erhalten, wenn man beachtet, daß die Komplexität der Funktion in geeigneter Beziehung zur Anzahl der Datenpunkte stehen muß. Bezogen auf die Funktion $k \sin \alpha x$ heißt das, daß ein vernünftiger Lösungsansatz die Zahl der Schwingungen begrenzen sollte. Genau dies wird durch das Prinzip der Regularisierung gewährleistet, das im folgenden Abschnitt beschrieben wird.

1.2.2 Regularisierung

Vergrößert man die Zahl der freien Parameter fortlaufend, dann sind neuronale Netze im allgemeinen in der Lage, die Trainingsdaten immer genauer anzunähern (Bishop, 1995). Allerdings nehmen die möglichen Erklärungen der Daten an Zahl und Komplexität zu. Dadurch steigt die Gefahr, ein falsches Modell auszuwählen. Ein wichtiger Grund dafür sind Struktur und starke Schwankungen, die gelernt wurden, aber durch die Daten nicht ausreichend gesichert sind. Um nun möglichst 'glatte' Funktionen zu finden, wendet man sogenannte Regularisierungstechniken an, mit denen man die Komplexität der Modelle kontrollieren kann. Die geeignete Regularisierung ist ein *wesentlicher und wichtiger Aspekt bei der Modellentwicklung und nicht etwa ein optionaler Zusatz*, wie das (Ramsay & Silverman, 1997) ausdrücken.

Methoden der Regularisierung gehen zurück auf die Entdeckung Hadamards am Anfang des Jahrhunderts, welcher Operatorgleichungen der Form

$$Af = F, f \in \mathcal{F}$$

untersuchte. A und F kann man sich beim Lernen aus Daten als die Eingabe- und Ausgabeinformation (als Matrix) vorstellen. Hadamard stellte fest, daß kleine Abweichungen auf der rechten Seite, F_δ statt F, die Lösung drastisch verändern können. Man spricht in diesem Fall von schlecht gestellten Problemen (engl. ill-posed problems). Bezogen auf das Lernen aus Daten bedeutet das, daß eine kleine Veränderung der Trainingsinformation zu einem gänzlich anderen Modell führen kann. Minimiert man nun den Fehler

$$E(f) = ||Af - F_\delta||^2,$$

dann ist damit noch keine gute Approximation der Lösung garantiert, wenn $\delta = |F - F_\delta|$ klein wird. Ausgehend von dieser Erkenntnis zeigt die Theorie der Regularisierung, daß man statt dessen durch Minimierung des regularisierten Fehlers

$$E(f) = ||Af - F_\delta||^2 + \gamma(\delta)\,\Omega(f)$$

eine Sequenz von Lösungen erhält, die bei geeigneter Wahl von γ gegen die gewünschte Lösung konvergiert, wenn δ gegen 0 strebt. $\Omega(f)$ gehört dabei zu einer Klasse von Funktionalen, die die Komplexität der Funktion angeben, wie z.B. die Summe der quadratischen Gewichte bei neuronalen Netzen (siehe Kapitel 2.3). $\gamma(\delta)$ ist ein konstanter Faktor, der von den Abweichungen - dem Rauschen in den Daten - abhängt. Ein ganz wesentlicher Aspekt geeigneter Regularisierung ist es, diesen Faktor geeignet zu bestimmen, wie wir in Kapitel 2.5 noch sehen werden.

Die Theorie der Regularisierung wurde von Tikhonov und anderen in den sechziger Jahren entwickelt (Vapnik, 1982, Vapnik, 1995). Vapnik führt aus, daß schlecht gestellte Probleme insbesondere dann auftreten, wenn man Ursache und Wirkung vertauscht: Man sucht die unbekannten Ursachen ausgehend von den bekannten Wirkungen. Für das Lernen aus Daten ist es wichtig zu wissen, daß das grundlegende Problem - eine Dichtefunktion anhand der Daten zu schätzen - schlecht gestellt ist. Aus diesem Grund ist die Verwendung von Regularisierungsmethoden zwingend.

1.2.3 Fluch der Dimensionen

In realen Anwendungen stehen nur begrenzt viele Daten zur Verfügung, um die Parameter eines Modells zu bestimmen, weil z.B. bei Zeitreihen keine längere Historie existiert oder die Beschaffung der Daten teuer sein kann wie im Falle medizinischer Klassifikationsprobleme. Die Dimensionalität des Eingabevektors muß aber in einer vernünftigen Relation zur Anzahl der Daten stehen. Durch Hinzunahme weiterer Merkmale steigt zwar der Informationsgehalt der Eingabe bezüglich der Ausgabe, aber gleichzeitig stehen pro Dimension immer weniger Datenpunkte zur Verfügung, um die Parameter zu bestimmen. Das bedeutet, daß die Klasse der Funktionen, innerhalb derer die Lösung gesucht wird, mit jeder zusätzlichen Komponente stark wächst. Dieses Problem ist als *Fluch der Dimensionen* (engl.: *curse of dimensionality*) bekannt (White, 1989, Bishop, 1995) und wird manchmal auch als das *empty space phenomenon* bezeichnet (Scott & Thompson, 1983, Silverman, 1986). Als Konsequenz daraus ergibt sich, daß ein Verzicht auf zusätzliche Information zu einer genaueren Approximation des zugrundeliegenden Prozesses in einem niedrig-dimensionaleren Raum führen kann.

Ein weiterer Aspekt, den es zu beachten gilt, ist der Zusammenhang zwischen der Zahl der Eingabekomponenten und der Größe des Regularisierungstermes. Verwendet man neuronale Netze als Modellklasse und werden bei sonst gleichbleibender Netzstruktur weitere Eingabemerkmale hinzugefügt, dann vergrößert sich die Komplexität des Netzes entsprechend. Das heißt, wenn die zusätzlichen Merkmale nur wenig neue Information beitragen und damit der Fehler auf den Daten sich kaum verringert, so sorgen sie doch dafür, daß die Komplexität und damit auch der Regularisierungterm $\Omega(f)$ wächst. Bei neuronalen Netzen nimmt in diesem Fall die Summe der quadratischen Gewichte rasch zu. Die Konsequenz davon ist, daß man bei hochdimensionalen Eingaberäumen als Ergebnis des Lernprozesses häufig nahezu lineare Lösungen erhält (vgl. Kapitel 5. Der Grund dafür ist, daß durch die kleinen Gewichte alle Neuronen des Netzes im linearen Bereich arbeiten.

Entwirft man seine Problemlösung von vornherein als Komitee, dann ist es nicht nötig, für die Gesamtlösung auf Eingabekomponenten und somit auf Information zu verzichten. Vielmehr kann man Modelle trainieren, die verschiedene Merkmale verwenden, das heißt, unterschiedliche Sichten auf das Problem haben. Jedes einzelne Modell verwendet also einen kleinen Eingaberaum, d.h. die Problematik vieler Eingabekomponenten wird vermieden, während das Komitee durch die Kombination seiner Mitglieder aber die Information des gesamten Eingaberaumes ausnutzen kann.

Bei den realen Problemstellungen, die in dieser Arbeit betrachtet werden, ist es tatsächlich so, daß man eine große Anzahl möglicher Merkmale zur Verfügung hat, aus denen man aus den eben genannten Gründen eine Teilmenge auswählen muß. Das folgende Beispiel in Abbildung 1.4 verdeutlicht diese Problematik. Mit den 1000 zufällig gewählten Eingabevektoren wurden entsprechende Modelle erstellt. Liegt die Größe zwischen 5 und 10 Komponenten, dann zeigen sich die besten Resultate. Die Zahl der Fehlklassifikationen des besten Modells mit sechs Merkmalen ist um ein Drittel kleiner wie bei Verwendung aller Merkmale! Die 'richtigen' sechs Merkmale müssen aber ersteinmal bestimmt werden. Wünschenswert ist also ein Algorithmus, der gerade die geeigneten Eingabemerkmale findet.

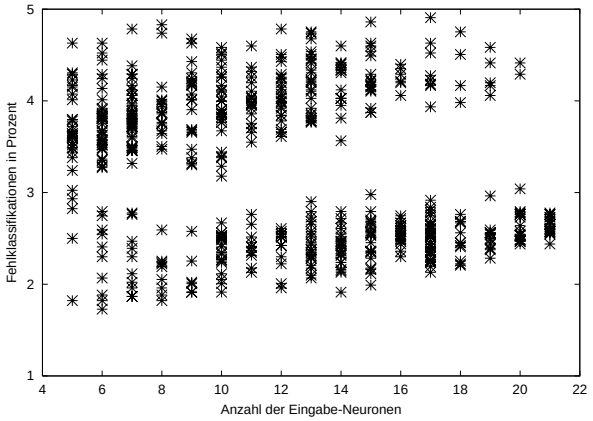


Abbildung 1.4: Die Abbildung zeigt die Auswirkungen des *'curse of dimensionality'* für das Schilddrüsen-Klassifikationsproblem, das in Kapitel Kapitel 5.2.2 noch ausführlicher beschrieben ist. Für 1000 zufällig gewählte Eingaberäume, die mindestens 5 und maximal 21 Komponenten besitzen, wurden die zugehörigen Netze trainiert. Der minimale Klassifikationsfehler wird für ein Modell mit 6 Merkmalen erreicht und liegt bei 1.7%. Damit ist er um ein Drittel kleiner wie bei der Verwendung aller Dimensionen 2.6%. Aufallend ist, daß es in der Mitte einen Bereich gibt (3% bis 3.5%), in dem kaum Netze liegen. Das deutet darauf hin, daß es wenige wichtige Merkmale gibt, die die Leistung ganz entscheidend beeinflussen. Fehlen diese, dann ist die Güte der Netze deutlich geringer. Man beachte, daß für jede Anzahl an Eingabeneuronen (< 21), viele Merkmalskombinationen existieren. Der Mittelwert des Fehlers wäre deswegen hier nicht besonders aussagefähig.

1.2.4 Auswahl der Trainingsdaten

Eine weitere fundamentale Einsicht in die Problematik des Lernens aus Daten leitet sich aus dem sogenannten *Bias-Varianz-Dilemma* ab. Der Fehler, den ein Modell auf den Daten macht, läßt sich in zwei Teile aufspalten: Bias und Varianz. Der Bias beschreibt die durchschnittliche Abweichung der Modelle von den empirischen Daten, während die Varianz angibt, wie sehr das einzelne Modell von der speziellen Wahl der Trainingsdaten abhängt. Die beiden Größen verhalten sich komplementär: Viele Methoden, die zu einer Verringerung der Varianz führen, ziehen einen steigenden Bias nach sich und umgekehrt. Dies bezeichnet man als das *Bias-Varianz-Dilemma*. Um eine gute Generalisierungsfähigkeit zu erreichen, muß man die beiden Größen richtig gewichten (Geman *et al.*, 1992, Heskes, 1998).

Hat man einen guten Ausgleich zwischen den beiden Größen gefunden, dann garantiert das nicht, daß man eine einzige optimale Lösung findet. Vielmehr wird man für eine praktische Anwendung immer eine Vielzahl von Modellen erhalten, die eine ähnliche Güte haben, von denen dann letztendlich eines ausgewählt werden muß. Bildet man in geeigneter Weise ein *Komitee* aus den Modellen, dann hat man ohne Mehraufwand zwei Vorteile: Erstens zieht man Nutzen aus allen Modellen, die erstellt wurden, d.h. die Information über diese Modelle wird verwertet. Zweitens ist die Berechnung der Güte der einzelnen Modelle immer mit einem Rauschen behaftet aufgrund der Endlichkeit der Datenmenge oder approximativer Annahmen der Berechnungsvorschrift. Mittelt man über mehrere Modelle, wird dieser Einfluß reduziert.

Die mögliche Zunahme an Generalisierungsfähigkeit begründet sich auch dadurch, daß durch *Komiteebildung* die Varianz des Gesamtsystems reduziert wird, ohne daß sich der Bias ändert. Man findet also einen günstigeren Ausgleich zwischen den beiden Größen, als es für einzelne Modelle möglich ist (Krogh & Vedelsby, 1995). Eine wesentliche Voraussetzung

dafür ist, daß die Komiteemitglieder möglichst unabhängig voneinander entscheiden, d.h. eine andere Sicht auf das Problem haben. Bildet man von den gegebenen Daten mehrere Teilmengen durch zufällige Auswahl von Mustern und trainiert für jede so gewonnene Trainingsmenge ein Modell, dann wird das die Leistung des Komitees steigern. Das heißt auch, das die Vorgabe einer Datenmenge bei der Komiteebildung das Gesamtergebnis weniger beeinflußt, dadurch daß über mehrere Samples integriert wird.

1.2.5 Modellauswahl

Ein weiteres Problem bei der Modellbildung ist das Fehlen eines klaren Auswahlkriteriums. Hat man erst einmal verschiedene Modelle entwickelt, dann muß man sich am Ende des Prozesses für eines entscheiden oder aus einer plausiblen Teilmenge ein Komitee bilden. Die gängigste Methode ist die Verwendung zweier zusätzlicher Datenmengen - einer Cross-Validierungsmenge und einer Testmenge. Man wählt das Modell aus, das auf der Cross-Validierunsgmenge den kleinsten Fehler hat, und überprüft die Auswahl nochmals auf einer zusätzlichen Datenmenge (Bishop, 1995). Der Nachteil dieses Vorgehens ist, daß man weniger Daten zum Trainieren zur Verfügung hat und die Methode bei wiederholter Anwendung zum Overfitting neigt. Das heißt, verwirft man seine Modellentwicklung mehrfach, dann steigt die Wahrscheinlichkeit, daß man zufälligerweise gute Werte erzielt.

Wünschenswert ist deshalb ein Selektionsprozeß, der möglichst ohne zusätzliche Daten auskommt. Hierzu lassen sich einige Gütekriterien angeben, die auf statistischen Verfahren beruhen (Bishop, 1995). Ein Beispiel ist das Gütemaß, das der Bayes'sche Ansatz berechnet, die sogenannte Evidenz. Man wählt dabei unter allen trainierten Modellen dasjenige aus, das die höchste Evidenz hat (MacKay, 1992, Bishop, 1995, Gutjahr, 1999).

Auch statistische Gütekriterien sind mit einem Rauschen behaftet aufgrund idealisierender oder impliziter Annahmen, z.B. über die genaue Form des Rauschens in den Daten. Bishop schlägt deshalb beispielsweise vor, die Evidenz als groben Indikator zu verwenden und mehrere Modelle mit hoher Evidenz zu kombinieren, um negative Einflüsse eines ungenauen Gütekriteriums zu reduzieren.

In jedem Fall sollte eine Entwurfsmethodik eine klare Empfehlung enthalten, welches oder welche Modelle letztendlich verwendet werden sollen. Die Komiteebildung bietet hier einen nicht zu unterschätzenden Vorteil, da sie von der Aufgabe, eine Auswahl zu treffen, weitgehend entlastet. Auch Modelle mit einem relativ großem Fehler können hier noch einen Zugewinn an Leistung bringen: *Ein Kamerad, der fällt, kann durch die anderen wieder aufgerichtet werden.*

1.3 Zielsetzung und Aufbau der Arbeit

Folgt man dem in der Wissenschaft allgemein anerkannten Prinzip von *Occam's Razzor*, dann bevorzugt man unter den möglichen Erklärungen für die Daten diejenigen, die den Zusammenhang mit möglichst wenig Annahmen möglichst gut beschreiben (Cover & Thomas, 1991, Bishop, 1995). Diesem Sparsamkeitsprinzip begegnet man auf mehreren Ebenen des Modellentwurfs - bei der Optimierung der freien Parameter, bei der Auswahl der Merkmale

und der Wahl der Trainingsdaten. Eine wichtige Folgerung daraus ist, daß man die resultierenden Probleme wesentlich abmildern kann, wenn man über mehrere Modelle geeignet integriert.

Problemlösungen auf Basis von Komitees sind also ein wichtiges Konzept zur Optimierung der Leistung neuronaler Modelle. Entwirft man ein neuronales System als Komitee, dann sollte man diese Entscheidung bereits auf den einzelnen Entwurfsebenen beachten. Das heißt, der Entwurfsprozeß muß so gestaltet sein, daß auch verschiedenartige Lösungen generiert werden können. Das Ziel der vorliegenden Arbeit ist die Erarbeitung einer allgemein einsetzbaren Entwurfsmethodik, welche die verschiedenen Entwicklungsstufen eines neuronalen Modells nicht nacheinander, sondern integriert in einer einzigen Optimierungsphase bearbeitet und durch Synergieeffekte eine signifikante Leistungssteigerung ermöglicht. Die erste Stufe umfaßt die Parameteroptimierung verbunden mit der geeigneten Regularisierung. Die zweite Stufe betrifft die Optimierung der Eingabestruktur und der Topologie. Auf der letzten Stufe wird die Abhängigkeit von der speziellen Wahl der Trainingsdaten behandelt. Die eigentliche Schwierigkeit liegt dabei darin, den Prozeß so zu gestalten, daß einerseits die Optimierung auf den einzelnen Ebenen ermöglicht, andererseits die Suche aber so gestaltet wird, daß möglichst unabhängige Netze gefunden werden. Es muß an dieser Stelle betont werden, daß der gesamte Optimierungsprozeß ohne die Verwendung von Kreuzvalidierung auskommt. Alle wesentlichen Parameter werden automatisch adaptiert. Unkritische Parameter des Suchprozesses lassen sich einfach durch Vorüberlegungen oder Erfahrungswerte festlegen, wie später noch im Detail ausgeführt wird.

1.3.1 Abgrenzung

Die vorliegende Arbeit stellt eine Methodik bereit, um in einem Parameterraum nach 'guten' Lösungen zu suchen, deren zugehörige Modelle im stochastischen Sinne aber auch möglichst unabhängig sind. Welche Modellklasse konkret verwendet wird, ob neuronale Netze, Entscheidungsbäume, Gauß-Prozesse oder Supportvektor-Maschinen, ist von untergeordneter Bedeutung, solange für das Modell ein analytisches Gütekriterium berechnet werden kann. Von nicht zu unterschätzender Wichtigkeit ist allerdings die Frage, ob für die Modellklasse eine Methode der automatischen Regularisierung in ähnlicher Weise wie das Bayes'sche Lernen für Neuronale Netze existiert. Ist dies nicht der Fall, dann gibt es zumindest einen Parameter, der durch Kreuzvalidierung optimiert werden muß. Bei Supportvektor-Maschinen gibt man bisher beispielsweise noch eine Toleranzgrenze ξ für Fehlklassifikationen vor (Burges, 1998). Es ist offensichtlich, daß das für eine automatische Suche, bei der jedes Modell prinzipiell eine andere Merkmalsstruktur und Trainingsdaten haben kann, ungeeignet ist, da ξ jedesmal individuell bestimmt werden müßte. Auch die Methoden für die Merkmalsselektion und Topologieoptimierung sind gegenüber der Komiteebildung zweitrangig, da für den Suchprozeß vor allem die Unabhängigkeit der Modelle betrachtet wird.

Neben dem Vergleich der hier entwickelten Konzepte zur Auswahl des Netzes mit der höchsten Evidenz beim Bayes'schen Lernen, sind insbesondere auch andere bekannte Verfahren zur Komiteebildung von Interesse. Hier sind vor allem das sogenannte Bagging und Boosting zu nennen (Breiman, 1996, Freund & Schapire, 1996). Abzugrenzen ist die Arbeit gegen Ansätze, die die Fehlerfunktion verändern, um Komiteemitglieder zu gewinnen. Der Ansatz von Rosen, der eine Unkorreliertheitsbedingung für neuronale Netze als Strafterm

formuliert (Rosen, 1996), weist ganz entscheidende Schwachstellen auf. Ebenso erwähnenswert ist der Ansatz von Hashem, der ebenfalls Korrelationen zwischen Modellen betrachtet, aber kein analytisches Kriterium zur Optimierung des Komitees verwendet, sondern iterativ von einer Kreuzvalidierungsmenge Gebrauch macht (Hashem, 1999). Für verrauschte Daten führt dieses iterative Vorgehen unweigerlich zu starker Anpassung an die Kreuzvalidierungsmenge, d.h. Overfitting. In Kapitel 3.2 wird Problematik dieser Ansätze genauer betrachtet.

Im experimentellen Teil der Arbeit werden vier prinzipielle Verfahren anhand verschiedener Anwendungen verglichen: (I) Bayes'sches Lernen ohne Komiteebildung, (II) Bagging und Boosting auf Basis regularisierter neuronaler Netze, (III + IV) die hier vorgestellten Verfahren - evolutive Optimierung der Evidenz und das integrierte Konzept zur Komiteebildung. Dabei werden für jedes Verfahren 10 Versuche mit verschiedenen Initialisierungen durchgerechnet. Das heißt, trainiert man normalerweise 100 Netze mit Bayes'schem Lernen, um davon eines auszuwählen, dann wird dieses Training zehnmal wiederholt. Entsprechend werden die anderen drei Verfahren zehnmal durchgeführt. Die Ergebnisse werden auf Basis eines Signifikanztests (t-Test auf Gleichheit der Mittelwerte) verglichen, der im Anhang A.1.1 beschrieben ist. Damit soll gewährleistet werden, daß die Verbesserungen nicht auf Zufälligkeiten beruhen, wie das leider oft der Fall ist (Beck-Bornholdt & Dubben, 1998). Eine Methode wird dann als besser gewertet, wenn sie im Mittel einen signifikant geringeren Fehler aufweist.

Als Anwendungen werden einerseits reale Problemstellungen betrachtet, z.B. die Prognose von Finanzzeitreihen oder Absatzzahlen von Zeitungen, die hier am Institut im Rahmen von Kooperationsprojekten mit der Industrie bearbeitet wurden. Andererseits greife ich auf bekannte Benchmarksammlungen zurück, da hier auch Vergleichswerte von anderen Forschungsgruppen vorliegen, insbesondere für das Boosting-Verfahren, von dem verschiedene Varianten existieren. Es sei noch angemerkt, daß Boosting für Probleme mit großem Rauschanteil, wie ich sie hier betrachte, nur bedingt geeignet ist. Das Verfahren konzentriert sich vor allem auf die Muster, die schwer zu lernen sind. Ein hoher Rauschanteil führt deshalb zu deutlichen Overfitting-Effekten (Rätsch *et al.*, 1998).

Die Kombination mit anderen Verfahren zur Steigerung der Generalisierungsleistung, z.B. die in jüngster Zeit populäre Methode des *Multi-Tasking* (Caruana, 1996, Gutjahr, 1999), werden hier nicht weiter betrachtet, um nicht ins Uferlose auszuschweifen. Die einzelnen Methoden schließen sich meist nicht gegenseitig aus, sondern sind im Idealfall miteinander kombinierbar, womit sich gegebenenfalls weitere Synergieeffekte realisieren lassen.

1.3.2 Vorgehensweise und Aufbau

Der hier vorliegende Ansatz läßt sich zuerst in den Bereich der Komiteebildung einordnen, stützt sich aber auch wesentlich auf Konzepte, die klassischerweise anderen Teilgebieten des maschinellen Lernens zugeordnet werden: Neuronale Netze und Bayes'sches Lernen, hierarchische Clusterverfahren und Evolutionäre Suchstrategien. Diese Bausteine werden in den ersten Kapiteln eingeführt. Die Abbildung 1.5 gibt einen graphischen Überblick über die Bausteine und die inhaltlichen Abhängigkeiten der Arbeit.

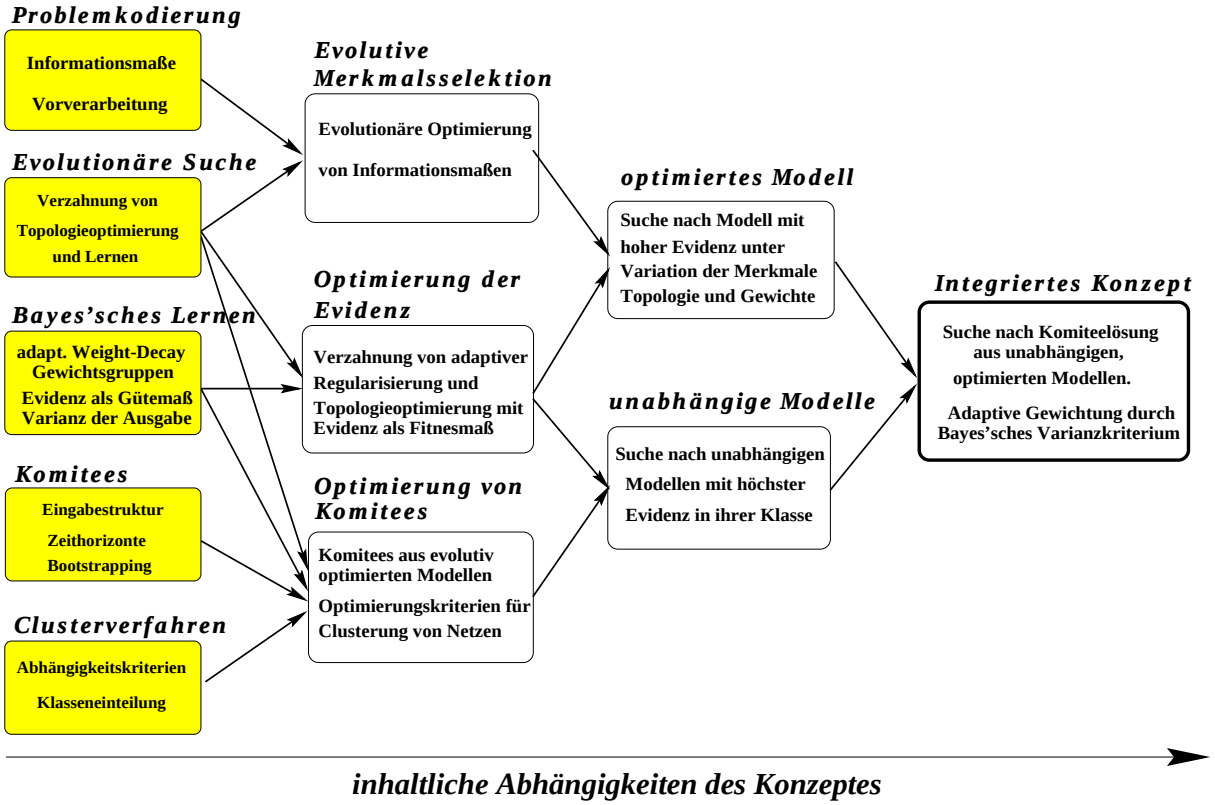


Abbildung 1.5: Die Abbildung verdeutlicht die Schritte ausgehend von den Bausteinen über die Optimierung eines neuronalen Netzes und der Evolution unabhängiger Netze zum Ziel der Arbeit: Problemlösung durch Komitees neuronaler Netze.

Als Modellklasse werden im zweiten Kapitel neuronale Netze eingeführt. Neben der Fehlerfunktion und der eigentlichen Parameteroptimierung wird der Regularisierung viel Raum eingeräumt. Insbesondere das wichtige Konzept der automatischen Regularisierung mit der Bayes'schen Methode wird detailliert behandelt. Das Gütemaß, das beim Bayes'schen Lernen berechnet wird, die sogenannte *Evidenz*, wird später als Fitneßwert im evolutionären Suchverfahren verwendet. Anhand eines Beispiels wird die in der Literatur bisher noch nicht behandelte Problematik beleuchtet, daß die Änderung der Gewichtung des Regularisierungstermes, d.h. die Änderung der Fehlerfunktion, nicht unbedingt mit jedem Gradientenabstiegsverfahren verträglich ist.

Der Titel dieser Arbeit redet von Komitees zur Lösung von Problemen. In Kapitel drei wird der bisherige Forschungsstand dargelegt. Ich werde darauf eingehen, unter welchen Bedingungen Komitees mehr leisten als einzelne Modelle und welche Methoden bisher existieren, um ein Komitee zu entwickeln. Ausgehend von der Dekomposition des Komiteefehlers werde ich in diesem Kapitel ein Kriterium herleiten, um die Zusammensetzung von Komitees zu optimieren. Abschließend leite ich auf einen zentralen Punkt dieser Arbeit hin - die Clusterung von Modellen basierend auf dem hergeleiteten Optimierungskriterium. Das Kriterium berücksichtigt die stochastische Abhängigkeit der Modelle. Da diese anhand der Daten geschätzt werden muß, schließt Kapitel drei mit einer Berechnungsmethode für die

stochastische Abhängigkeit.

Ein wesentliches Ziel dieser Arbeit ist es, ein integriertes Konzept zur Modellentwicklung vorzulegen, das wesentliche Schritte des Entwurfs vereinigt. Der Inhalt von Kapitel 4 ist die Klammer, die die einzelnen Komponenten zusammenhält - evolutionäre Suchverfahren. Ich werde das System ENZO, aus dem diese Arbeit hervorgegangen ist, beschreiben, seine Einschränkungen darlegen, und schließlich die grundlegende Idee dieser Arbeit vorstellen: Wie kann man Information über andere Populationsmitglieder, d.h. neuronale Netze, gezielt nutzen, um Komitees zu evolvieren? Hier werde ich insbesondere darlegen, in welcher Weise die Clusterung von Netzen in den Suchprozeß eingebracht wird. Weiterhin werden die genetischen Operatoren vorgestellt, mittels derer in dieser Arbeit neue Suchpunkte aus den bestehenden generiert werden. Dazu gehört auch der wichtige Aspekt der Parameteroptimierung, der auf die neuen Suchpunkte angewendet wird. Ich werde aufzeigen, wie sich das Bayes'sche Lernen mit einem Suchverfahren effizient kombinieren läßt. Damit sind die Konzepte vorgestellt, die in den folgenden Kapiteln eingehend untersucht werden sollen.

Kapitel 5 beinhaltet die Entwicklung der verbindenen Elemente - Zwischenstationen auf dem Weg zu einem integrierten Konzept.

Zuerst betrachte ich die Evidenz eines Modells als Optimierungskriterium und zeige, daß sich bei festgehaltener Eingabestruktur durch geeignete Verzahnung des Bayes'schen Lernens und evolutionärer Suche ohne höheren Aufwand Modelle mit größerer Evidenz finden lassen. Dieses Verfahren wird dann um die Optimierung der Topologie und des Merkmalsvektors erweitert. Der zweite Baustein dreht sich um die Frage, wie man eine geeignete Kodierung für ein Problem finden und dem 'Fluch der Dimensionen' Rechnung tragen kann. Hierzu integriere ich statistische Kriterien zur Bewertung des Informationsgehaltes (von Kodierungen) in das evolutionäre Suchverfahren. Neben dem künstlichen Problem aus Abbildung 1.1, das ich in vielen Spielarten verwenden werde, um ganz spezielle Aspekte zu beleuchten, werden noch einige Probleme aus Benchmarksammlungen betrachtet (Neal, 1998, Merz & Murphy, 1998, Prechelt, 1994). Abschließend zeige ich mittels der verrauschten Sinusdaten die Grenzen der Modelloptimierung auf, indem verschiedene Parameter, z.B. der Rauschanteil oder die zugrundeliegende Verteilung, variiert werden. Die dort aufgezeigten Probleme sind grundsätzlicher Natur und nicht etwa nur auf neuronale Netze beschränkt.

Nach den Vorarbeiten des fünften Kapitels werde ich mich dann in Kapitel sechs der systematischen Entwicklung unabhängiger Modelle zuwenden. Die grundlegenden Fragen lauten hier, ein Maß der Unabhängigkeit der Modelle in das evolutionäre Suchverfahren zu integrieren. Diese Integration soll auch noch der zweiten Forderung Genüge leisten, daß die Evidenz der Modelle maximiert wird. Die multimodale Suche erlaubt es, weitere sinnvolle Mutationsoperatoren zu definieren, z.B. auf der Trainingsmenge, die für jedes Netz individuell gewählt wird. Für jeden Operator wird der Nachweis geführt, daß er für den Optimierungsprozeß von Nutzen ist. Eine Evolution ist ein fortschreitender Prozeß. Ist es günstiger, wenn die Nachkommen ihre Vorfahren vollständig ersetzen, oder soll die bisherige Lösung die Möglichkeit bekommen, erhalten zu bleiben? Mit der Entscheidung für eine Verzahnung der Suche mit dem Bayes'schen Lernen verbleibt nur die Wahl der zweiten Strategie. Damit verbindet sich auch die Frage, welche Modelle am Ende der Evolution denn nun tatsächlich ausgewählt werden sollen. Idealerweise konvergiert der Algorithmus, wenn er mehrere lokale Optima der Fitneßfunktion gefunden hat. Den Abschluß von Kapitel sechs bildet eine adaptive Ge-

wichtung der Komiteemitglieder mit Hilfe des Bayes'schen Varianzkriteriums. Damit ist das integrierte Konzept zur Entwicklung von Komiteelösungen vollständig.

Kapitel 7 widmet sich der Evaluation des Ansatzes durch eine konkrete Anwendung. Der Schwerpunkt dieses Kapitels liegt vor allem auf der Frage, ob sich mit dem vorgeschlagenen Ansatz signifikante Verbesserungen bei der Modellentwicklung für reale Problemstellungen erzielen lassen. Im Rahmen der Zusammenarbeit mit dem Axel Springer Verlag wurden Modelle für Prognose von Absatzzahlen von Zeitungen erstellt. Da für diese Anwendung mehrere hundert Datensätze vorliegen, läßt sich eindeutig bewerten, ob Verbesserungen der Systemleistung signifikant sind. An dieser Problemstellung, der Prognose von Verkaufszahlen vieler Händler, möchte ich ebenfalls zeigen, daß ein modularer Baukasten sinnvoll ist. Reale Probleme unterliegen oft zusätzlichen Einschränkungen die neben dem primären Optimierungsziel betrachtet werden müssen. Im Falle der Prognose für viele Verkaufsstellen wäre das beispielsweise die Wartbarkeit der Systeme, die wesentlich einfacher ist, wenn nicht für jede Verkaufsstelle eine Speziallösung generiert wird.

Den Abschluß bildet eine Zusammenfassung und Bewertung der Arbeit, mit einer Skizze weiterführender Fragestellungen, die zu betrachten lohnend sein könnte.

Abbildung 1.5 zeigt die inhaltlichen Abhängigkeiten der Arbeit. Die grau hinterlegten Felder zeigen die Grundlagen, auf denen diese Arbeit aufbaut. Die weißen Felder korrespondieren zu Konzepten, die in dieser Arbeit erstmalig vorgeschlagen und zu einem integrierten Optimierungskonzept, das alle grundlegenden Entwurfsentscheidungen einbezieht, ausgestaltet werden. Zusammengefaßt ergaben sich in den folgenden Bereichen Fortschritte:

Modelloptimierung:

- Geeignete Verzahnung des iterativen Bayes'schen Lernens mit einem Suchverfahren
- Evidenz als Optimierungskriterium des Suchverfahrens
- Ersetzen von Suchaufwand durch Vorberechnung geeigneter Schritte mit statistischen Verfahren (z.B. Mutual Information)
- Untersuchung der Kompatibilität von Gradientenabstiegsverfahren mit automatischer Regularisierung

Komiteebildung:

- Herleitung eines Optimierungskriteriums für die Zusammensetzung von Komitees
- Bestimmung der Zusammensetzung eines Komitees durch Klassenbildung auf Basis einer Ähnlichkeitsmatrix (stochastische Abhängigkeit)
- Adaptive Gewichtung der Komiteemitglieder mittels der Varianz der Ausgabe, die das Bayes'sche Verfahren berechnet

Evolutionäre Strategien:

- Multimodale Evolution durch Einbeziehen der stochastischen Abhängigkeiten der Netze, d.h. kein Kollabieren des Algorithmus in einen einzelnen Suchpunkt
- Gleichzeitige Optimierung zweier Kriterien - Evidenz und stochastische Unabhängigkeit - durch den evolutionären Algorithmus
- Integriertes Optimierungskonzept, das fünf wesentliche Teilaufgaben bei der Problemlösung mit neuronalen Netzen behandelt und dadurch Synergieeffekte ausnutzt: Gewichts-, Topologie- und Merkmalsoptimierung sowie Datenabhängigkeit und Modellauswahl.

Kapitel 2

Neuronale Netze und Bayes'sches Lernen

Das Ziel des Lernens ist es, einen funktionalen Zusammenhang auf Basis empirischer Daten implizit in Form eines Modells zu gewinnen. Dazu benötigt man einen Modelltyp, d.h. eine Klasse von Funktionen mit freien Parametern, die es erlauben, sich dem Datensatz anzupassen. Mittels der Daten werden die freien Parameter bestimmt, um den Zusammenhang möglichst gut zu modellieren. In dieser Arbeit werden als Modelltyp vorwärtsgerichtete neuronale Netze (Feed-Forward Netze, Multilayer Perceptrons) verwendet. Die Nomenklatur zur Beschreibung der grundlegenden Konzepte für neuronale Netze folgt der von (Bishop, 1995). Auf die Herleitung und detaillierte Beschreibung der aus der Literatur bekannten Verfahren verzichte ich hier weitestgehend.

Ist für ein gegebenes Problem eine Kodierung der Daten und eine Netztopologie gewählt, dann bedarf es, um die Parameter des neuronalen Netzes einzustellen, der Wahl einer für das Problem geeigneten Fehlerfunktion sowie der Festlegung einer Datenmenge zum Training, weiterhin einer Methodik, den Regularisierungsterm zu gewichten, sowie der Wahl eines Lernverfahrens, das die Parameter des Netzes adaptiert. Diese Punkte werden im folgenden behandelt. Anschließend wird umfassend der Bayes'sche Ansatz vorgestellt, wie er hier verwendet wird, um den Regularisierungsterm automatisch zu gewichten.

2.1 Fehlerfunktionen

Wie bereits ausgeführt, ist das Ziel des Lernens, ein erwartetes Risiko zu minimieren, indem der den Daten zugrundeliegende Prozeß approximiert wird. Diesen Prozeß beschreibt man im allgemeinen durch die gemeinsame Verteilung $p(\mathbf{x}, t) = p(t|\mathbf{x})p(\mathbf{x})$ im Eingabe- und Zielraum. t bezeichnet die Ausgabe und $\mathbf{x}$ die Eingabe. Um für einen neuen, bisher unbekannten Wert von $\mathbf{x}$ die Ausgabe zu schätzen, benötigt man im Anwendungsfall vor allem ein Modell der bedingten Wahrscheinlichkeit $p(t|\mathbf{x})$. Das neuronale Netz soll genau diese Aufgabe leisten. In diesem Fall hängt die bedingte Wahrscheinlichkeit, die das Netz modelliert, von den

Parametern des Netzes ab und wird auch als *Likelihood*-Funktion bezeichnet. Im folgenden betrachte ich die Ausgabe t der einfacheren Schreibweise wegen nur als eindimensional.

Die Likelihood-Funktion für einen Datensatz ergibt sich, wenn die Datenpunkte unabhängig gewählt sind, zu

$$\mathcal{L} = \prod_n p(\mathbf{x}^n, t^n) = \prod_n p(t^n|\mathbf{x}^n)p(\mathbf{x}^n).$$

Um eine Fehlerfunktion zu gewinnen, minimiert man den negativen Logarithmus der Likelihood und geht damit auch von der Produktform zu einer Summenform über:

$$E = -\ln \mathcal{L} = -\sum_n \ln p(t^n|\mathbf{x}^n) - \sum_n \ln p(\mathbf{x}^n).$$

Der zweite Summand $\sum_n \ln p(\mathbf{x}^n)$ ist unabhängig von den Parametern des Modells, d.h. eine Konstante, und kann bei der Minimierung gestrichen werden. Unter verschiedenen Annahmen über die Form der Verteilung $p(t|\mathbf{x})$ gelangt man zu verschiedenen Fehlerfunktionen (Bishop, 1995).

Ausgehend vom Maximium-Likelihood Prinzip und der Annahme von Zielwerten t, die mit normalverteiltem Rauschen behaftet sind, was für Regressionsprobleme eine plausible Annahme ist, erhält man die quadratische Fehlerfunktion

$$E_Q = \frac{1}{2}\sum_n (y(\mathbf{x}^n, \mathbf{w}) - t^n)^2.$$

Dabei steht $y(\mathbf{x}^n, \mathbf{w})$ für die Ausgabe des Modells, d.h. des neuronalen Netzes, während $\mathbf{w}$ der Parametervektor dieses Netzes ist. In Kapitel 2.5.1 wird die Likelihood-Funktion noch detaillierter behandelt. Betrachtet man dagegen Klassifikationsprobleme, dann geht man von unverrauschten oder korrekt gelabelten Daten aus. Wählt man für ein Zwei-Klassen Problem eine 1/0 Kodierung mit $t = 1$, wenn $\mathbf{x}$ zur Klasse 1 gehört und $t = 0$ im anderen Fall, dann muß man eine binäre Ausgabe modellieren, d.h. die Netzausgabe kann als bedingte Wahrscheinlichkeit für $t = 1$ unter der Eingabe x betrachtet werden. Die bedingte Wahrscheinlichkeit für t ergibt sich zu

$$p(t|\mathbf{x}) = y^t(1-y)^{1-t}.$$

Damit erhält man die Cross-Entropy Fehlerfunktion

$$E_{CE} = -\sum_n t^n \ln y^n + (1-t^n)\ln(1-y^n).$$

Dieses Konzept läßt sich auf eine 1-aus-N Kodierung verallgemeinern. Man erhält die sogenannte Multi-Cross-Entropy Fehlerfunktion

$$E_{MCE} = -\sum_n \sum_{k=1}^{N} t_k^n \ln y_k^n.$$

Die Fehlerfunktion wird mittels eines Gradientenabstiegs minimiert. Hierbei wird der *Back-propagation* Algorithmus eingesetzt, um die partiellen Ableitungen der Fehlerfunktion nach den Gewichten des neuronalen Netzes $\frac{\partial E}{\partial w_{ij}}$ zu berechnen (Rumelhart *et al.*, 1986a). Es bleibt anzumerken, daß man zu jeder Fehlerfunktion eine 'natürliche' Aktivierungsfunktion angeben kann, so daß die sogenannten *Fehlersignale* $\delta_k = \frac{\partial E}{\partial a_k}$ der Ausgabeneuronen gerade die einfache Form $y_k - t_k$ annehmen. Die Herleitungen finden sich ausführlich bei (Bishop, 1995). Dabei ist a_k die Aktivierung des k-ten Ausgabeneurons. Die folgende Tabelle stellt die Problemstellungen und die zugehörigen Fehler- bzw. Aktivierungsfunktionen zusammen.

Problemtyp	Fehlerfunktion	Aktivierungsfunktion
Regressionsproblem	Summe der Quadrate	linear
Klassifikation bei 2 Klassen	Cross-Entropy	logistisch
1 aus N Klassifikation	Multi-Cross-Entropy	verallgemeinerte logistische

In den im experimentellen Teil untersuchten Problemstellungen wird jeweils die 'passende' Fehler- und Aktivierungsfunktion verwendet.

2.2 Datenabhängigkeit

Die Auswahl der Daten zum Training ist ein oft unterschätztes Problem bei der Modellentwicklung. Eine tiefere Einsicht kann man durch die Betrachtung des Fehlers im Falle unendlich vieler Daten gewinnen. Im folgenden verwende ich der Einfachheit halber den quadratischen Fehler. Aus dem durchschnittlichen Fehler pro Muster

$$\mathrm{E}_{\mathrm{Train}} = \frac{1}{2N} \sum_{n=1}^{N} \left(y(\mathbf{x}^n, \mathbf{w}) - t^n\right)^2 \tag{2.1}$$

gewinnt man durch Grenzübergang für $N \to \infty$ den Generalisierungsfehler

$$\mathrm{E}_{\mathrm{Gen}} = \frac{1}{2} \int \int \left(y(\mathbf{x}, \mathbf{w}) - t\right)^2 \; p(t, \mathbf{x}) \; dt d\mathbf{x}. \tag{2.2}$$

Mit den bedingten Erwartungswerten von t,

$$\mathcal{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x}) \, dt \qquad \text{bzw.}$$

$$\mathcal{E}[t^2|\mathbf{x}] = \int t^2 p(t|\mathbf{x}) \, dt,$$

und $p(t, \mathbf{x}) = p(t|\mathbf{x})p(\mathbf{x})$ sowie $y(\mathbf{x}, \mathbf{w}) - t = y(\mathbf{x}, \mathbf{w}) - \mathcal{E}[t|\mathbf{x}] + \mathcal{E}[t|\mathbf{x}] - t$ gewinnt man die Form

$$\begin{aligned} \mathrm{E}_{\mathrm{Gen}}(w) &= \frac{1}{2}\int \left(y(\mathbf{x},\mathbf{w}) - \mathcal{E}[t|\mathbf{x}] \right)^2 p(\mathbf{x})\, d\mathbf{x} \\ &\quad + \frac{1}{2}\int \Big(\mathcal{E}[t^2|\mathbf{x}] - \mathcal{E}[t|\mathbf{x}] \Big)^2 p(\mathbf{x})\, d\mathbf{x} \end{aligned} \tag{2.3}$$

Der zweite Term ist unabhängig vom neuronalen Netz. In der Praxis hängt der Fehler des neuronalen Netzes $y(\mathbf{x},\mathbf{w})$ vom endlichen Datensatz D ab, mit dem es trainiert wurde. Um diese Abhängigkeit zu eliminieren, kann man über alle möglichen N-elementigen Datensätze mitteln, d.h. wir betrachten statt des ersten Integranden den Erwartungswert

$$\mathcal{E}_D \left[\left(y(\mathbf{x},\mathbf{w}) - \mathcal{E}[t|\mathbf{x}] \right)^2 \right].$$

Damit ergibt sich für den Erwartungswert des ersten Integranden die Darstellung

$$\begin{aligned} &\mathcal{E}_D \left[\left(y(\mathbf{x},\mathbf{w}) - \mathcal{E}[t|\mathbf{x}]\right)^2 \right] = \\ &\quad \left(\mathcal{E}_D[y(\mathbf{x},\mathbf{w})] - \mathcal{E}[t|\mathbf{x}]\right)^2 + \mathcal{E}_D \left[\left(y(\mathbf{x},\mathbf{w}) - \mathcal{E}_D[y(\mathbf{x},\mathbf{w})]\right)^2 \right]. \end{aligned} \tag{2.4}$$

Gleichung 2.4 zusammen mit Gleichung 2.3 bezeichnet man als die Bias-Varianz Dekomposition des Fehlers (Geman *et al.*, 1992, Bishop, 1995):

$$\begin{aligned} (\text{Bias})^2 &= \frac{1}{2}\int \left(\mathcal{E}_D[y(\mathbf{x},\mathbf{w})] - \mathcal{E}[t|\mathbf{x}]\right)^2 p(\mathbf{x})\, d(\mathbf{x}) \\ \text{Varianz} &= \frac{1}{2}\int \mathcal{E}_D \left[\left(y(\mathbf{x},\mathbf{w}) - \mathcal{E}_D[y(\mathbf{x},\mathbf{w})]\right)^2 \right] p(\mathbf{x})\, d(\mathbf{x}). \end{aligned} \tag{2.5}$$

Die Bedeutung der Bias-Varianz Dekomposition erschließt sich, wenn man die beiden Extremfälle betrachtet: Im einen Fall wählt man als Modell eine feste Funktion $g(\mathbf{x})$ völlig unabhängig von den Daten. Da es keine Flexibilität besitzt, um sich den Daten anzupassen, hat es einen hohen Bias. Im anderen Fall wählt man eine Funktion, die die Daten perfekt wiedergibt. Die Daten werden also 'auswendig' gelernt, auch ein eventueller Rauschanteil. Das Modell hat eine hohe Varianz, da das Hinzufügen oder Entfernen von Datenpunkten dann zu einem gänzlich anderen Modell führt. Abbildung 2.1 illustriert das Gesagte.

Die beiden Größen verhalten sich in natürlicher Weise komplementär. Funktionen, die die Trainingsdaten perfekt abbilden, haben in der Regel eine hohe Varianz und damit einen hohen Generalisierungsfehler. Regularisiert man die Funktion, sinkt die Varianz. Ist die Regularisierung allerdings zu stark, dann wird der Bias und damit auch wieder der Generalisierungsfehler groß.

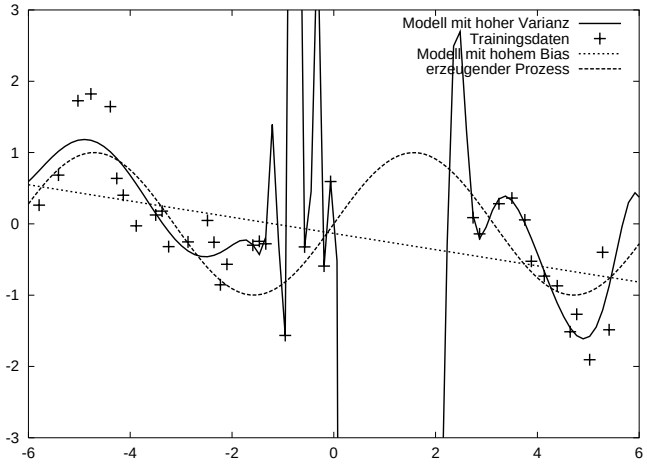


Abbildung 2.1: Das Bias-Varianz-Dilemma illustriert am Beispiel aus Abbildung 1.1b: Ein Modell mit wenig Freiheitsgraden ist zu unflexibel und hat einen hohen Bias, ein Modell mit vielen Freiheitsgraden hat eine hohe Varianz und lernt die Trainingsdaten nahezu 'auswendig'. Insbesondere in Bereichen, in denen wenig Daten vorhanden sind, weicht die gelernte Funktion im letzten Fall stark von der gesuchten ab.

Die hier vorgestellte Dekomposition geht von dem quadratischen Fehler aus. Daher wäre es wünschenswert, eine Bias-Varianz Dekomposition zu entwickeln, die kein bestimmtes Fehlermaß vorausetzt. Dazu stellte jüngst Heskes einen schönen Ansatz vor, der auf der Kullback-Leibler Distanz zwischen der tatsächlichen Verteilung der Zielwerte und einer Schätzung dieser Verteilung basiert. Damit läßt sich der Fehler allgemein in zwei Terme, Bias und Varianz, aufspalten. Der Bias entspricht der Kullback-Leibler Distanz zwischen der Zielverteilung und einem Mittelwert aller Schätzer, die Varianz der Distanz zwischen dem Mittelwert und dem gegebenen Schätzer (Heskes, 1998).

In der Praxis kann man die Problematik der Datenabhängigkeit dadurch abmildern, daß man aus einer einmal festgelegten Trainingsmenge mehrere Teilmengen bildet, mit denen die Netze trainiert werden. Dies ist insbesondere dann wichtig, wenn für ein Problem nur wenig Daten erhältlich und diese eventuell noch mit starkem Rauschen behaftet sind.

2.3 Regularisierung

Die Theorie der Regularisierung besagt, wenn man aus empirischen Daten einen funktionalen Zusammenhang lernen möchte, daß man den regularisierten Fehler minimieren sollte, um eine geeignete Lösung zu erhalten. Für die Entwicklung neuronaler Netze addiert man zu einem auf den Daten gemessenen Fehler E_D einen weiteren Term E_R, der die Komplexität des Modells mißt:

$$E = E_D + \lambda E_R. \tag{2.6}$$

E_D steht dabei für eine der obigen Fehlerfunktionen. Der Gewichtsvektor $\mathbf{w}$ wird nun bei festgehaltenem λ durch das Training so bestimmt, daß die Fehlerfunktion E minimal wird. Das neuronale Netz muß nun einerseits die Trainingsdaten lernen, andererseits aber auch eine möglichst geringe Komplexität E_R aufweisen. Der für neuronale Netze am häufigsten

verwendete Regularisierungsterm ist die Summe über die quadratischen Gewichte, der sogenannte *Weight-Decay* (Hertz *et al.*, 1991, Bishop, 1995):

$$E_R = \frac{1}{2} \sum_i w_i^2.$$

Ein geeignetes λ bestimmt man experimentell durch Kreuzvalidierung, d.h. man beobachtet den Fehler auf einer genügend großen Menge von Daten, die man von der Trainingsmenge abtrennt, und vergrößert λ ausgehend von 0 so lange, bis der Overfitting-Effekt gerade noch unterdrückt wird. Die Güte des Ergebnisses hängt von der speziellen Wahl der Kreuzvalidierungsmenge ebenso ab, wie von der Gewichtsinitialisierung und der Feinheit des Rasters, mit dem λ vergrößert wird.

Mittels des Bayes'schen Theorems läßt sich ein Verfahren herleiten, das die Regularisierung automatisch und adaptiv während des Trainings berechnet. Insbesondere wird dazu keine zusätzliche Validierungsmenge benutzt, so daß mehr Daten zum Training und Testen zur Verfügung stehen.

2.4 Parameteroptimierung

Der Erfolg des Backpropagation beruhte darauf, daß man einen effizienten Algorithmus angeben konnte, um die Ableitung der Fehlerfunktion nach den Gewichten in einem mehrschichtigen Netz zu berechnen, um einen Gradientenabstieg für die Fehlerfunktion zu realisieren (Rumelhart *et al.*, 1986a, Bishop, 1995). Die Rückwärtspropagierung des Fehlers ist zu unterscheiden von der Anpassung der Gewichte während des Gradientenabstiegs. Die Parameteroptimierung besteht im wesentlichen aus zwei Teilen: Zuerst wird die Richtung bestimmt, in der das Minimum gesucht wird, dann wird die Länge des Schrittes in diese Richtung, d.h. die Gewichtsänderung, berechnet.

Die einfachste Variante des Gradientenabstiegs, nämlich sich nur nach der Größe des Gradienten zu richten, ist in der Literatur ebenfalls als Backpropagation bekannt. Das Verfahren findet bereits für Fehlerfunktionen, die in Abhängigkeit der Gewichte quadratisch sind, das Minimum nicht, wenn die Länge des Gradienten (skaliert mit der Lernrate) größer ist als die Entfernung zu dem Punkt im Gewichtsraum, der in Gradientenrichtung auf der selben Höhenlinie liegt (vgl. Abbildung 2.2).

Der wesentliche Nachteil des einfachen Verfahrens besteht darin, daß es beim aktuellen Schritt keine Information verwendet, die in vorhergehenden Schritten gewonnen wurde. Ein Gedächtnis dieser Art wird z.B. von den sogenannten Kongugierten Gradienten Verfahren (Bishop, 1995, Fletcher, 1995), insbesondere *Scaled Conjugate Gradient* (Møller, 1993), oder auch dem hier am Institut entwickelten Verfahren *Rprop* verwendet (Riedmiller, 1994).

Scaled Conjugate Gradient (SCG) versucht die Fehlerfunktion als quadratisch anzunehmen und springt in das Minimum dieses Paraboloid in der vorher festgelegten Suchrichtung. Ist an diesem Punkt der so vorhergesagte Fehler (mittels des Paraboloids) nahe am tatsächlichen Fehler, dann paßt die quadratische Annahme. Ist der Unterschied zu groß, verwirft man die Annahme. Im ersten Fall befindet man sich nun nahe bei einem Minimum und verkleinert

die Schrittweite um einen konstanten Faktor (z.B. auf die Hälfte). Im zweiten Fall befindet man sich weit weg, dann vergrößert man die Schrittweite.

Wählt man als nächste Suchrichtung wieder den negativen Gradienten, dann liegt diese senkrecht zur vorherigen Richtung, wenn man im vorigen Schritt genau das Minimum gefunden hat. Im zweidimensionalen Fall läuft man also einen Zick-Zack Weg (mit rechten Winkeln) ab. Das Verfahren der konjugierten Gradienten versucht dies zu vermeiden. Die neue Richtung wird senkrecht zu allen bisherigen Richtungen gewählt, so daß das im vorigen Abschnitt erhaltene Minimum nicht wieder verlassen wird.

Die Idee des *Rprop* Lernalgorithmus basiert darauf, jedes Gewicht für sich zu betrachten und für jedes Gewicht eine individuelle Schrittweite zu bestimmen. Dazu verwendet man nur noch die Information, ob man im vorigen Schritt in dieselbe Richtung gegangen ist. Der Betrag des Gradienten wird nicht mehr verwendet. Ist die Richtung dieselbe, dann wird die Schrittweite für dieses Gewicht vergrößert (es wird beschleunigt), ist die Richtung entgegengesetzt, dann wird die Schrittweite verkleinert (es wird abgebremst). Beschleunigung und Abbremsung werden jeweils durch Multiplikation eines Faktors > 1 bzw. < 1 erreicht.

Abbildung 2.2 zeigt für ein Netz mit zwei Gewichten zu einem linearen Regressionsproblem den Weg der drei Verfahren zum Minimum. Die Fehlerfunktion ist durch Höhenlinien im Gewichtsraum dargestellt. Ausgehend von dem Startpunkt entfernt sich Backpropagation immer weiter vom Minimum. SCG findet das Minimum in wenigen Schritten, da die Annahme einer quadratischen Funktion genau zutrifft. Für *Rprop* ist die Beschleunigungsphase am Anfang und kurz vor Erreichen des Minimums typisch. Diese Eigenschaft verlangsamt für den gezeigten Fall die Konvergenz gegenüber SCG, macht das Verfahren aber robust im Falle, daß die Fehlerfunktion von einer quadratischen Form abweicht oder daß sich die Fehlerfunktion ändert.

Verändert man den Gewichtungsfaktors λ des Strafterms während des Lernens, dann birgt das eine in der Literatur bisher noch nicht behandelte Problematik: Die Fehlerfunktion, deren Minimum man sucht, ändert sich bei jeder neuen Schätzung des Faktors. Es ist nicht offensichtlich, ob und wieweit ein Lernverfahren mit einer automatischen Gewichtung kompatibel ist, wie sie durch das Bayes'sche Verfahren realisert wird. Anhand der beiden Verfahren, *Scaled Conjugate Gradient* und *Rprop*, wird diese Problematik nach der Einführung des Bayes'schen Lernens nochmals beleuchtet.

2.5 Bayes'sches Lernen

Die Arbeiten zur Anwendung des Bayes'schen Lernens bei neuronalen Netzen gehen im wesentlichen auf die Dissertation von MacKay zurück (MacKay, 1992). Eine umfassende Diskussion findet man vor allem in (Bishop, 1995) und der Dissertation von Gutjahr (Gutjahr, 1999). Im Rahmen dieser Theorie läßt sich der Parameter λ stochastisch interpretieren und ein Optimierungsverfahren für λ angeben. Bei 'empirischen' Regularisierungsverfahren wird der Gewichtungsfaktor λ mittels Kreuzvalidierung eingestellt. Dadurch ergeben sich einige Nachteile:

- Es stehen weniger Daten zum Training zur Verfügung.

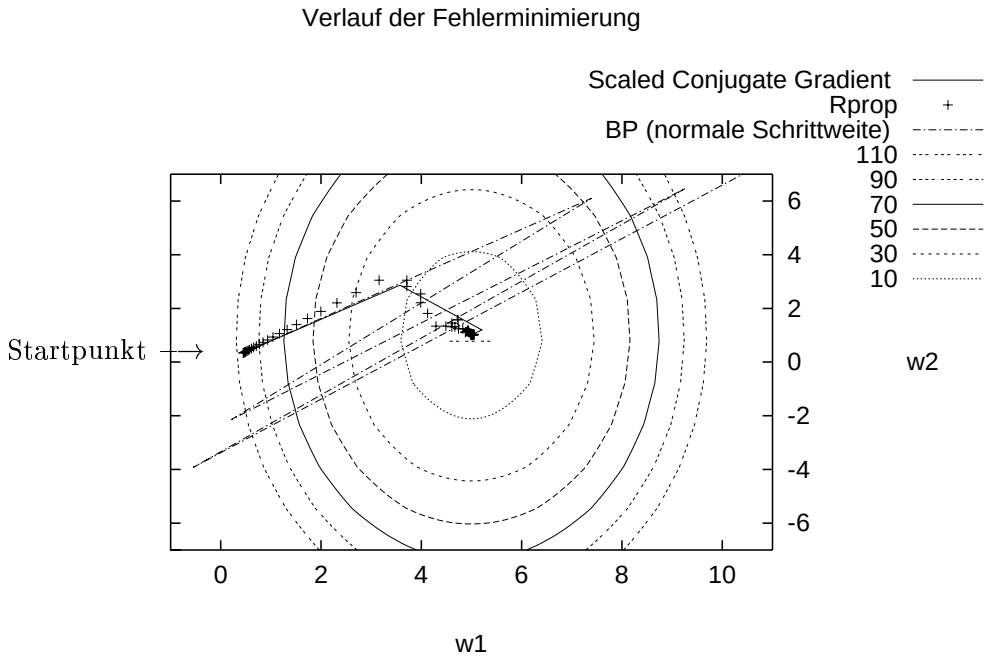


Abbildung 2.2: Verlauf des Gradientenabstiegs im Gewichtsraum bei einer zweidimensionalen, parabelförmigen Fehlerfunktion für Backpropagation, *Scaled Conjugate Gradient* und *Rprop* ausgehend vom selben Startpunkt links im Bild. Der Weg für *Rprop* ist durch einzelne Kreuze gekennzeichnet, für SCG durch die durchgezogene Linie, die die drei Schritte bis zum Minimum verbindet. Die gestrichelte Zick-Zack-Linie zeigt den Weg für Backpropagation, der nicht zum Minimum führt. Die Fehlerfunktion ist mittels Höhenlinien dargestellt. Die gezeigte Fehlerfunktion erfüllt genau die Annahmen für die quadratische Approximation des SCG-Verfahrens. Im allgemeinen sind diese nicht erfüllt. Die für Rprop typische Beschleunigungs- und Abbremsphase ist deshalb oft von Vorteil, da die Fehlerfunktion dabei beliebige Form haben kann.

- Fügt man die Validierungsmenge der Trainingsmenge hinzu, um eine möglichst große Menge zu erhalten, dann ist der empirisch ermittelte Wert ungenau.
- Die Parameter müssen individuell für jede Topologie und Datenaufteilung bestimmt werden. Dies ist insbesondere dann problematisch, wenn die Topologie, z.B. der Eingabevektor, ebenfalls optimiert werden soll.
- Die Werte hängen stark von der Validierungsmenge ab. Dies kann insbesondere dann zu Problemen führen, wenn die Menge klein ist oder wenn die Daten stark verrauscht sind.

Die Bayes'sche Regel wird für das Training neuronaler Netze in der Weise angewendet, daß man sein Wissen über das Problem als a-priori Annahme formuliert, z.B. eine spezielle

Verteilung der Gewichte $p(\mathbf{w})$ annimmt, und mittels der Regel von Bayes nach Beobachtung der Daten (Training) in eine a-posteriori Verteilung überführt:

$$p(\mathbf{w}|D) = \frac{p(D|\mathbf{w}) \cdot p(\mathbf{w})}{p(D).} \tag{2.7}$$

Die bedingte Wahrscheinlichkeit $p(D|\mathbf{w})$ ist der datenabhängige Term in der Formel, den man gewöhnlich als Likelihood-Funktion bezeichnet. Er gibt dabei an, inwieweit das Modell mit Gewichtsvektor $\mathbf{w}$ die Daten erklärt, d.h. es wird die Wahrscheinlichkeit geschätzt, daß die Ausgabedaten $\mathbf{y}$ bei Eingabe $\mathbf{x}$ von dem Modell generiert wurden. Dadurch trägt man der Tatsache Rechnung, daß man durch das Training Information darüber gelernt hat, inwiefern ein bestimmtes $\mathbf{w}$ mit den Daten konsistent ist. Um die a-posteriori Wahrscheinlichkeit auszuwerten, muß man die Annahmen für die *a-priori Verteilung* und die Likelihood-Funktion konkretisieren.

Im folgenden werden drei Stufen des Bayes'schen Ansatzes durchlaufen: Zuerst sucht man nach einem optimalen Gewichtsvektor. Um diesen für eine variable Gewichtung von E_R zu gewinnen, optimiert man auf der zweiten Stufe diese Gewichtung des Regularisierungstermes. Abschließend vergleicht man noch verschiedene Topologien miteinander, um ein geeignetes Modell zu erhalten. Auf einer Stufe werden immer zwei Aspekte betrachtet: Einerseits benötigt man ein Verfahren, um die Parameter zu optimieren, die die a-posteriori Wahrscheinlichkeit maximieren. Andererseits möchte man auch eine Approximation dieser a-posteriori Verteilung haben, um mit dieser arbeiten zu können.

2.5.1 Optimierung des Gewichtsvektors

Im Gegensatz zum Maximum-Likelihood Ansatz, bei dem eine Fehlerfunktion minimiert wird und als Resultat ein Gewichtsvektor geliefert wird, verwendet der Bayes'sche Ansatz eine Wahrscheinlichkeitsverteilung über dem Gewichtsraum, die wiedergibt, inwieweit eine bestimmte Gewichtseinstellung dem gesuchten Modell entspricht. Diese Funktion entspricht zu Beginn einer a-priori Annahme, die dann nach dem Training, d.h. nach der Beobachtung der Daten, in eine a-posteriori Verteilung überführt wird. Diese Verteilung benötigt man auf der nächsten Stufe.

A-priori Wahrscheinlichkeit für die Gewichte

Für die a-priori Verteilung wählt man im einfachsten Fall eine Normalverteilung um 0 mit Varianz $1/\alpha$:

$$p(w_q) = \sqrt{\alpha}\varphi(\sqrt{\alpha}w_q).$$

Dabei bezeichnet $\varphi(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}$ die Dichte der Standardnormalverteilung und w_q die q-te Komponente des Gewichtsvektors. Über den sogenannten *Hyperparameter* α wird direkt die Komplexität des Modells gesteuert. Ein großes α sorgt dafür, daß sich die Gewichte nahe

um Null konzentrieren, das Modell also eine geringe Komplexität hat. Die Verteilung $p(\mathbf{w})$ ergibt sich unter der Annahme, daß die Gewichte voneinander unabhängig sind, als Produkt der Verteilungen für die einzelnen Gewichte w_q

$$\begin{aligned} p(\mathbf{w}) &= \prod_{q=1}^{W} p(w_q) &= \prod_{q=1}^{W} \sqrt{\frac{\alpha}{2\pi}} e^{-\frac{\alpha}{2}(w_q)^2} \\ &&= (\frac{\alpha}{2\pi})^{\frac{W}{2}} e^{-\frac{\alpha}{2}\sum_{q=1}^{W} w_q^2} \\ &&= (\frac{\alpha}{2\pi})^{\frac{W}{2}} e^{-\alpha E_W}. \end{aligned} \tag{2.8}$$

Mit der obigen Verteilungsannahme setzt man voraus, daß die Muster ebenso symmetrisch um Null skaliert sind. In (Nautze & Gutjahr, 1997, Gutjahr, 1999) werden die Annahmen dahingehend abgeschwächt, daß der Mittelwert der Verteilung auch als Parameter aufgefaßt und dann durch das Training optimiert wird. Ebenso ist die Annahme der Unabhängigkeit der Gewichte nicht ganz zutreffend. Bei der empirisch begründeten Verwendung des Weight-Decays macht man diese Annahmen implizit. Die Einbettung in eine Theorie zwingt also dazu, sich der Voraussetzungen bewußt zu werden und auch die Grenzen der Verfahren zu erkennen.

Likelihood Funktion

Der erste Term aus der Bayes'schen Regel 2.7 hängt von dem auf den Daten gemessenen Fehler ab. In die spezielle Form der Likelihood Funktion fließt die gewählte Fehlerfunktion mit ein. Ich beschränke mich auf den interessanteren Fall der Regressionsprobleme. Für Klassifikationsprobleme entfallen einige der im folgenden gemachten Annahmen, da in den Zielwerten keine Unsicherheit enthalten ist, d.h. im eindimensionalen Fall sind sie entweder 0 oder 1.

Über die Verteilung der Zielwerte wird die einfache Annahme gemacht, daß sie von einer stetigen Funktion generiert wurden und mit normalverteiltem Rauschen mit Varianz $1/\beta$ behaftet sind. d.h.

$$t = h(\mathbf{x}) + \epsilon.$$

Die Verteilung des Fehlers ist dann durch

$$p(\epsilon) = \sqrt{\beta}\varphi(\sqrt{\beta}\epsilon)$$

gegeben. Identifiziert man $h(\mathbf{x})$ mit einem neuronalen Netz $y(\mathbf{x}, \mathbf{w})$, dann erhält man mit den beiden vorangehenden Gleichungen die Wahrscheinlichkeit den (tatsächlichen) Zielwert t bei Eingabe $\mathbf{x}$ zu schätzen:

$$\begin{aligned} p(t|\mathbf{x},\mathbf{w}) &= \sqrt{\beta}\varphi\left(\sqrt{\beta}(y(\mathbf{x},\mathbf{w})-t)\right) \\ &= \frac{1}{C_\beta} e^{-\frac{\beta}{2}(y(\mathbf{x},\mathbf{w})-t)^2}. \end{aligned} \tag{2.9}$$

Für den Likelihood Term ergibt sich damit:

$$\begin{aligned} p(D|\mathbf{w}) &= \prod_{n=1}^{N} p(t^n|\mathbf{x}^n,\mathbf{w}) \\ &= (\frac{\beta}{2\pi})^{\frac{N}{2}} e^{-\frac{\beta}{2}\sum_{n=1}^{N}(y(\mathbf{x}^n,\mathbf{w})-t^n)^2} \\ &= \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} e^{-\beta E_D}. \end{aligned} \tag{2.10}$$

A-posteriori Wahrscheinlichhkeit für die Gewichte

Mit der Wahl eines Priors, Gleichung (2.8), und einer Likelihood Funktion, Gleichung (2.10), hat man die Vorraussetzungen geschaffen, die a-posteriori Verteilung der Gewichte mittels der Regel von Bayes zu bestimmen:

$$\begin{aligned} p(\mathbf{w}|D) &= \frac{1}{C_\alpha}\frac{1}{C_\beta}\frac{p(\mathbf{D}|\mathbf{w})p(\mathbf{w})}{p(D)} \\ &= \frac{1}{p(D)C_\alpha C_\beta} e^{-\beta E_D - \alpha E_W} \\ &= \frac{1}{C_S} e^{-S(\mathbf{w})}. \end{aligned} \tag{2.11}$$

$S(\mathbf{w})$ ist also eine spezielle Ausprägung der allgemeinen Fehlerfunktion aus (2.6). Der optimale Gewichtsvektor maximiert den Wert der a-posteriori Verteilung. Da der Normalisierungsterm $C_S = p(D)C_\alpha C_\beta$ nicht von den Gewichten abhängt, findet man diesen Gewichtsvektor, indem man den negativen Exponenten minimiert:

$$\beta E_D + \alpha E_W = \frac{\beta}{2}\sum_{n=1}^{p}(y(\mathbf{x}^n,\mathbf{w})-y^n)^2 + \frac{\alpha}{2}\sum_{q=1}^{N} w_q^2 \quad \longrightarrow \quad min. \tag{2.12}$$

Mit $\lambda = \frac{\alpha}{\beta}$ entspricht die Minimierung dieser Funktion dem Lernen mit Weight-Decay. Die stochastische Interpretation der Parameter ermöglicht es jetzt aber einen Algorithmus zur Optimierung dieser Parameter zu entwickeln, anstatt diese empirisch zu bestimmen.

Taylor-Approximation der a-posteriori Verteilung:

Obwohl der Ausdruck für die a-posteriori Verteilung in Gleichung (2.11) exakt ist, kann die Normalisierungskonstante im allgemeinen nicht analytisch berechnet werden (Bishop, 1995). Um die Wahrscheinlichkeitsverteilung der Netzausgaben berechnen zu können, verwendet MacKay eine Approximation durch eine Gaußfunktion für die a-posteriori Verteilung (MacKay, 1992). Mittels der Taylorapproximation um das Minimum $S(\mathbf{w}_{min})$ erhält man für $S(\mathbf{w})$ unter Beachtung, daß der Term erster Ordnung gerade verschwindet,

$$S(\mathbf{w}) = S(\mathbf{w}_{min}) + \frac{1}{2}(\mathbf{w} - \mathbf{w}_{min})^T \mathbf{A} (\mathbf{w} - \mathbf{w}_{min}). \tag{2.13}$$

Die Matrix $\mathbf{A}$ ist die Hessematrix der regularisierten Fehlerfunktion $S(\mathbf{w})$ an der Stelle $\mathbf{w}_{min}$:

$$\mathbf{A} = \nabla\nabla S(\mathbf{w}_{min}) = \beta \nabla\nabla E_D(\mathbf{w}_{min}) + \alpha \mathbf{I} = \mathbf{H} + \alpha \mathbf{I}. \tag{2.14}$$

Somit bekommt man für die a-posteriori Verteilung aus (2.11), die jetzt eine Gaußfunktion ist:

$$\begin{aligned} p(\mathbf{w}|D) &\approx \frac{1}{C_S} e^{-S(\mathbf{w})} \\ &\approx \frac{1}{C_S} e^{-S(\mathbf{w}_{min}) - \frac{1}{2}(\Delta\mathbf{w})^T \mathbf{A} (\Delta\mathbf{w})} \\ &= \frac{e^{(-S(\mathbf{w}_{min}))}}{C_S} e^{-\frac{1}{2}(\Delta\mathbf{w})^T \mathbf{A} (\Delta\mathbf{w})}. \end{aligned} \tag{2.15}$$

Der Normalisierungsfaktor C_S ergibt sich durch Integration über den Gewichtsraum, die ich hier nicht ausführen möchte (siehe (Bishop, 1995)), zu:

$$C_S = \int e^{-S(\mathbf{w})} d\mathbf{w} = (2\pi)^{W/2} \det(\mathbf{A})^{-1/2} e^{-S(\mathbf{w}_{min})}. \tag{2.16}$$

Die Berechnung des Integrals ist bei neuronalen Netzen analytisch nicht möglich, was die Vereinfachung durch die Taylor-Approximation notwendig macht. Die ursprünglich gewonnene exakte Verteilung 2.11 wurde also durch eine Normalverteilung angenähert, die ihren wahrscheinlichsten Wert im Minimum der Fehlerfunktion $S(\mathbf{w})$ hat.

2.5.2 Der Evidenz-Ansatz zur Optimierung von α und β

Der Evidenz-Ansatz ist ein hierarchisches Optimierungskonzept. Auf der untersten Stufe wurde die Verteilung von Gewichten betrachtet, um einen optimalen Gewichtsvektor zu finden. Auf dieser zweiten Ebene werden Verteilungen von Hyperparametern, α und β, gesucht. Im folgenden wird die Topologie des Netzes als gegeben vorausgesetzt. Die dritte Stufe wird dann abschließend die Evidenz eines neuronalen Modells mit der Topologie als Parameter betrachten.

Optimierung von α und β

Um 'vernünftige' Werte für α und β zu finden, wertet man wieder die a-posteriori Verteilung von α und β aus.

$$p(\alpha, \beta|D) = \frac{p(D|\alpha, \beta)p(\alpha, \beta)}{p(D)}. \tag{2.17}$$

Die a-priori Verteilung für α und β wählt man so, daß sie keinen Wert bevorzugt. Man nennt solche Verteilungen auch *nicht-informativ*. Da die Wahrscheinlichkeitsmasse in diesem Fall unendlich ist, nennt man solche a-priori Verteilungen auch *uneigentlich* (engl.: *improper priors*). Sie sind in der Statistik eine bekannte Methodik, siehe z.B. (Carlin & Louis, 1996, Bishop, 1995).

Die optimalen Werte für α und β findet man nach diesen Voraussetzungen, indem man $p(D|\alpha, \beta)$ maximiert. Dieser Ausdruck wird auch als *Evidenz* für α und β bezeichnet. Die Methode die Hyperparameter so zu bestimmen, heißt deshalb auch Evidenz-Ansatz. Durch Integration über dem Gewichtsraum ergibt sich, wenn man beachtet, daß der Prior für die Gewichte von β und die Likelihood von α unabhängig ist:

$$\begin{aligned} p(D|\alpha, \beta) &= \int p(D|\mathbf{w}, \alpha, \beta)p(\mathbf{w}|\alpha, \beta)d\mathbf{w} \\ &= \int p(D|\mathbf{w}, \beta)p(\mathbf{w}|\alpha)d\mathbf{w} \\ &= \frac{1}{C_\alpha}\frac{1}{C_\beta}\int e^{-\beta E_D - \alpha E_W}\, d\mathbf{w} \\ &= \frac{1}{C_\alpha}\frac{1}{C_\beta}\int e^{-S(\mathbf{w})}. \end{aligned}$$

Das Integral entspricht gerade dem Term C_S aus Gleichung 2.16. Man maximiert nun den Logarithmus der Evidenz und erhält durch partielles Ableiten damit die gesuchten Formeln zur iterativen Berechnung von α und β:

$$\begin{aligned} \ln p(D|\alpha, \beta) &= -\alpha E_W(\mathbf{w}_{min}) - \beta E_D(\mathbf{w}_{min}) - \frac{1}{2}\ln|\mathbf{A}| \\ &\quad + \frac{W}{2}\ln\alpha + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) \qquad (2.18) \\ &\rightarrow \quad \text{max.} \end{aligned}$$

Um an dieser Stelle weiterzurechnen, vernachlässigt man die Abhängigkeit des Minimums $\mathbf{w}_{min}$ von α und β und reduziert diesen Fehler später durch eine iterative Approximation.

Auswertung für α: Sei $\mathbf{H}$ die Hessematrix der unregularisierten Fehlerfunktion E_D und λ_i deren Eigenwerte. Dann ergibt partielles Differenzieren nach α und Nullsetzen

$$\begin{aligned} 0 &\stackrel{!}{=} \frac{\partial \ln p(D|\alpha,\beta)}{\partial\alpha} \\ &= -E_W - \frac{\partial}{\partial\alpha}\frac{1}{2}\ln|\mathbf{A}| + \frac{W}{2\alpha} \\ &= -E_W - \frac{\partial}{\partial\alpha}\frac{1}{2}\ln(\prod_i(\lambda_i+\alpha)) + \frac{W}{2\alpha} \\ &= -E_W - \frac{\partial}{\partial\alpha}\frac{1}{2}\sum_i \ln(\lambda_i+\alpha) + \frac{W}{2\alpha} \\ &= -E_W - \frac{1}{2}\sum_i^W \frac{1}{\lambda_i+\alpha} + \frac{W}{2\alpha}. \end{aligned} \tag{2.19}$$

Auswertung für β: Für die Eigenwerte gilt $d\lambda_i/d\beta = \lambda_i/\beta$, d.h., sie sind proportional zu β (Bishop, 1995). Partielles Differenzieren nach β und Nullsetzen ergibt

$$\begin{aligned} 0 &\stackrel{!}{=} \frac{\partial \ln p(D|\alpha,\beta)}{\partial\beta} \\ &= -E_D - \frac{\partial}{\partial\beta}\frac{1}{2}\sum_i \ln(\lambda_i+\alpha) + \frac{N}{2\beta} \\ &= -E_D - \frac{1}{2\beta}\sum_i \frac{\lambda_i}{\lambda_i+\alpha} + \frac{N}{2\beta}. \end{aligned} \tag{2.20}$$

Zusammenfassend ergibt sich für α und β

$$2\alpha E_W = \gamma \tag{2.21}$$

$$2\beta E_D = N - \gamma\,, \tag{2.22}$$

dabei wurde

$$\gamma = \sum_{i=1}^W \frac{\lambda_i}{\lambda_i+\alpha} = W - \sum_i^W \frac{\alpha}{\lambda_i+\alpha} \tag{2.23}$$

verwendet, und die Eigenwerte der Hessematrix **H** der nicht regularisierten Fehlerfunktion E_D wurden mit λ_i bezeichnet. Die Größe γ bezeichnet man auch als Anzahl der effektiven Parameter. Ist die Topologie des Netzes wesentlich größer als für die Aufgabe erforderlich, dann werden viele Eigenwerte der Hessematrix fast Null. Das spielt später eine Rolle, wenn wir die Evidenz eines Modells betrachten. Auf die Topologieoptimierung gehe ich in Kapitel 4.1 bei der Frage, wie man gute Modelle findet, nochmals genauer ein.

Iterative Berechnung der Hyperparameter In der praktischen Anwendung des Ansatzes muß man nun sowohl die optimalen Werte für α und β als auch für $\mathbf{w}$ bestimmen. Dies löst man durch einen iterativen Algorithmus, indem man zuerst ein Minimum der Fehlerfunktion bezüglich $\mathbf{w}$ sucht und dann α und β neu bestimmt gemäß

$$\alpha^{neu} = \frac{\gamma}{2E_W} \qquad \text{und}$$

$$\beta^{neu} = \frac{N-\gamma}{2E_D}.$$

Die beiden Schritte werden abwechselnd wiederholt bis zur Konvergenz von α und β. Damit ergibt sich der folgende iterative Algorithmus für α, β und $\mathbf{w}$:

1. **Initialisierung:** Setze (α, β) auf ihre Startwerte gemäß der gewählten a-priori Verteilung , wähle $\mathbf{w}$ zufällig aus dieser Verteilung.
2. **Wiederhole**
   Trainiere das neuronale Netz eine bestimmte Anzahl an Epochen, um $S(\mathbf{w})$ zu minimieren. Berechne (α, β) gemäß obiger Formeln
   bis (α, β) **konvergiert.**

Genau an dieser Stelle läßt sich das Bayes'sche Lernen mit beträchtlichem Effizienzgewinn mit einem Suchverfahren verzahnen (Ragg & Gutjahr, 1998). Jeder Iterationsschritt liefert einen neuen Suchpunkt $(\mathbf{w}, \alpha, \beta)^{(i)}$, den der Algorithmus weiterverfolgen oder verwerfen kann. In Kapitel 4.5 wird diese Methodik eingeführt und in Kapitel 5 eingehend untersucht. Die Entscheidung, welche Suchpunkte weiterverfolgt werden sollen, basiert auf der 'geschätzten' Qualität des neuronalen Netzes, der Modellevidenz. Abbildung 2.8 zeigt, daß die Evidenz bereits während des Iterationsprozesses eine relative Bewertung der Modelle gegeneinander erlaubt.

Jeffreys Prior für α und β

Der Prior für α und β wurde konstant angenommen, um keinen Wert zu bevorzugen (Bishop, 1995). Gutjahr hat in seiner Dissertation (Gutjahr, 1999) gezeigt, daß diese Annahme für das Bayes'sche Lernen ein Problem mit sich bringt: Uniforme a-priori Verteilungen sind nicht invariant unter bijektiven Transformationen.

Die Zufallsvariable θ, mit $p(\theta) = 1$ für $\theta > 0$, wird durch die bijektive Transformation, $\nu = \ln(\theta)$, auf die gesamte reelle Achse abgebildet. Die Verteilung von ν ist nun gegeben durch $p(\nu) = |J|\ p(\theta)$, wobei $J = \frac{\partial\theta}{\partial\nu}$ die Ableitung der inversen Transformation darstellt. Das führt aber zu einer Dichte, die nicht mehr gleichverteilt ist: $p(\nu) = e^{\nu}$. Mittels einer einfachen Transformation ist also aus einer *nicht-informativen* Zufallsvariablen θ eine *informative* Zufallsvariable ν geworden. Eine tiefergehende Diskussion der Problematik findet sich bei (Berger, 1980) und (Carlin & Louis, 1996).

Von Jeffreys wurde eine Methode vorgeschlagen, a-priori Wahrscheinlichkeitsverteilungen zu berechnen, die invariant unter bijektiven Transformationen sind, sogenannte *Jeffreys Prior* (Jeffreys, 1961). Berechnet man Jeffreys Prior für die Hyperparameter α und β, dann erhält man in beiden Fällen einen Prior, der konstant auf einer logarithmischen Skala ist (siehe (Gutjahr, 1999):

$$p(\beta) = \frac{1}{\beta} \qquad \text{bzw.} \qquad p(\alpha) = \frac{1}{\alpha}. \tag{2.24}$$

Damit erweitert sich Gleichung (2.18) um zwei weitere Terme zu

$$\begin{aligned} \ln p(\alpha,\beta|D) &\propto \ln p(D|\alpha,\beta) + \ln p(\alpha,\beta) \\ &= -\beta E_D - \alpha E_W - \frac{1}{2}\ln\ \det(\mathbf{A}) \\ &\quad -\frac{N}{2}\ln(2\pi) + \frac{N}{2}\ln(\beta) + \frac{W}{2} log(\alpha) \\ &\quad -\ln\alpha - \ln\beta. \end{aligned} \tag{2.25}$$

Die Ableitung nach α ergibt dann in analoger Weise zu 2.19 die Update-Regel:

$$\begin{aligned} 0 &\stackrel{!}{=} -E_W - \frac{1}{2}\sum_{j=1}^{W}\frac{1}{\lambda_j+\alpha} + \frac{W}{2\alpha} - \frac{1}{\alpha} \\ &= 2\alpha E_W + \alpha\sum_{j=1}^{W}\frac{1}{\lambda_j+\alpha} + W - 2 \\ \Longleftrightarrow\ & \alpha = \frac{\gamma}{2E_W} - \frac{1}{E_W}. \end{aligned} \tag{2.26}$$

Für β gilt entsprechend

$$\begin{aligned} 0 &\stackrel{!}{=} -E_D - \frac{1}{2}\frac{\gamma}{\beta} + \frac{N}{2\beta} - \frac{1}{\beta} \\ &= 2\beta E_D - N + \gamma + 2 \\ \Longleftrightarrow\ & \beta = \frac{N-\gamma}{2E_D} - \frac{1}{E_D}. \end{aligned} \tag{2.27}$$

In (Gutjahr, 1998, Ragg & Gutjahr, 1998, Gutjahr, 1999) wurde Jeffreys Prior für die Entwicklung von neuronalen Netzen mit der Bayes'schen Methode erfolgreich eingesetzt. In der vorliegenden Arbeit wurden bei der iterativen Bestimmung von α und β ausschließlich diese Formeln verwendet. Im praktischen Einsatz kann es sonst bei der Verwendung der Formeln (2.19) bzw. (2.20) insbesondere bei Klassifikationsproblemen vorkommen, daß die Parameter divergieren (Ragg & Gutjahr, 1998). Die Folge davon ist, daß eine konstante Funktion gelernt wird.

Die a-priori Verteilungen, die man durch Jeffreys Prior erhält, kann man als die *natürlichen* a-priori Verteilungen für die Hyperparameter bezeichnen. Insbesondere wechselt man nicht die Annahmen von der zweiten Stufe des Bayes'schen Ansatzes zur dritten Stufe (vgl. auch (Gutjahr, 1999)).

Approximation der Evidenz für α und β

Hat man die nach der Theorie optimalen Werte für die Hyperparameter α und β gefunden, dann kann man die Evidenz für α und β durch eine Gaußfunktion approximieren Man betrachtet die Evidenz $p(D|\ln\alpha, \ln\beta)$ als Funktion von $\ln\alpha$ bzw. $\ln\beta$ (siehe dazu auch (Gutjahr, 1999, Bishop, 1995)). Unter der Annahme der Unabhängigkeit der beiden Hyperparameter ergibt sich für $p(D|\ln\beta)$:

$$p(D|\ln\beta) = p(D|\ln\beta_{opt}) e^{-\frac{(\ln\beta - \ln\beta_{opt})^2}{2\sigma^2_{\ln\beta}}}. \tag{2.28}$$

Dies ist eine Gaußapproximation um den wahrscheinlichsten Wert β_{opt}. Logarithmiert man diese Gleichung, differenziert partiell nach β und beachtet noch, daß β_{opt} das Maximum der Funktion ist, dann gewinnt man daraus den Zusammenhang (siehe dazu insbesondere (Gutjahr, 1999), S.150)

$$\frac{1}{\sigma^2_{\ln\beta}} = -\beta\frac{\partial}{\partial\beta}\left(\beta\frac{\partial}{\partial\beta}\ln p(D|\ln\beta)\right). \tag{2.29}$$

Aufgrund von $\frac{\partial}{\partial\beta}\ln p(D|\ln\alpha, \ln\beta) = \frac{\partial}{\partial\beta}\ln p(D|\ln\beta)$ kann man in dieser Gleichung $\ln p(D|\ln\beta)$ durch den Ausdruck (2.18) substitutieren und dann noch den Zusammenhang aus der Update-Regel (2.22) anwenden. Damit ergibt sich

$$\frac{1}{\sigma^2_{\ln\beta}} = \frac{1}{2}N - \gamma + \frac{1}{2}\sum_{i=1}^{W}\frac{\alpha\lambda_i}{(\alpha+\lambda_i)^2}. \tag{2.30}$$

Die Summanden des zweiten Terms sind nur dann signifikant von Null verschieden, wenn λ_i und α in der gleichen Größenordnung liegen. Das wird selten für mehrere Eigenwerte der Fall sein (Gutjahr, 1999, Bishop, 1995). Den zweiten Term kann man also vernachlässigen und erhält damit eine einfache Approximation für die Varianz

$$\sigma^2_{\ln\beta} \approx \frac{1}{2}N - \gamma. \tag{2.31}$$

In analoger Weise erhält man für die Verteilung von $\ln \alpha$ mit Gleichung (2.18) und (2.21)

$$\frac{1}{\sigma^2_{\ln \alpha}} = -\alpha \frac{\partial}{\partial \alpha} \left(\alpha \frac{\partial}{\partial \alpha} \ln p(D| \ln \alpha) \right). \tag{2.32}$$

Daraus erhält man für die Varianz der Verteilung

$$\frac{1}{\sigma^2_{\ln \alpha}} = \frac{\gamma}{2} + \frac{1}{2} \sum_{i=1}^{W} \frac{\alpha \lambda_i}{(\alpha + \lambda_i)^2} \approx \frac{\gamma}{2}. \tag{2.33}$$

Damit hat man nun auch eine Approximation der Evidenz für α und β, nachdem vorher bereits ein Algorithmus angegeben werden konnte, mit dem man die Hyperparameter α und β auf ihre optimalen Werte einstellen kann.

2.5.3 Die Evidenz eines Modells

Um ein Qualitätsmaß, die Modellevidenz, für ein neuronales Netz $\mathcal{H}$ zu gewinnen, läßt sich aufbauend auf der Evidenz für α und β in einer dritten Stufe wieder das Bayes'sche Theorem anwenden:

$$P(\mathcal{H}|D) \quad = \quad \frac{p(D|\mathcal{H}) \cdot P(\mathcal{H})}{p(D)}.$$

Dabei geht man von neuronalen Netzen mit unterschiedlicher Topologie aus, deren Gewichtsvektor mit der bisher entwickelten Methodik optimiert wurde. Falls die a-priori Wahrscheinlichkeit für alle Modelle gleich ist, genügt es wieder, die bedingte Wahrscheinlichkeit $p(D|\mathcal{H})$, die Evidenz für $\mathcal{H}$ zu betrachten und mit den anderen zu vergleichen, um ein Auswahlkriterium zu erhalten.

Wie auch auf der zweiten Stufe erfordet die korrekte Anwendung der Bayes'schen Methode über unbekannte Parameter α und β zu integrieren. Die Modellevidenz ergibt sich dann zu

$$p(D|\mathcal{H}) = \int \int p(D|\alpha, \beta, \mathcal{H}) p(\alpha, \beta|\mathcal{H}) \, d\alpha \, d\beta.$$

Der erste Faktor $p(D|\alpha, \beta, \mathcal{H})$ entspricht der Evidenz aus der zweiten Stufe. Man beachte, daß im folgenden die a-priori Annahmen $p(\alpha) = \frac{1}{\alpha}$ und $p(\beta) = \frac{1}{\beta}$ vorausgesetzt werden. Das bedeutet für den praktischen Einsatz des Verfahrens, daß die Optimierungsvorschrift mittels der Jeffreys Prior implementiert werden muß, abweichend von dem zuvor in Kapitel 2.5.2 bzw. in (Bishop, 1995) angegebenen Algorithmus.

Die Integration über α und β läßt sich auf dieser Stufe mit den Gaußapproximationen aus den Gleichungen (2.29 und (2.32) durchführen. Im ersten Fall erhält man

$$p(D|\beta_{opt}) \int \exp\left(-\frac{(\ln \beta - \ln \beta_{opt})^2}{2\sigma^2_{\ln \beta}} \right) d \ln \beta = p(D|\beta_{opt}) \sqrt{2\pi} \sigma_{\ln \beta}. \tag{2.34}$$

Im zweiten Fall erhält man ein entsprechendes Ergebnis, so daß sich die Modellevidenz wie folgt ausdrücken läßt

$$p(D|\mathcal{H}) = p(D|\alpha_{opt}, \beta_{opt}) 2\pi \sigma_{\ln \alpha} \sigma_{\ln \beta}. \tag{2.35}$$

Für den Logarithmus der Evidenz ergibt sich mit 2.18, 2.31 und 2.33

$$\begin{aligned} \ln p(D|\mathcal{H}) &= -\beta_{opt} E_D - \alpha_{opt} E_W - \frac{1}{2} \ln \det(\mathbf{A}) \\ &- \frac{N}{2} \ln(2\pi) + \frac{N}{2} \ln(\beta_{opt}) + \frac{W}{2} \ln(\alpha_{opt}) \\ &+ 2 \ln \sqrt{2\pi} + \frac{1}{2} \ln \left(\frac{2}{\gamma} \right) + \frac{1}{2} \ln \left(\frac{2}{N - \gamma} \right). \end{aligned} \tag{2.36}$$

Auf dieser Stufe wird das Ergebnis sensitiver bezüglich kleiner Abweichungen, weil die Determinante der Hessematrix durch das Produkt der Eigenwerte bestimmt wird. Für den hier entwickelten Ansatz ist es ausreichend, daß man ein Qualitätsmaß zur Verfügung hat, das ausschließlich durch den Trainingsprozeß gewonnen wird und mit der tatsächlichen Güte des Modells korreliert ist. Auf diesem analytischem Gütekriterium läßt sich dann ein evolutionärer Suchprozeß ohne die Gefahr des Overfitting aufbauen, wie sie bei der iterativen Verwendung einer Kreuzvalidierungsmenge gegeben wäre.

Abschließend sei noch festgehalten, daß aufgrund der Vertauschbarkeit von versteckten Neuronen meist noch ein Symmetriefaktor berücksichtigt wird. Ein neuronales Netz mit einer versteckten Schicht und M verstecken Neuronen hat insgesamt $2^M M!$ äquivalente Gewichte, die zu derselben Ausgabe führen (Bishop, 1995, Gutjahr, 1999). Den Term $M!$ erhält man durch alle möglichen Permutationen der Neuronenanordnung im Netz. Der Faktor 2^M berücksichtigt, daß man die Vorzeichen der Gewichte am Eingang eines versteckten Neurons und am Ausgang vertauschen kann. Dem Ausdruck für die Modellevidenz sind dann noch die beiden Terme $\ln M! + M \ln 2$ hinzuzufügen. Dies hat die praktische Konsequenz, daß größere Topologien nicht so hart bestraft werden wie in Gleichung (2.36).

2.5.4 Praktische Gesichtspunkte des Bayes'schen Lernens

Der Theorie der Bayes'schen Regularisierung entspringt ein iteratives Verfahren, um wichtige Modellparameter, den Gewichtsvektor $\mathbf{w}$ und den Regularisierungsfaktor λ, automatisch einzustellen, ohne daß dazu weitere Daten oder Experimente von Seiten des Entwicklers nötig wären. Für den praktischen Einsatz ist besonders wichtig, daß wichtige Designentscheidungen automatisch getroffen werden. Möchte man beispielsweise für eine gegebene Topologie eine geeignete Gewichtung des Weight-Decay experimentell finden, dann muß man dazu nacheinander mehrere Werte ausprobieren und für jeden Wert mehrere Gewichtsinitialisierungen trainieren, um einen brauchbaren Mittelwert zu erhalten. Bei 5 Werten für λ und 10 Initialisierungen hat man bereits 50 Netze zu trainieren, nur um einen Parameter zu bestimmen! Variiert man dazu noch die Topologie, dann vervielfacht sich der Aufwand entsprechend, da sich optimale Gewichtung mit der Zahl der Parameter ändert.

Im folgenden sollen weitere Aspekte des Bayes'schen Lernens betrachtet werden. Die Bildung von Gewichtsgruppen ist aus Konsistenzgründen wichtig. Die Verteilung der Netzausgaben liefert ein weiteres wichtiges Kriterium, nämlich Konfidenzschranken für die Ausgabefunktion. Weiterhin soll gezeigt werden, daß die Kombination der Parameteroptimierung mit der adaptiven Regularisierung kompatibel ist, wenn man das Lernverfahrem *Rprop* einsetzt. Abschließend werde ich noch auf die Initialisierung der Hyperparameter und den Zeitpunkt der Anpassung eingehen.

Gewichtsgruppen

Ein Nachteil der Regularisierung mit einfachem Weight-Decay ist die Tatsache, daß dieser nicht konsistent ist mit Skalierung der Eingabe oder Ausgabe (Bishop, 1995). Wendet man auf die Daten eine Lineartransformation an, dann sollten sich die trainierten Netze ebenfalls nur bis auf eine Lineartransformation unterscheiden. Geht man von gleich initialisierten Netzen aus, dann kann man - wenn man keine Regularisierung verwendet - nach dem Training das eine Netz durch eine Lineartransformation der Gewichte in das andere überführen. Der Regularisierungsterm sollte mit dieser Eigenschaft verträglich sein. Um die Konsistenz zu gewährleisten, müßte man bei einem Netz mit einer versteckten Schicht zwei Gruppen von Gewichten bilden, indem die Gewichte von der Eingabe ausgehend in der einen, die Gewichte zur Ausgabe gehend in der zweiten Gruppe zusammengefaßt werden. Die Gewichtungsparameter für die beiden Teile des Strafterms sind heuristisch aber kaum noch vernünftig zu bestimmen. Ein großer Vorteil des Bayes'schen Lernens liegt auch darin, daß dort die Gewichte auf vielfältige Weise in Gruppen zusammengefaßt werden können, da die Gewichtung der Strafterme automatisch berechnet wird. Die Bestimmung der Hyperparameter läßt sich problemlos auf mehrere Gewichtsgruppen übertragen. Für eine ausführliche Darstellung sei wieder auf (Nautze, 1997) und (Gutjahr, 1999) verwiesen.

Die zu minimierende Fehlerfunktion $S(\mathbf{w})$ verändert sich gegenüber Gleichung (2.11) zu

$$S(\mathbf{w}) = \beta \cdot \frac{1}{2} \sum_{i=1}^{N} (y(x_i, \mathbf{w}) - t_i)^2 + \sum_{k=1}^{G} \left(\alpha_k \cdot \frac{1}{2} \sum_{w \in \mathcal{W}_k} w^2 \right). \tag{2.37}$$

Für jede Gewichtsgruppe ist jetzt also ein Hyperparameter α_k zu berechnen. Der Logarithmus der Modellevidenz ergibt sich dadurch im Vergleich zu Gleichung (2.36) zu

$$\begin{aligned} \ln p(D|\mathcal{H}) &= -\beta E_D - \sum_{k=1}^{G} \alpha_k E_{\mathcal{W}_k} - \frac{1}{2} \ln \det(\mathbf{A}) \\ &- \frac{N}{2} \ln(2\pi) + \frac{N}{2} \ln(\beta) + \sum_{k=1}^{G} \frac{W_K}{2} \ln(\alpha_k) \\ &+ (G+1) \ln \sqrt{2\pi} + \frac{1}{2} \sum_{k=1}^{G} \ln \left(\frac{2}{\gamma_k} \right) + \frac{1}{2} \ln \left(\frac{2}{N-\gamma} \right) \\ &+ \ln M! + M \ln 2. \end{aligned} \tag{2.38}$$

In meinen Untersuchungen hat sich die folgende Vorgehensweise bei der Einteilung der Gewichte als praktikabel erwiesen:

- Alle Gewichte zu einer Ausgabe bilden eine Gruppe, außer bei 1-aus-N Kodierungen.
- Jeweils die Gewichte zwischen zwei Schichten bilden eine Gruppe.
- Die Biase bilden eine eigene Gruppe, die nicht regularisiert wird, wenn der Mittelwert der Zielwerte von Null verschieden ist.
- Sind wenig Daten vorhanden und/oder sind die Daten stark verrauscht, dann ist es vernünftig, die Zahl der Hyperparameter so klein wie möglich zu halten, da diese anhand der Daten geschätzt werden müssen. Auf die Verwendung mehrerer Gruppen kann hier manchmal mit Gewinn verzichtet werden, da eine größere Anzahl von Gewichten in einer Gruppe die Bestimmung des Hyperparameters robuster macht. Dies ist beispielsweise dann der Fall, wenn die versteckte Schicht wenige Neuronen hat (< 5), da sonst mit nur wenigen Gewichten eine Normalverteilung geschätzt werden muß, was zu großen Ungenauigkeiten und damit Instabilitäten im Lernprozeß führen kann.

Eine spezielle Gruppeneinteilung macht man sich auch in dem Verfahren *Automatic Relevance Determination* (ARD) zunutze (MacKay, 1994, Neal, 1994, Gutjahr, 1999). Alle Gewichte, die von einer Eingabe ausgehen, werden in einer Gewichtsgruppe zusammengefaßt. Wird der zugehörige Hyperparameter groß, dann kann man die Eingabe löschen. Ein großer Nachteil des Verfahrens ist, daß man große versteckte Schichten benötigt, damit die Schätzung der Hyperparameter auf genügend Gewichten basiert. Dies ist zum einen nicht kompatibel mit einer Optimierung der Zahl der versteckten Neuronen, zum anderen verlängert es auch die Trainingszeit der Netze durch die größere Anzahl an Gewichten ganz erheblich. In der Praxis hat sich gezeigt, daß trotz des höheren Aufwandes aber keine besseren Resultate zu erzielen sind als mit dem Standard-Verfahren. Ein weiterer Nachteil des Verfahrens ist es, daß es nicht in ein modulares Konzept paßt, sondern nur für neuronale Netze anwendbar ist. Hier verfolge ich das Ziel, ein integriertes Konzept bereitzustellen, dessen Bausteine möglichst flexibel austauschbar sein sollen. Das heißt, die Optimierung der Eingabe/Ausgabe Relation sollte nur anhand der Daten erfolgen, ohne das spezielle Modell zu kennen, das zum Training verwendet wird.

Gutjahr führte in seiner Dissertation eine variable Einteilung der Gewichte in Gruppen ein (Gutjahr, 1999). Für jede Gruppe wird der Mittelwert μ_k als zusätzlicher Hyperparameter eingeführt, der ebenso mit einem Bayes'schen Ansatz optimiert wird. Nach der Optimierung des Mittelwert μ_k und Varianz α_k jeder Gruppe ist dann die zugehörige Normalverteilung, d.h. die neue a-priori Verteilung, bestimmt. Jedes Gewicht wird dann in die Gruppe eingeordnet, für die sein Wert die höchste Wahrscheinlichkeit hat. Anschließend wird der nächste Iterationsschritt durchgeführt. Die Einteilung der Gewichte wird dabei zunehmend stabiler. Dabei wird von einer festen Topologie ausgegangen. Eine Kombination dieser variablen Einteilung der Gewichte mit der Optimierung der Topologie, wie sie in dieser Arbeit vorgeschlagen wird, ist nicht ohne weiteres möglich. Jeder Mutationsschritt, bei dem ein Neuron entfernt wird, verändert die Gruppierung der Gewichte. Insbesondere dann, wenn zu dem

betreffenden Neuron viele Gewichte einer Gruppe gehören, wird die iterative Anpassung der Hyperparameter erheblich gestört. Es liegt außerhalb der Zielsetzung dieser Arbeit, die praktischen Probleme bei der kombinierten Optimierung der Topologie und vieler Hyperparameter mit variablen Gewichtsgruppen zu lösen.

Konfidenzintervalle für die Ausgabefunktion

Ein weiterer Vorteil des Bayes'schen Lernens besteht darin, daß man aufgrund der wahrscheinlichkeitstheoretischen Modellierung *Fehlerbalken* (engl.: *error bars*) für jeden Ausgabewert angeben kann. Ein trainiertes neuronales Netz entspricht im Bayes'schen Ansatz dem Maximum einer a-posteriori Verteilung von Gewichten. Für eine neues Muster erhält man jetzt nicht nur eine reelle Zahl als Ausgabe, sondern man kann die Verteilung der Gewichte nutzen, um eine Verteilung für die Netzausgabe zu gewinnen. Der Mittelwert der Verteilung entspricht dabei gerade der Ausgabe des neuronalen Netzes. Für die a-posteriori Verteilung $p(\mathbf{w}|D)$ wird die Approximation (2.15) verwendet.

Die Verteilung der Zielwerte läßt sich bei gegebener Eingabe $\mathbf{x}$ schreiben als

$$p(t|\mathbf{x}, D) = \int p(t|\mathbf{x}, \mathbf{w}) p(\mathbf{w}|D)\, d\mathbf{w}. \tag{2.39}$$

$p(t|\mathbf{x}, \mathbf{w})$ ist die Wahrscheinlichkeit, den tatsächlichen Zielwert zu schätzen aus Gleichung (2.9) bei Eingabe $\mathbf{x}$ und gegebenen $\mathbf{w}$. Setzt man die entsprechenden Ausdrücke ein und streicht alle Faktoren, die von t unabhängig sind, dann erhält man

$$p(t|\mathbf{x}, D) \propto \int e^{-\frac{\beta}{2}(t-y(\mathbf{x};\mathbf{w}))^2} e^{-\frac{1}{2}(\Delta\mathbf{w})^T \mathbf{A} (\Delta\mathbf{w})}\, d\mathbf{w}. \tag{2.40}$$

Bei Regressionsproblemen sollte das Ausgabeneuron nach Voraussetzung eine lineare Aktivierungsfunktion haben. Mit einer linearen Taylor-Entwicklung um den optimierten Gewichtsvektor $\mathbf{w}_{opt}$

$$y(\mathbf{x}, \mathbf{w}) = y(\mathbf{x}, \mathbf{w}_{opt}) + \mathbf{g}^T \Delta\mathbf{w} \ , \text{ wobei } \mathbf{g} \equiv \nabla_{\mathbf{w}} y|_{\mathbf{w}_{opt}},$$

läßt sich der Ausdruck (2.40) weiter vereinfachen zu

$$p(t|\mathbf{x}, D) \propto \int e^{-\frac{\beta}{2}(t-y_{opt}-\mathbf{g}^T\Delta\mathbf{w})^2 - \frac{1}{2}(\Delta\mathbf{w})^T \mathbf{A} (\Delta\mathbf{w})}\, d\mathbf{w}. \tag{2.41}$$

Das Integral läßt sich nun analytisch auswerten (für die Berechnung siehe (Gutjahr, 1999, Bishop, 1995)) und man erhält eine Gaußverteilung der Form

$$p(t|\mathbf{x}, D) = \frac{1}{\sqrt{2\pi\sigma_t^2}} e^{-\frac{(t-y_{opt})^2}{2\sigma_t^2}}. \tag{2.42}$$

Der Mittelwert der Verteilung ist y_{opt} und die Varianz ist gegeben durch die Beziehung

$$\sigma_t^2 = \frac{1}{\beta} + \mathbf{g}^T \mathbf{A}^{-1} \mathbf{g}. \tag{2.43}$$

Da der Hyperparameter β bzw. die Varianz des Rauschanteils als unabhängig von der aktuellen Eingabe betrachtet wird, bestimmt vor allem der zweite Term $g^T \mathbf{A}^{-1} g$ die Form der Verteilung. Williams zeigt für lineare Modelle, daß die bestimmenden Faktoren die Distanz zu den Trainingspunkten und die Krümmung der approximierenden Funktion sind (Williams *et al.*, 1995). In Bereichen mit wenig Trainingsinformation und starken Schwankungen der Ausgabefunktion ist die Varianz groß. Mittels der Varianz lassen sich also Konfindenzwerte für die Ausgabefunktion berechnen. Diesen Umstand lohnt es sich bei der Bildung von Komitees zu berücksichtigen. Abbildung 2.3 zeigt die Varianz für ein trainiertes Netz für das Sinusproblem.

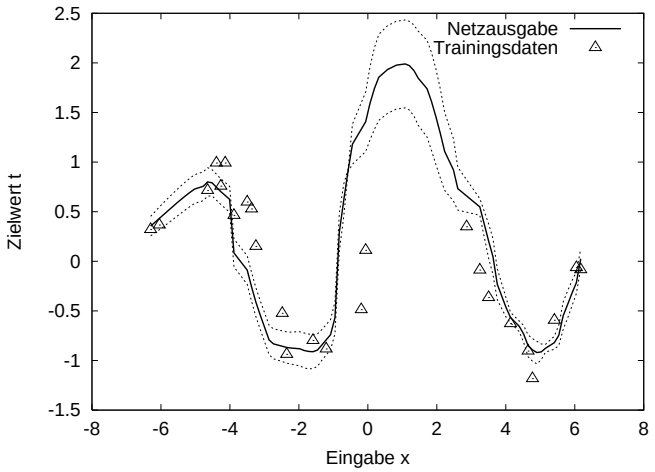


Abbildung 2.3: Die Abbildung zeigt die Ausgabe eines neuronalen Netzes, das mit einer Teilmenge von Daten aus Abbildung 1.1a trainiert wurde. Die beiden dünneren Linien ober- und unterhalb der Ausgabe geben die Varianz $t \pm \sigma^2$ wieder. In den Bereichen, in denen wenig Trainingsdaten vorhanden sind, ist die Varianz deutlich größer. Ähnliches gilt für Bereiche, in denen die Kurve stark gekrümmt ist und gleichzeitig benachbarte Datenpunkte starke Unterschiede aufweisen.

Für Klassifikationsprobleme läßt sich eine ähnliche Approximation für die Varianz herleiten, siehe (Bishop, 1995)):

$$\sigma^2(\mathbf{x}) = \mathbf{g}^T \mathbf{A}^{-1} \mathbf{g}.$$

Der Parameter β entfällt hier, da die Ausgaben keinem Rauschen unterliegen. Die Bayes'sche Methodik ermöglicht also für jedes neue Muster einen Konfidenzwert festzulegen. Diese zusätzliche Information soll in dem hier behandelten Ansatz dazu verwendet werden, die Gewichtung für die Mitglieder eines Komitees von neuronalen Netzen zu definieren (vgl. Kapitel 6.3). Ein Netz wirkt dann entsprechend seiner Varianz an der Ausgabe des Komitees für eine Eingabe $\mathbf{x}$ mit.

Parameteroptimierung und adaptive Regularisierung

Der Erfolg des Backpropagation beruhte darauf, daß man einen effizienten Algorithmus angeben konnte, wie in einem mehrschichtigen Netz die Ableitung des Fehlers nach den

Gewichten zu berechnen ist, um einen Gradientenabstieg für die Fehlerfunktion zu realisieren (Rumelhart *et al.*, 1986b, Rumelhart *et al.*, 1986a). Um den Gradientenabstieg durchzuführen, werden üblicherweise schnelle Lernverfahren, wie Konjugierte Gradienten Verfahren (Bishop, 1995, Fletcher, 1995, Møller, 1993) oder das simple, aber robuste *Rprop* (Riedmiller, 1994) eingesetzt.

Die iterative Anpassung des Gewichtungsfaktors λ birgt eine in der Literatur bisher noch nicht behandelte Problematik: Die Fehlerfunktion, deren Minimum man sucht, ändert sich bei jeder neuen Schätzung der Hyperparameter durch das Bayes'sche Verfahren. Abbildung 2.4 veranschaulicht den Sachverhalt. Der Strafterm entspricht dabei einer Parabel um den Ursprung, die zu der gesamten Fehlerfunktion mit einem skalierbaren Anteil beiträgt. Je größer der Skalierungsfaktor, desto stärker wird die Fehlerfunktion geglättet. Lokale Minima können bei einer Verstärkung der Skalierung verlorengehen oder wesentlich kleiner werden.

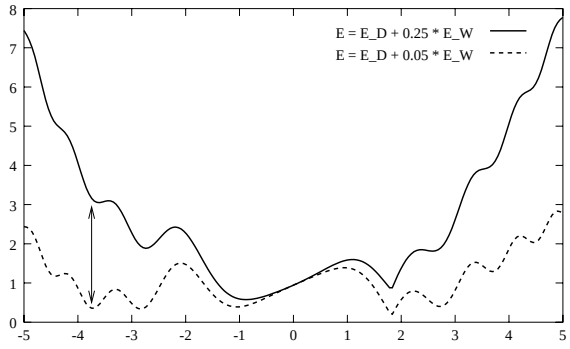


Abbildung 2.4: Die Abbildung veranschaulicht für eine einfache Fehlerfunktion die Änderung, die bei der iterativen Berechnung des Gewichtungsfaktors auftritt. Die erste Fehlerfunktion entspricht der gestrichelten Kurve, die nach Anpassung der Hyperparameter in die durchgezogene Kurve übergeht. Minima, die weit vom Ursprung entfernt liegen verschwinden oder werden stark geglättet, wie durch den Pfeil angedeutet.

Es ist nicht offensichtlich, ob und wieweit ein Lernverfahren mit der automatischen Gewichtung kompatibel ist. Anhand der beiden Verfahren, *Scaled Conjugate Gradient* und *Rprop*, wird diese Problematik im folgenden näher beleuchtet.

Für den Datensatz aus Abbildung 1.1a wurden 50 Gewichtsinitialisierungen sowohl mit *Rprop* als auch mit SCG trainiert. Das Netz hat eine 1-10-6-1 Topologie mit shortcut Verbindungen. Es wurde die übliche Einteilung in Gewichtsgruppen vorgenommen, wobei der Bias nicht regularisiert wurde. Der Update der Hyperparameter erfolgte alle 100 Epochen bei einer Gesamttrainingsdauer von 700 Epochen. Initialisiert wurde α_k mit 0.05 und β mit 1. Tabelle 2.1 gibt den mittleren Trainings- und Testfehler wieder.

	$\bar{E}_{Train}$	(σ_{Train})	$\bar{E}_{Test}$	(σ_{Test})
Rprop	0.0075	(0.001)	0.104	(0.06)
SCG	0.0076	(0.001)	0.139	(0.34)

Tabelle 2.1: Die Tabelle faßt die Ergebnisse des Trainings für den Datensatz aus Abbildung 1.1a zusammen. Die mit Rprop trainierten Modelle sind signifikant besser als mit SCG. Es wurden jeweils 50 Netze mit den 25 Datenpunkten aus Abbildung 2.3 trainiert.

Die bessere Generalisierungsleistung für Rprop ist signifikant. Dazu wurde ein t-Test auf Gleichheit der Mittelwerte zum Signifikanzniveau $\alpha = 0.05$ durchgeführt, dessen Nullhypothese bei einer Schwelle von $t_{0,95;50} = 1.68$ mit 2.28 abgelehnt wird. Berechnet man die Korrelation zwischen der Evidenz der Modelle und dem Fehler auf der Testmenge, dann ist

der Wert für Rprop etwas günstiger: -0.43 im Vergleich zu -0.27 (Abbildung 2.5). Auffallend ist auch, daß SCG zu einigen Lösungen führt, bei denen der Testfehler wesentlich größer als 0.4 ist, die Evidenz dies aber nicht erkennen läßt. Eine mögliche Erklärung für diesen Umstand ist das Verhalten bei der Anpassung der Hyperparameter.

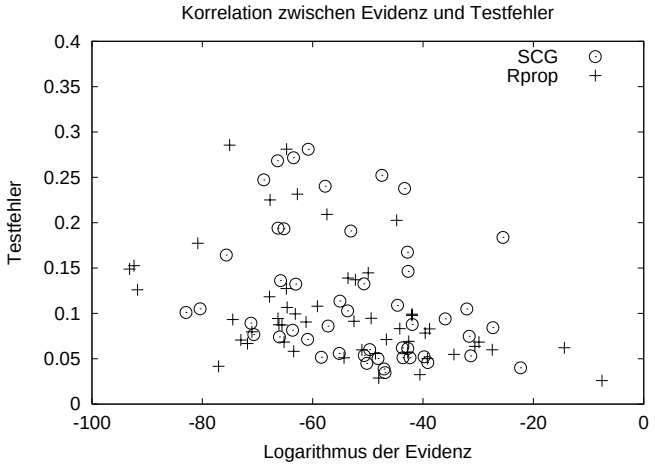


Abbildung 2.5: Die Abbildung veranschaulicht für den Datensatz aus Abbildung 1.1a den Zusammenhang zwischen Evidenz und Testfehler. Sowohl für Rprop (Kreuze) als auch für SCG (Kreise) ist die Korrelation negativ (-0.43 bzw. -0.27). Insbesondere gibt es beim Training mit SCG einige besonders schlechte Netze, die aus Darstellungsgründen nicht gezeigt sind (Starke Verzerrung des Maßstabes).

Im Experiment kann man beobachten, daß die Parameter, die bei SCG die Bestimmung der Schrittweite steuern, in diesen Fällen sehr schnell sehr klein werden. Das heißt, die quadratische Approximation wird als gut angenommen und die Schrittweite entsprechend verkleinert. Dadurch kann ein Minimum, das durch eine Anpassung der Hyperparameter nahezu verschwindet (vgl Abbildung 2.4), trotzdem nicht mehr verlassen werden. Da *Rprop* keinerlei Annahmen an die Form der Fehlerfunktion stellt, sondern vielmehr immer eine Beschleunigungs- und Abbremsphase durchläuft (vgl. Abbildung 2.2), tritt dieser Effekt nicht auf. Die Abbildungen 2.6 und 2.7 zeigen einen Trainingsverlauf, bei dem gerade der beschriebene Effekt eintritt. Nachdem die Hyperparameter das erstemal angepaßt wurden, verändert sich der Fehler für SCG im Prinzip nicht mehr. Nach einigen Schritten verharrt SCG auf der Stelle. Der Gradient ändert sich nicht mehr, wie im rechten Bild zu erkennen ist. Rprop findet dagegen jedesmal ein Minimum. Die Generalisierungsleistung ist für dieselbe Gewichtsinitialisierung bei Rprop wesentlich größer.

Das ungünstige Verhalten von SCG erklärt sich auch dadurch, daß Konjugierte Gradienten bereits für sich eine Regularisierungsmethode sind (Hanke, 1995). SCG kann nur ganz bestimmte Bewegungen im Gewichtsraum machen, während Rprop keinerlei Einschränkungen diesbezüglich kennt. Diese Gegebenheiten rechtfertigen es, für diese Arbeit ausschließlich das Lernverfahren Rprop einzusetzen.

Abbildung 2.8 zeigt noch die Entwicklung der Evidenz für fünf ausgewählte Netze für das obige Beispiel. Für die Kombination mit dem evolutionären Suchverfahren sollte man zwei Eigenschaften beachten. Erstens verläuft die Entwicklung der Evidenz oft nicht-monoton. Das birgt Probleme für Suchverfahren, die die bisher beste Lösung immer konservieren. Zum zweiten zeigt sich, daß Modelle, die am Ende eine hohe Evidenz haben, auch während des Iterationsprozesses tendentiell eine höhere Evidenz aufwiesen. Für die Verzahnung mit dem evolutionären Suchverfahren ist diese Eigenschaft wichtig, da höhere Evidenz einen Selektionsvorteil bringen soll.

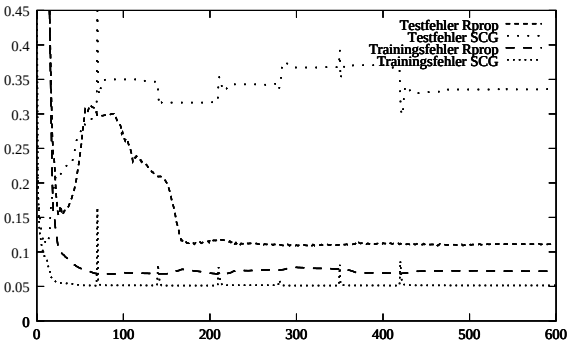


Abbildung 2.6: Verlauf des Trainings- und Testfehlers (pro Muster) für ein ausgewähltes Netz. Es ist nur der Fehler auf den Daten E_D angegeben. Für SCG ist deutlich zu erkennen, daß das Lernverfahren manchmal im selben lokalen Minimum steckenbleibt, während Rprop zu einem Minimum mit kleinerem Gesamtfehler gelangt.

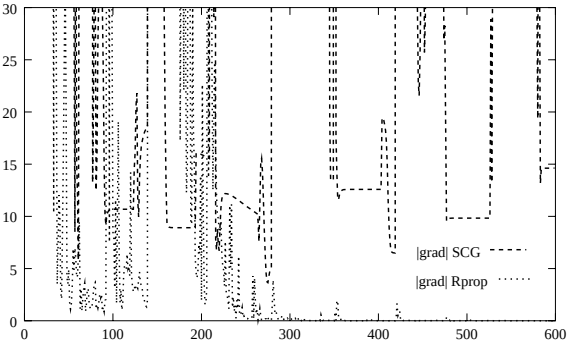


Abbildung 2.7: Verlauf der Norm des Gradienten. SCG macht nach jeder Anpassung der Hyperparameter einige Anpassungsschritte und verharrt dann auf der Stelle, erkennbar daran, daß die Norm konstant, aber größer als 0 bleibt. Rprop findet dagegen jedesmal wieder ein Minimum. Die Norm konvergiert gegen 0.

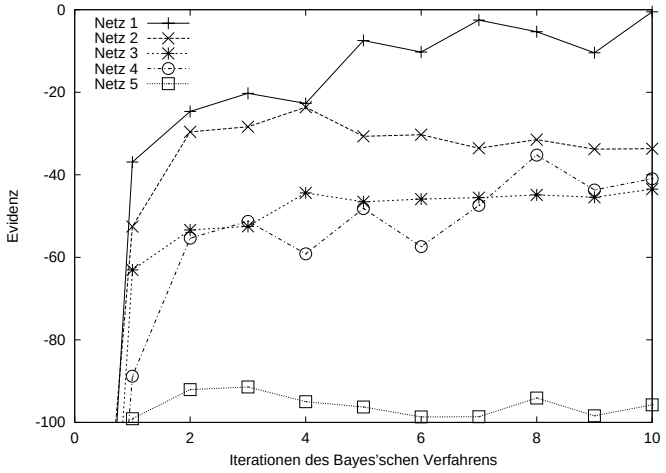


Abbildung 2.8: Die Abbildung veranschaulicht für den Datensatz aus Abbildung 1.1a die Entwicklung der Evidenz für fünf ausgewählte Modelle. Es ist deutlich zu erkennen, daß Modelle, die am Ende eine höherer Evidenz haben, auch während des Trainings tendentiell bessere Werte hatten. Der Verlauf der Evidenz ist durchaus unterschiedlich für verschiedene Gewichtsinitialisierungen und Phasen des Trainings: Nicht-monoton (Netze 1,2,4), ansteigend (Netz 1,4), fallend (Netze 2,5) oder relativ stabil (Netze 2,3).

Initialisierung und Anpassung der Hyperparameter

Das Verfahren zur Bestimmung der optimalen Hyperparameter setzt voraus, daß man die Parameter am Anfang initialisiert, bevor man das Netz trainiert, bis ein lokales Minimum erreicht ist. Es ist also die Varianz des Priors zu wählen und die Gewichte sind gemäß dieser Verteilung zu initialisieren. Über eine sinnvolle Initialisierung der Hyperparameter sind in der Literatur keine Angaben zu finden.

Eine Möglichkeit ist es, am Anfang keine Regularisierung durchzuführen, d.h. mit $\alpha = 0$ zu beginnen. Sind die Daten allerdings mit starkem Rauschen behaftet, dann führt das zuerst zu Overfitting-Effekten. Es empfiehlt sich also zumindest eine leichte Regularisierung im ersten Schritt. Da die Gewichte auch aus einer Verteilung mit Varianz $1/\alpha$ gewählt werden sollen, empfiehlt es sich, einen Wert aus dem Intervall $[0.1, 1.0]$ zu wählen. Tabelle 2.2 verdeutlicht am Beispiel aus Abbildung 1.1a die Unterschiede zwischen richtiger und unpassender Initialisierung. Die Ergebnisse sind signifikant besser für die richtige Initialisierung (t-Wert

5.72 bei Schwelle $t_{0.95;50} = 1.68$). Das Ergebnis fällt allerdings nicht so deutlich aus, wie man es vielleicht erwarten würde. Das deutet darauf hin, daß das Bayes'sche Verfahren in vielen Fällen den Gewichtungsparameter auch bei einer Verletzung der a-priori Annahmen vernünftig einstellt. Abbildung 2.9 illustriert den Zusammenhang.

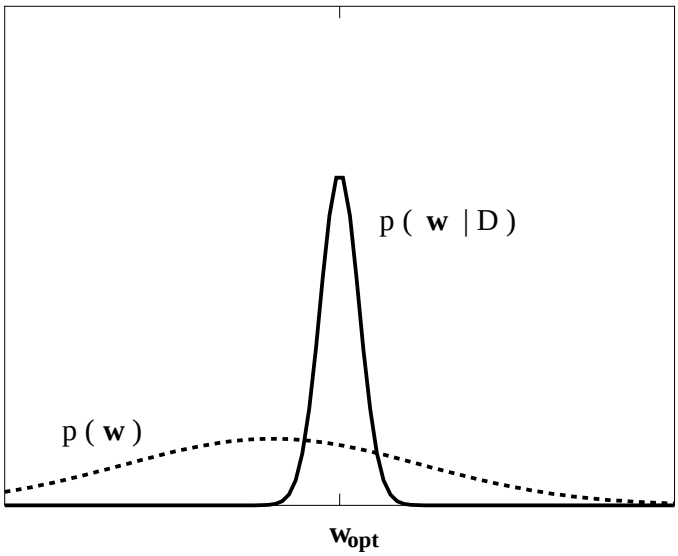


Abbildung 2.9: Schematische Darstellung der a-priori Verteilung der Gewichte $p(\mathbf{w})$ und der a-posteriori Verteilung, die das Bayes'sche Verfahren berechnet. Der wahrscheinlichste Gewichtsvektor ist an der Stelle des Maximums der a-posteriori Verteilung. Im allgemeinen wird die a-posteriori Verteilung eine komplexe Struktur haben.

Ausgehend von einer breiten Verteilung versucht das Bayes'sche Verfahren die Hyperparameter so einzustellen, daß die a-posteriori Verteilung um den wahrscheinlichsten Wert konzentriert ist. Ist die a-priori Annahme verletzt, beispielsweise weil die Gewichte mit zu kleinen Werten initialisiert werden, dann bedeutet das, daß der Prior schon sehr eng um einen Wert konzentriert ist, das Bayes'sche Verfahren aber von einer breiten Verteilung ausgeht. Ist der Fehler nun groß, dann wird das Bayes'sche Verfahren die Gewichtung des Regularisierungsterms erhöhen. Die a-posteriori Verteilung wird dann noch konzentrierter ausfallen. Die Hyperparameter können unter diesen Umständen divergieren. Im Experiment kann man diese Auswirkungen beobachten.

Ein weiteres Problem betrifft den genauen Zeitpunkt der Anpassung der Hyperparameter. Gemäß der Annahmen in Gleichung (2.13) ist es notwendig, den Gradientenabstieg vor der Neuberechnung von α und β bis in ein Minimum durchzuführen. In der praktischen Berechnung kann man sich dem Minimum nur bis auf einen Unterschied ϵ der Fehlerwerte nähern. Das heißt, man muß einen Zeitpunkt festlegen, wann das Training abgebrochen werden soll. In der Literatur wurde mehrfach vorgeschlagen, den Abbruch automatisch vorzunehmen, entweder wenn die Fehlerdifferenz zum Vorgängerschritt unter einen vorgegebenen Wert δ fällt (Penny & Roberts, 1998) oder wenn dasselbe für die Norm des Gradienten gilt. In beiden Fällen wurde vorgeschlagen, die Schwelle δ bei jeder Anpassung abzusenken. Die Vorgehensweise von Penny und Roberts ist bereits deshalb problematisch, da auch in steilen Bereichen der Fehlerfunktion abgebrochen werden kann, falls der letzte Schritt über ein Minimum hinaus geht und wieder zu einem ähnlichen Fehler führt (Gutjahr, 1999).

Ich möchte dagegen argumentieren, daß man ebensogut eine feste Epochenzahl festlegen kann, nach der die Anpassung erfolgt. Die minimale Zahl an Epochen ermittelt man, indem man einige Netze nach Initialisierung der Hyperparameter solange trainiert, bis die Norm der Gradienten klein genug ist. Stellt man die zu trainierende Epochenzahl größer ein, dann

sind die Annahmen in Gleichung 2.13 nur genauer erfüllt. Man sollte aufgrund des längeren Trainings also keinen Unterschied feststellen. Andererseits ist die Festlegung der Schwelle δ und der Geschwindigkeit der Absenkung wiederum ein Parameter, den es zu bestimmen gilt. Eine feste Epochenzahl scheint mir leichter zu bestimmen zu sein.

		$\bar{E}_{Train}$	(σ_{Train})	$\bar{E}_{Test}$	(σ_{Test})
I.	$\alpha_k = 0$	0.0078	(0.001)	0.130	(0.10)
II.	Anpassung fix	0.0075	(0.001)	0.104	(0.06)
III.	Anpassung automatisch	0.0074	(0.001)	0.111	(0.08)

Tabelle 2.2: Die Tabelle faßt die Ergebnisse des Trainings für den Datensatz aus Abbildung 1.1a zusammen für die drei Versuche: (I) Die Hyperparameter α_k sind mit 0 initialisiert. (II) Anpassung der Hyperparameter erfolgt alle 100 Epochen. (III) Anpassung der Hyperparameter erfolgt automatisch, erstmalig dann, wenn die Norm des Gradienten kleiner als δ ist. Dieser Schwellwert wird nach jeder Anpassung auf die Hälfte verkleinert. Die besseren Ergebnisse für die geeignete Initialisierung sind signifikant, nicht jedoch für den Unterschied zwischen fixer und automatischer Anpassung.

Insbesondere birgt die Absenkung der Schwelle bei einer Kombination des Bayes'schen Lernens mit einem evolutionären Suchverfahren nicht zu unterschätzende Probleme: Bei der Mutation der Netze sollte auch die Absenkungsgeschwindigkeit angepaßt werden, was praktisch unmöglich ist, da man eine quantitative Schätzung darüber bräuchte, wie stark das neue Netz sich verändert hat. Das Beispiel aus Abbildung 1.1a zeigt wiederum, daß keine der Methoden signifikant bessere Ergebnisse liefert (Tabelle 2.2). Der t-Test auf Gleichheit der Mittelwerte liefert 0.56 (Schwelle $t_{0.95;50} = 1.68$). Die Abbildungen 2.10 und 2.11 zeigen den Trainings- und Testfehler zu einem Trainingslauf. Die beiden Verfahren kommen zu einem ähnlichen Resultat bezüglich des Fehlers. In Abbildung 2.10 erkennt man an den Sprüngen jeweils den Zeitpunkt, zu dem die Anpassung der Hyperparameter stattfand, da sich an diesen Stellen die Fehlerfunktion ändert. Man kann schlußfolgern, daß es den Vorlieben des Anwenders überlassen werden kann, für welche Variante er sich entscheiden möchte.

2.6 Zusammenfassung

In diesem Kapitel wurde der Funktionsapproximator vorgestellt, der in meinem Ansatz verwendet wird. Dabei habe ich zuerst unabhängig von neuronalen Netzen verschiedene Fehlerfunktionen vorgestellt, die man erhält, wenn man Annahmen über die zugrundeliegenden Daten macht. Ausgehend von der quadratischen Fehlerfunktion erhält man die Bias-Varianz Dekomposition des Fehlers, an der man ablesen kann, daß es sinnvoll ist, mit mehreren verschiedenen Datensätzen zu trainieren und darüber zu mitteln. Ebenso erkennt man daran, daß Bias und Varianz sich in aller Regel komplementär verhalten, was wiederum die Verwendung von Regularisierung begründet. Nachdem die Fehlerfunktion festliegt, benötigt man noch ein Verfahren, um die Parameter des Netzes einzustellen. Es wurde ausgeführt, daß dazu in jedem Fall ein geeignetes Lernverfahren eingesetzt werden muß.

Das Bayes'sche Lernen bettet das Training neuronaler Netze in einen wahrscheinlichkeitstheoretischen Kontext ein. Dadurch macht man sich die Annahmen, die dem Training zugrunde liegen, explizit bewußt. Das Verfahren erlaubt es, die Gewichtung des Regularisierungstermes adaptiv zu berechnen, und zwar nur anhand der Trainingsdaten. Ferner ergibt

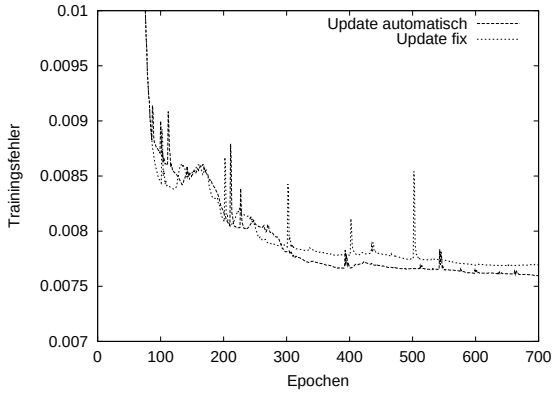


Abbildung 2.10: Verlauf des Trainingsfehlers (pro Muster) für ein ausgewähltes Netz. Es ist nur der Fehler auf den Daten E_D angegeben. An den Sprüngen kann man jeweils erkennen, wann die Anpassung der Hyperparameter erfolgte. Für den automatischen Update ist das erwartungsgemäß unregelmäßiger, abhängig davon, wie schnell die Schwelle δ abgesenkt wird.

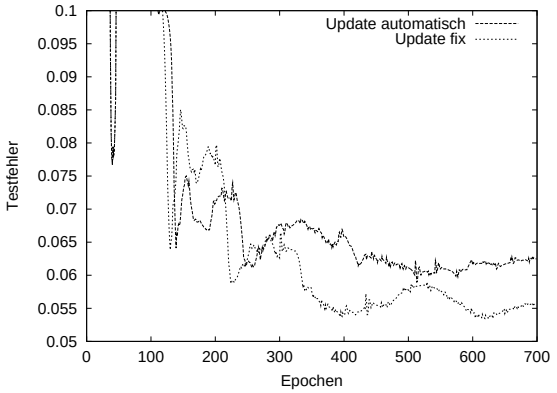


Abbildung 2.11: Verlauf des Testfehlers entsprechend zu dem linken Bild. Beide Verfahren kommen zu einem ähnlichen Fehler am Ende.

die Betrachtung auf einer höheren Stufe ein Vergleichskriterium, die Modellevidenz, für verschiedenartige Netze. Dieses Kriterium wird später in dem evolutionären Suchverfahren als Fitneßwert verwendet. In meinen Untersuchungen zu der Bayes'schen Methode kam ich zu dem Ergebnis, daß

- bei geringer Anzahl an Daten bzw. bei geringer Anzahl an versteckten Neuronen es sinnvoll ist, nur eine Gewichtsgruppe zu bilden,
- die adaptive Regularisierung nicht mit jedem Verfahren zur Parameteroptimierung verträglich ist, insbesondere mit solchen, die bereits implizit eine Art Regularisierung durchführen, die aber im Bayes'schen Kontext nicht berücksichtigt wird,
- die Frage, wann man die Hyperparameter anpaßt, ein zusätzlicher, wenn auch unkritischer Parameter des Bayes'schen Lernens ist.

Kapitel 3

Komitees

Neben Verfahren, die die Generalisierungsfähigkeit eines einzelnen neuronalen Netzes verbessern, läßt sich die Leistungsfähigkeit eines Systems dadurch steigern, daß man ein Komitee von mehreren Netzen verwendet (Perrone & Cooper, 1994, Bishop, 1995, Krogh & Vedelsby, 1995), was auch experimentell immer wieder bestätigt wurde (Perrone & Cooper, 1994, Gutjahr, 1996, Ragg & Gutjahr, 1997b). Die Gründe für eine Leistungssteigerung können verschiedener Natur sein. Insbesondere scheint es lohnenswert, diese gemeinsam zu betrachten und in ein Optimierungskonzept zu integrieren.

3.1 Vorteile der Komiteebildung

Eine wesentliche Einsicht in die Vorteile der Komiteebildung läßt sich gewinnen, wenn man die Korrelationen der Abweichungen $\epsilon_i(x)$ der Komiteemitglieder $y_i(x)$ vom gesuchten Prozeß $h(x)$ betrachtet:

$$\epsilon_i(x) = y_i(x) - h(x).$$

Unter der Voraussetzung, daß die Abweichungen Mittelwert 0 haben und unkorreliert sind, d.h. der Erwartungswert $\mathcal{E}[\epsilon_i \epsilon_j] = 0$ ist, kann man zeigen, daß sich der Fehler E_{COM} eines Komitees aus L Netzen bis auf $\frac{1}{L} E_{AV}$ reduzieren läßt, wenn E_{AV} der durchschnittliche Fehler der einzelnen Netze ist. Mit der angenommenen Unkorreliertheit und der Definition des Komitees durch

$$y_{COM}(x) := \frac{1}{L} \sum_{i=1}^{L} y_i(x) \tag{3.1}$$

folgt sofort aus dem Vergleich des durchschnittlichen Fehlers

$$E_{AV}(x) = \frac{1}{L} \sum_{i=1}^{L} \mathcal{E}[\epsilon_i^2]$$

mit dem Komiteefehler

$$\begin{aligned} E_{Kom} &= \mathcal{E}\left[\left(\frac{1}{L}\sum_i \epsilon_i\right)^2\right] \\ &= \frac{1}{L^2}\sum_i \mathcal{E}[\epsilon_i^2] \end{aligned}$$

die Behauptung. Andererseits kann man zeigen, daß die Komiteebildung zumindest keine Verschlechterung bewirkt, wenn diese Voraussetzungen nicht gegeben sind. Mit der Cauchy-Schwarz'schen Ungleichung (Heuser, 1986)

$$\left(\sum_{i=1}^{L} \epsilon_i\right)^2 \leq L \sum_{i=1}^{L} \epsilon_i^2$$

folgt sofort

$$E_{COM} \leq E_{AV}. \tag{3.2}$$

Die Ungleichung $E_{COM} \leq E_{AV}$ gilt für beliebige Fehlerfunktionen E, solange E konvex ist, d.h.

$$E(\sum_i a_i y_i) \leq \sum_i a_i E(y_i)$$

mit $\sum_i a_i = 1$. Mit der Jensen-Ungleichung (Cover & Thomas, 1991)

$$E(g(x)) \geq g(E(x))$$

folgt die Behauptung direkt aus der Definition von E_{COM} und E_{AV}.

Betrachtet man statt der Abweichungen vom gesuchten Prozeß, die ja für einen Algorithmus, wie er hier entwickelt werden soll, nicht zur Verfügung stünden, die Ausgabefunktionen y_i direkt, so kann man für Klassifikationsprobleme mittels des Gesetzes der großen Zahlen ein ähnliches Ergebnis erzielen (Gutjahr, 1996): Ein Klassifikator als Komitee von unendlich vielen, stochastisch unabhängigen neuronalen Netzen, von denen jedes mit Wahrscheinlichkeit $p > 0.5$ richtig klassifiziert, macht nie eine falsche Entscheidung. In der Praxis wird man sich aber damit begnügen müssen, daß die Leistung des Komitees zwar besser wird als die Leistung der einzelnen Netze, aber aufgrund der gemachten Annahmen deutlich von einem perfekten Klassifikator verschieden ist, da die stochastische Unabhängigkeit der Komiteemitglieder für viele Netze nicht ohne weiteres zu bewerkstelligen ist.

Den Fehler des Komitees kann man in ähnlicher Weise zur Bias-Varianz Dekomposition (Kapitel 2.2) in zwei Terme aufspalten. Dadurch gewinnt man eine wichtige Einsicht in die

Verbesserung der Generalisierungsfähigkeit (Krogh & Vedelsby, 1995, Bishop, 1995). Ich werde die Gleichung ausführlich herleiten, da sie mir auch den Ausgangspunkt liefert, um ein analytisches Kriterium zur Auswahl der Komiteemitglieder zu entwickeln und meine Vorgehensweise mathematisch zu begründen.

Gesucht ist ein Zusammenhang zwischen dem durchschnittlichen Fehler des Komitees und dem durchschnittlichen Fehler der einzelnen Netze, den man ausgehend vom Erwartungswert des Komiteefehlers entwickeln kann. Die Abhängigkeiten der Funktionen von $\mathbf{x}$ werde ich in der Rechnung der Übersichtlichkeit halber weglassen. Für die quadratische Fehlerfunktion gilt

$$
\begin{aligned}
\mathcal{E}\left[(y_{COM}-h)^2\right] &= \mathcal{E}\left[(h-y_{COM})^2\right] \\
&= \mathcal{E}[h^2 - 2hy_{COM} + y_{COM}^2] \\
&= -2\mathcal{E}[hy_{COM}] + \mathcal{E}[h^2] + \mathcal{E}[y_{COM}^2] \\
&= -2\mathcal{E}[hy_{COM}] + \mathcal{E}[h^2] + 2\mathcal{E}[y_{COM}^2] - \mathcal{E}[y_{COM}^2] \\
&= -2\mathcal{E}\left[h\frac{1}{L}\sum_{i=1}^{L} y_i\right] + \mathcal{E}\left[\frac{1}{L}\sum_{i=1}^{L} h^2\right] + 2\mathcal{E}\left[y_{COM}\frac{1}{L}\sum_{i=1}^{L} y_i\right] - \mathcal{E}\left[\frac{1}{L}\sum_{i=1}^{L} y_{COM}^2\right] \\
&= \frac{1}{L}\sum_{i=1}^{L}\left(-2\mathcal{E}[hy_i] + \mathcal{E}[h^2] + 2\mathcal{E}[y_i y_{COM}] - \mathcal{E}[y_{COM}^2]\right) \\
&= \frac{1}{L}\sum_{i=1}^{L}\Big((\mathcal{E}[y_i^2] - 2\mathcal{E}[hy_i] + \mathcal{E}[h^2]) \quad - (\mathcal{E}[y_i^2] - 2\mathcal{E}[yy_COM] + \mathcal{E}[y_{COM}^2])\Big) \\
&= \frac{1}{L}\sum_{i=1}^{L}\mathcal{E}\left[(y_i(\mathbf{x}) - h(\mathbf{x}))^2\right] - \frac{1}{L}\sum_{i=1}^{L}\mathcal{E}\left[(y_i(\mathbf{x}) - y_{COM}(\mathbf{x}))^2\right]. \qquad (3.3)
\end{aligned}
$$

Bei der Herleitung wurde die Beziehung $\sum_{i=1}^{L}\frac{1}{L} = 1$ ausgenutzt. Die Dekomposition bleibt damit auch für beliebige Gewichtungen des Komitees

$$
y_{COM}(x) = \sum_{i=1}^{L} w_i y_i(x) \qquad (3.4)
$$

gültig, solange die Beziehung $\sum_{i=1}^{L} w_i = 1$ erfüllt ist.

Im folgenden werde ich die Formel etwas genauer analysieren. Der erste Term der Formel hängt nur von den Fehlern der einzelnen Netze ab, während der zweite in schöner Analogie zur Bias-Varianz Dekomposition die Streuung der Netze mißt. Bleibt jetzt der durchschnittliche Fehler konstant, dann wird der Komiteefehler kleiner, wenn die Streuung größer wird. Eine Methode zur Komiteebildung muß also einerseits versuchen, möglichst verschiedenartige Netze zu finden, andererseits aber auch dafür sorgen, daß die Netze möglichst gut sind. Auf den zweiten Term von Gleichung (3.3) werde ich später wieder zurückkommen, um daraus ein Optimierungskriterium zu entwickeln.

Für die Cross-Entropy Fehlerfunktion ist bisher keine Dekomposition des Komiteefehlers bekannt. Aufgrund der logarithmischen Abhängigkeit ist diese auch nicht auf direktem Wege zu gewinnen. Es gilt aber zumindest allgemein die Abschätzung

$$E_{CE}(y,h) \leq E_Q(y,h) \tag{3.5}$$

Mit der Ungleichung von Jensen (Cover & Thomas, 1991, Henze, 1995) und $\ln(1+x) \leq x$ für $x \geq 0$ läßt sich die logarithmische Fehlerfunktion nach oben abschätzen:

$$\begin{aligned} E_{CE}(y,h) &= -\sum_n h^n \ln y^n + (1-h^n)\ln(1-y^n) \\ &\leq -\sum_n \ln\left(h^n y^n + (1-h^n)(1-y^n)\right) \\ &= -\sum_n \ln\left(1 + 2h^n y^n - h^n - y^n\right) \\ &\leq -\sum_n 2h^n y^n - h^n - y^n. \end{aligned} \tag{3.6}$$

Subtrahiert man E_{CE} von E_Q, dann erhält man

$$\begin{aligned} E_Q(y,h) - E_{CE}(y,h) &= \sum_n (y-h)^2 - \left(-\sum_n 2hy - h - y\right) \\ &= \sum_n y^2 - 2hy + h^2 - \sum_n 2hy - h - y \\ &= \sum_n y^2 + h^2 - h - y \\ &= \sum_n y(y-1) + h(h-1) \\ &\geq 0. \end{aligned} \tag{3.7}$$

wobei ich die Abhängigkeit von n der Übersichtlichkeit halber weggelassen habe. Minimiert man nun Gleichung (3.3), dann wird auch der logarithmische Fehler des Komitees klein.

3.1.1 Ungenaue Gütekriterien und Datenabhängigkeit

Ein wesentlicher Grund für die bessere Generalisierung von Komitees ist, daß die Leistung eines Netzes nicht genau bestimmt werden kann. Schätzt man die Güte durch den Fehler auf einer unabhängigen Datenmenge, dann variiert die Leistung mit der zufälligen Verteilung des Rauschens in den Daten. Unterscheidet sich die Verteilung von der, die auf den Trainingsdaten gegeben ist, dann verzerrt diese Zufallskomponente das Gütemaß.

Ebenso ist die Evidenz, die im Bayes'schen Ansatz als Gütemaß berechnet wird, aufgrund idealisierender Annahmen kein perfekter Indikator für die Leistung der Netze, sondern bestenfalls damit korreliert. Abbildung 2.5 verdeutlichte die Korrelation von Evidenz und

Testfehler an einem Beispiel. Die Abhängigkeit von der Ungenauigkeit der Schätzung läßt sich reduzieren, wenn man über mehrere Netze mittelt.

Andererseits liegt durch die Wahl einer Trainingsmenge ebenso eine bestimmte Verteilung des Rauschens fest, die mehr oder weniger gut mit den Annahmen des Trainingsprozesses übereinstimmen kann. Experimentelle Untersuchungen in Kapitel 5 werden zeigen, daß dies einen gewichtigen Einfluß auf die Modellauswahl haben kann. Bei der Vorstellung des Bias-Varianz Dilemmas (Kapitel 2.2) wurde ausgeführt, daß auch aus diesem Grund eine Mittelung über mehrere Netze sinnvoll ist.

3.1.2 Bayes'sches Lernen und Komitees

Im Bayes'schen Ansatz wird jedem Gewichtsvektor $\mathbf{w}$ eine Wahrscheinlichkeit zugeordnet, daß er ein Modell der Daten ist. Trainiert man jetzt mehrere Modelle und findet lokale Minima m_i, dann kann man die a posteriori Verteilung durch die Summe der zugehörigen Gaußverteilungen des Bayes'schen Ansatzes modellieren. Man erhält also

$$\begin{aligned} p(\mathbf{w}|D) &= \sum_i p(m_i, \mathbf{w}|D) \\ &= \sum_i p(\mathbf{w}|m_i) P(m_i, D). \end{aligned}$$

Diese Approximation der a posteriori Verteilung verwendet man jetzt anstatt der einfacheren Variante, um andere Größen zu berechnen, die von Interesse sind. Es sei betont, daß sich durch die Summe von Normalverteilungen die tatsächliche Verteilung der Gewichte, die multimodal sein kann, besser approximieren läßt als durch eine einzelne Normalverteilung (Bishop, 1995, Barber & Bishop, 1998). Dieser Umstand gilt nicht nur für die Integration über mehrere Initialisierungen, sondern für jede andere Form der Komiteebildung ebenso. Im hier entwickelten Ansatz integriere ich zusätzlich über verschiedene Trainingsdatensätze, verschiedende Eingabevektoren und Netztopologien.

3.1.3 Große Eingaberäume

Ein weiterer Vorzug der Komiteebildung ist dadurch gegeben, daß eine große Anzahl von möglichen Eingabekomponenten aufgeteilt werden kann, so daß die einzelnen Netze verschiedenartige Sichten auf das Problem haben und potentiell verschiedene Lösungen lernen können. Durch die Reduktion auf niedrigdimensionale Eingabevektoren wird im Falle einer begrenzten Datenmenge erst eine 'vernünftige' Abbildung gelernt werden (Baum & Haussler, 1989, White, 1989, Bishop, 1995). Andererseits läßt sich der Eingaberaum unter Umständen so aufteilen, daß mehrere gleichwertige, aber evtl. stochastisch unabhängige Netze bestimmt werden. Durch die Komiteebildung zieht man aber trotzdem Nutzen aus allen Eingabemerkmalen.

Wenn keine mögliche Einteilung vorgegeben ist, dann kann man in der Praxis so vorgehen, daß man eine Population von Netzen initialisiert, jedes mit einer zufälligen Eingabestruktur. Mittels statistischer Kriterien verändert man dann die Eingabestruktur schrittweise, wobei die besseren Netze beibehalten werden. In Kapitel 4 und insbesondere 4.5 wird die Optimierung der Eingabestruktur wieder aufgegriffen.

Es sei noch angemerkt, daß man, falls das Kriterium zur Bewertung der Eingabestruktur vom gewählten Modelltyp unabhängig ist, dieses nutzen kann, um die einzelnen Eingaben nach ihrer Wichtigkeit zu sortieren. Statt einer Zufallsauswahl lassen sich so die Suchpunkte zu Beginn der Suche wesentlich günstiger auswählen. In Kapitel 5 wird zur Wirkungsweise der geeigneten Initialisierung ein Experiment durchgeführt.

3.2 Gewinnung und Kombination der Komiteemitglieder

Soll die Bildung eines Komitees über die Mittelung vieler trainierter Modelle hinausgehen, benötigt man ein Auswahlkriterium für die Modelle. Bisher gibt es keine brauchbare Theorie, aus der sich ein Algorithmus ableiten läßt. Eine häufig angewendete Vorgehensweise versucht möglichst verschiedenartige, aber trotzdem gut generalisierende Modelle zu finden, die mittels heuristischer Selektionskriterien, unter anderem auch Cross-Validierung, zu einem Komitee verbunden werden (Sharkey, 1999). Die bekannten Methoden zur Generierung und zur Auswahl von Modellen für ein Komitee sollen im folgenden kurz zusammengefaßt werden. Grundsätzlich lassen sich die folgenden Methoden unterscheiden, die sich nicht unbedingt gegenseitig ausschließen (Sharkey, 1999):

- Aufteilung der Daten zum Training der Modelle
- Variation der Struktur bzw. Topologie der Modelle und der Informationsquellen, d.h. der verwendeten Merkmale
- Integration über verschiedene Initialisierungen
- Einbeziehen der Abhängigkeiten zwischen den Modellen, z.B. durch Bestrafung starker Kovarianzen der Ausgabefunktionen.

3.2.1 Aufteilung der Daten

Aufteilung der Daten ist eine der am häufigsten verwendeten Methoden. Sie liegt dem sogenannten *Bagging* (Breiman, 1996) und auch *Boosting* (Freund & Schapire, 1996, Schapire *et al.*, 1997) zugrunde, beides Methoden, die mit Erfolg eingesetzt wurden.

Bagging verwendet ein einfaches Bootstrapping, d.h. es wird für jedes zu trainierende Modell ein bestimmter Teil der Muster durch Ziehen mit Zurücklegen ausgewählt. Alle Modelle erhalten dann das gleiche Gewicht bei der Kombination, d.h. es wird einfach der Mittelwert gebildet. Beim Boosting-Algorithmus ist das Sampling etwas komplizierter, und zwar werden zu Anfang alle Muster mit gleicher Wahrscheinlichkeit ausgewählt, eine Teilmenge der

Trainingsmenge gezogen und ein Modell trainiert. Die Auswahlwahrscheinlichkeit der Muster mit großem Fehler wird dann erhöht bzw. bei kleinem Fehler erniedrigt. Für das Modell wird noch, basierend auf dem Fehler, ein Gewichtungsfaktor bestimmt. Dieser Schritt wird n mal wiederholt. Das Komitee besteht am Ende aus den n Modellen, die jedes ihr spezielles Gewicht berechnet haben (Freund & Schapire, 1996).

Die starke Konzentration auf Muster mit großem Fehler macht das Verfahren offensichtlich anfällig für Ausreißer. Ist der Datensatz beispielsweise mit einem Rauschen behaftet wie in Abbildung 1.1b in der Einleitung dargestellt, dann muß dieses Vorgehen zwangsläufig zu Overfitting führen. Daß dem tatsächlich so ist, haben (Rätsch *et al.*, 1998) nachgewiesen. Boosting gehört zu den *hard margin* Algorithmen, die sich auf die schwierigsten Muster konzentrieren, ohne aber ein Modell für das Rauschen in den Daten zu verwenden. Durch Erweiterung um einen Regularisierungsterm, diesmal zur Musterauswahl, können diese Nachteile behoben werden. Bei zunehmendem Rauschen konvergieren die Auswahlwahrscheinlichkeiten wieder zur selben Verteilung wie bei Bagging.

Betrachtet man das Verfahren im Lichte der Dekompositionsgleichung (3.3), dann sieht man, daß Boosting die Varianz unter den Komiteemitgliedern vergrößert, der durchschnittliche Fehler der Netze wird aber nur konstant bleiben oder kleiner werden, wenn die Daten nicht verrauscht sind. Die Konzentration auf wenige Muster mit großem Fehler kann ohne geeignete Regularisierung also fatal sein.

3.2.2 Merkmale und Netzkomplexität

Netze mit verschiedenen Informationen zu trainieren, ist besonders dann interessant, wenn viele Informationsquellen zur Verfügung stehen, aus denen man nur schwer eine Auswahl treffen kann. Für die Prognose von Zeitreihen lassen sich hier verschiedene Aspekte der Daten abdecken: langfristige Information, kurzfristige Trends, fundamentale Daten oder technische Kennzahlen. Für diesen Einsatzbereich konnten hiermit Erfolge erzielt werden (Gutjahr, 1996, Ragg & Gutjahr, 1997b).

Eine Integration über Modelle verschiedener Komplexität durchzuführen, läßt sich wieder besonders schön im Bayes'schen Ansatz begründen, und zwar dadurch, daß jedes Modell $\mathcal{H}_i$ wieder eine gewisse a posteriori Wahrscheinlichkeit hat. Die natürliche Behandlung von unbekannten Parametern im Bayes'schen Ansatz ist die Integration über den ganzen Parameterraum. Das heißt, bei gegebenen Daten D gewinnt man die Verteilung einer Größe Q im Bayes'schen Ansatz durch Summation über die Modelle

$$\begin{aligned} p(Q|D) &= \sum_i p(Q, \mathcal{H}_i|D) \\ &= \sum_i p(Q|D) P(\mathcal{H}_i, D). \end{aligned}$$

In einer extremen Form gelangt man zu dem Ansatz 'Mixture of Experts' (Jordan & Jacobs, 1994, Bishop, 1995), bei dem die Ausgaben nicht mehr integriert werden, sondern jeweils

ein Experte für ein neues Muster ausgewählt wird. Die größte Fehlerquelle ist hier das Auswahlverfahren, das für jeden Experten seine Zuständigkeit bzw. Nicht-Zuständigkeit für ein bestimmtes Muster erkennen können muß. Für eine 'weiche' Variante dieser Methode kann man die Varianz der Netzausgabe verwenden, die im Bayes'schen Ansatzes berechnet wird (Gleichung (2.43)). Über sie läßt sich gerade eine Art 'Arbeitsbereich' der Modelle definieren. Auf diesen Aspekt werde ich in Kapitel 6 zurückkommen.

3.2.3 Korrelation der Netze

Es wurde schon dargelegt, daß unkorrelierte Fehlerfunktionen einen hohen Zugewinn an Leistung versprechen. Abweichend von den obigen Kriterien hat Rosen einen Ansatz vorgestellt, der die Korrelation der Netze in der Fehlerfunktion beim Training berücksichtigt (Rosen, 1996), vgl. auch (Liu & Yao, 1998). Es wird direkt ein Komitee von Netzen trainiert. Die prinzipielle Kritik an diesem Ansatz ist, daß die einzelnen Modelle nicht 'lokal optimal' sein müssen, sondern vielmehr beliebige Größen für die Gewichte und damit auch weniger glatte Ausgabefunktionen generiert werden können. Erst die Komiteebildung sorgt durch die Integration wieder für eine Glättung. Rosen gewichtet den Strafterm genauso stark wie den Fehlerterm. Ein experimenteller oder ein Bayes'scher Ansatz zur Gewichtung des Strafterms wäre aufgrund der Anzahl der Netze wesentlich schwerer, wenn überhaupt, zu bewerkstelligen. Für die verwendete Gewichtung fehlt aber eine geeignete Begründung.

Folgerichtig versucht der Ansatz von (Rosen, 1996) der Korrelation starkes Gewicht einzuräumen. Die Gleichung (3.3) zeigt aber auch hier die Problematik: Die Streuung der Netze wird maximiert, aber die Generalisierungsfähigkeit der einzelnen Modelle wird nicht ausreichend berücksichtigt. Hierzu müßte also neben dem Term, der die Korrelationen bestraft, auch noch ein Regularisierungsterm, der die Komplexität der einzelnen Modelle berücksichtigt, verwendet werden. Man hätte hier also drei Hyperparameter einzustellen. Für ein Bayes'sches Verfahren zur automatischen Gewichtung der einzelnen Terme sehe ich keinen sinnvollen Anhaltspunkt.

Eine a posteriori Optimierung auf Basis der linearen Abhängigkeiten schlägt Hashem vor (Hashem, 1999). Der Entwurfsprozeß für die Netze wird nicht explizit in Betracht gezogen, sondern es wird von trainierten Modellen ausgegangen. Unter Zuhilfenahme einer zusätzlichen Cross-Validierungsmenge wird nun das Komitee optimiert, indem zuerst gemäß der Leistung der einzelnen Netze auf der Cross-Validierungsmenge die Gewichte im Komitee eingestellt werden. Ist das so gebildete Komitee besser als das beste einzelne Netz und als das Komitee mit identischen Gewichten (einfacher Durchschnitt), dann terminiert der Algorithmus mit diesem Komitee. Andernfalls wird versucht, durch Entfernen von Netzen mit starken linearen Abhängigkeiten dieses Kriterium zu erfüllen. Mißlingt das, dann wird je nach Leistung auf der Cross-Validierungsmenge das beste Netz ausgewählt oder das Komitee mit identischen Gewichten.

Die prinzipielle Kritik an diesem Ansatz zielt auf die iterative Verwendung einer zusätzlichen Datenmenge. Zum einen steht und fällt das Verfahren mit der geeigneten Wahl dieser Menge, zum anderen führt die iterative Verwendung der Daten zum Bestimmen von Parametern wie im Falle von Boosting zu Overfitting bei verrauschten Daten. Ohne zusätzliche *Glattheitsbedingung*, d.h. ohne einen geeigneten Regularisierungsterm, ist dieses Vorgehen

wenig empfehlenswert. Außerdem wird ein möglicher Zuwachs an Leistung verschenkt, da beim Entwurfsprozeß der Modelle nicht darauf geachtet wird, daß diese ein Komitee bilden sollen.

3.3 Konsequenzen

Die bisher vorgestellten Methoden zeigen deutlich, daß es mehrere Ansatzpunkte gibt, die Generalisierungsleistung eines neuronalen Systems zu optimieren, von denen jeder seine Berechtigung hat. Ein wesentliches Ziel meiner Arbeit ist es, diese Entwurfskriterien in ein einziges Optimierungskonzept zu integrieren.

Die Bias-Varianz Dekomposition hat gezeigt, daß es bei verrauschten Daten unerläßlich ist, über mehrere Modelle zu integrieren, um die Abhängigkeit von den gewählten Trainingsdaten zu minimieren. Die Ausführungen zum *Fluch der Dimensionen* zeigen die Notwendigkeit, die Größe des Eingaberaumes im richtigen Verhältnis zur Zahl der Daten zu wählen. Die Tatsache, daß es sich beim *Lernen aus Daten* um ein *schlecht gestelltes Problem* handelt, macht die Verwendung von Regularisierungsmethoden zwingend. Der Aufwand für geeignete Gewichtung des Regularisierungsterms ist mit heuristischen Methoden nicht mehr zu leisten, wenn man die einzelnen Modelle mit verschiedenen Daten trainiert. Abgesehen davon wird durch Cross-Validierung eine neue Abhängigkeit von der Datenauswahl eingeführt. Die Bayes'sche Methode zum Training neuronaler Netze ist also ein unverzichtbarer Baustein eines Optimierungskonzeptes.

Im meinem Ansatz möchte ich vermeiden, daß durch die Bedingung auf der höheren Ebene - die Unabhängigkeit der Netze - die Parameteroptimierung auf der unteren Ebene - das Trainieren mit geeigneter Regularisierung - verändert wird, wie das Gleichung (3.3) auch nahelegt. Vielmehr muß der Suchprozeß auf der höheren Ebene so gestaltet sein, daß er möglichst unabhängige Netze findet, von denen aber trotzdem jedes für seine Trainingsmenge ein 'gutes' Modell darstellt. Dazu muß man das Kriterium der Unabhänigkeit in geeigneter Weise in den Suchprozeß einbringen. Im folgenden werde ich hierzu meinen Ansatz entwickeln, der dann im nächsten Kapitel in die evolutionäre Suche eingebunden wird.

3.4 Ein analytisches Optimierungskriterium

Statt Cross-Validierungs Techniken in den Suchprozeß einzubringen, muß es das Ziel sein, den Entwicklungsprozeß weitgehend auf der Optimierung analytischer Kriterien aufzubauen. Dazu lohnt es sich, den zweiten Term aus Gleichung (3.3) näher zu untersuchen, der die Streuung der Netze mißt.

$$\begin{aligned}\frac{1}{L}\sum_{i=1}^{L}\mathcal{E}\left[(y_i(\mathbf{x})-y_{COM}(\mathbf{x}))^2\right] &= \frac{1}{L}\sum_{i=1}^{L}\mathcal{E}[y_i^2(\mathbf{x})]-\frac{2}{L}\sum_{i=1}^{L}\mathcal{E}[y_{COM}(\mathbf{x})y_i(\mathbf{x})] \\ &\quad +\frac{1}{L}\sum_{i=1}^{L}\mathcal{E}[y_{COM}^2(\mathbf{x})] \qquad (3.8)\end{aligned}$$

Der Term soll maximiert werden. Dazu bringe ich den zweiten und dritten Summand in eine neue Form, die sich dann besser interpretieren läßt.

$$\begin{aligned}
-\frac{2}{L}\sum_{i=1}^{L}\mathcal{E}[y_{COM}(\mathbf{x})y_i(\mathbf{x})] &= -\frac{2}{L}\sum_{i=1}^{L}\mathcal{E}\left[\frac{1}{L}\sum_{j=1}^{L}y_j y_i\right] \\
&= -\frac{2}{L^2}\sum_{i=1}^{L}\sum_{j=1}^{L}\mathcal{E}[y_j y_i] \\
&= -\frac{2}{L^2}\left(2\sum_{i=1}^{L}-1\sum_{j=i+1}^{L}\mathcal{E}[y_j y_i]+\sum_{i=1}^{L}\mathcal{E}[y_i^2]\right) \\
&= -\frac{4}{L^2}\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\mathcal{E}[y_j y_i]-\frac{2}{L^2}\sum_{i=1}^{L}\mathcal{E}[y_i^2]. \qquad (3.9)
\end{aligned}$$

Dabei habe ich über die Diagonalelemente extra summiert und die Beziehung $y_j y_i = y_i y_j$ ausgenutzt. Für den dritten Term erhält man mit Gleichung (3.1) und Ausmultiplizieren

$$\begin{aligned}
\frac{1}{L}\sum_{i=1}^{L}\mathcal{E}[y_{COM}^2] &= \mathcal{E}[y_{COM}^2] \\
&= \mathcal{E}\left[\left(\frac{1}{L}\sum_{i=1}^{L}y_i\right)\left(\frac{1}{L}\sum_{j=1}^{L}y_j\right)\right] \\
&= \mathcal{E}\left[\frac{1}{L^2}\left(\sum_{i=1}^{L}y_i^2+2\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}y_i y_j\right)\right] \\
&= \frac{1}{L^2}\sum_{i=1}^{L}\mathcal{E}[y_i^2]+\frac{2}{L^2}\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\mathcal{E}[y_i y_j]. \qquad (3.10)
\end{aligned}$$

Setzt man die beiden Ausdrücke 3.9 und 3.10 wieder in Gleichung 3.8 ein und berücksichtigt noch $\mathcal{E}[XY] = \mathcal{E}[X]\mathcal{E}[Y] + Cov(X,Y)$, eine Eigenschaft der Kovarianz (Henze, 1997), dann verbleibt man mit

$$\begin{aligned}
&\frac{1}{L}\sum_{i=1}^{L}\mathcal{E}[y_i^2]-\frac{2}{L^2}\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\mathcal{E}[y_j y_i]-\frac{1}{L^2}\sum_{i=1}^{L}\mathcal{E}[y_i^2] \\
&= \frac{L-1}{L^2}\sum_{i=1}^{L}\mathcal{E}[y_i^2]-\frac{2}{L^2}\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\mathcal{E}[y_j y_i] \\
&= \frac{L-1}{L^2}\sum_{i=1}^{L}(\mathcal{E}[y_i])^2+\frac{L-1}{L^2}\sum_{i=1}^{L}Cov(y_i,y_i) \\
&\quad -\frac{2}{L^2}\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\mathcal{E}[y_j]\mathcal{E}[y_i]-\frac{2}{L^2}\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}Cov(y_i,y_j). \qquad (3.11)
\end{aligned}$$

Die Erwartungswerte von y_i hängen davon ab, mit welchen Daten das spezielle Modell trainiert wurde, da der Mittelwert $\bar{y}_i$ nach dem Training gerade dem Mittelwert der Zielwerte entsprechen sollte. Den ersten und dritten Summanden kann man also als nahezu konstant ansehen. Da die Kovarianz einer Zufallsvariablen mit sich selbst gerade ihre Varianz ergibt, $Cov(X, X) = V(X)$, sagt der zweite Summand aus, daß die Varianz der Netze maximal groß sein sollte, während der letze Summand aufgrund des Minuszeichens möglichst klein werden sollte. Die Größe der Kovarianz $Cov(y_j(\mathbf{x}), y_i(\mathbf{x}))$ hängt aber davon ab, wie stark die stochastische Abhängigkeit zwischen y_j und y_i ist. Aus der Multiplikationsformel für Erwartungswerte folgt, daß die Kovarianz $Cov(X, Y)$ zweier Zufallsvariablen X und Y gerade dann 0 ist, wenn diese stochastisch unabhängig sind und maximal, wenn $X = Y$ gilt (Henze, 1997). Damit sollte man, bei sonst gleichen Bedingungen, Netze bevorzugen, die stochastisch weniger abhängig sind und maximale Varianz haben. Statt den Term 3.11 zu maximieren, kann man damit ein äquivalentes Kriterium angeben, das aber den Vorteil hat, im Falle von Klassifikationsproblemen ebenfalls direkt anwendbar zu sein (siehe die Aussagen zu Klassifikationsproblemen in Kapitel 3.1 und die Ausführungen in (Gutjahr, 1996)).

Setzen wir für den Moment voraus, wir hätten ein Maß $I(X, Y)$ für die stochastische Unabhängigkeit[1] zweier Zufallsvariablen bzw. Funktionen X und Y, dann läßt sich aus Gleichung (3.11) das folgende Optimierungskriterium, ein Heterogenitätskriterium Het_{COM}, ableiten:

$$Het_{COM} \quad := \quad \frac{L-1}{L^2} \sum_{i=1}^{L} I(y_i, y_i) - \frac{2}{L^2} \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} I(y_i, y_j) \longrightarrow \max . \tag{3.12}$$

Damit habe ich ein analytisches Kriterium definiert, das man algorithmisch optimieren kann, um aus einer gegebenen Menge von trainierten Netzen diejenigen auszuwählen, die das beste Komitee formen (Ragg, 2000). Es ist bereits für kleine L unmöglich, alle $\binom{L}{k}$ Kombinationen für alle k zu betrachten, um das Kriterium zu optimieren. Eine Möglichkeit, den Aufwand zu reduzieren und trotzdem eine gute Kombination von Funktionen zu finden, ist, die trainierten Netze zuerst geeignet in Gruppen zusammenzufassen und dann aus jeder Gruppe einen Vertreter auszuwählen. Ein Verfahren hierzu wird im nächsten Abschnitt entwickelt.

3.5 Clusterung von neuronalen Netzen

Die einfachste und schnellste Methode, Elemente basierend auf ihrer Ähnlichkeit in Klassen zusammenzufassen, sind agglomerative, hierarchische Clusterverfahren. Die Idee dabei ist folgende: Man ordnet zuerst jedes Netz in eine eigene Gruppe ein und legt dann sukzessive jeweils die beiden Gruppen zusammen, die sich am ähnlichsten sind. Um die neuronalen Modelle zu clustern, benötigen wir also noch ein Maß für die Ähnlichkeit zweier Netze.

In Kapitel 3.6.2 werden wir sehen, daß Mutual Information die Kullback-Leibler Distanz zwischen der gemeinsamen Verteilung und der Produktverteilung zweier Zufallsvariablen ist.

[1] in Kapitel 3.6 wird es definiert und ein Schätzverfahren dafür angegeben

Mittels der Kullback-Leibler Distanz läßt sich aber keine Metrik definieren, da die Dreiecksungleichung nicht erfüllt ist (Cover & Thomas, 1991). Die Ähnlichkeitsbeziehungen zwischen den neuronalen Netzen liegen also nur als Matrix vor. Sie lassen sich nicht als Punkte in einem euklidischen Raum darstellen. Damit scheiden alle Clusterverfahren aus, die auf einer Metrik basieren.

Möchte man eine möglichst optimale Klasseneinteilung finden, dann hat man hier ein ähnliches Problem zu lösen wie bei der Aufgabe, aus einer Menge von Merkmalen eine geeignete Teilmenge zu selektieren Man kann den lokal günstigsten Schritt berechnen, d.h. jeweils die beiden ähnlichsten Klassen zusammenfassen, eine Folge der lokal optimalen Schritte muß aber nicht zum globalen Optimum führen. Die Vorausschau um mehr als einen Schritt auszuweiten, scheitert am kombinatorischen Aufwand (In Kapitel 4.5 wird diese Frage genauer behandelt. Abbildung 4.5 zeigt das Vorgehen bei der Merkmalsselektion, Abbildung 4.8 eine plausible Klasseneinteilung). Im folgenden beschreibe ich, welche Methoden man anwenden kann, um effizient eine Klasseneinteilung zu bestimmen, insbesondere das sogenannte *Complete-Linkage-Verfahren*, das ich in meiner Arbeit einsetze.

Wie oben bereits ausgeführt beginnt man damit, jedes Netz zuerst in eine eigene Gruppe einzuordnen und dann sukzessive die ähnlichsten zusammenzufassen. Die Ähnlichkeit $Sim(G_1, G_2)$ zweier Gruppen G_1 und G_2 definiert man wiederum über die stochastische Abhängigkeit ihrer Elemente

$$Sim(G_1, G_2) := \min_{i,j} I(y_i, y_j) \quad \text{mit } y_i \in G_1; y_j \in G_2 \tag{3.13}$$

und bestimmt dann, wie ausgeführt, das Maximum aller solcher Paare. In jedem Schritt kann man nun prinzipiell ein Heterogenitätskriterium über alle Gruppen ausrechnen, indem man zuerst zu je zwei Gruppen die Elemente minimaler Ähnlichkeit bestimmt, über alle Paare summiert und noch entsprechend normiert. Das oben definierte Kriterium (3.12) bekommt dann folgende Gestalt

$$Het_k := \frac{L-1}{L^2} \sum_{i=1}^{L} SIM(G_i, G_i) - \frac{2}{L^2} \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} SIM(G_i, G_j). \tag{3.14}$$

Aus jeder Gruppe muß man dann noch einen Vertreter bestimmen, beispielsweise das Netz mit der höchsten Evidenz oder eine Art 'mittleres' Element. Die Auswahl eines Vertreters bringt allerdings ein weiteres Problem mit sich. Solange sich die Netze relativ ähnlich sind, ist die Selektion eines Vertreters unbedenklich. Problematisch wird es nur, wenn die Homogenität der Gruppen stark abnimmt, d.h. mit zunehmender Verschmelzung, kommen völlig verschiedene, evtl. sogar stochastisch unabhängige Netze in einer Gruppe zusammen. Das macht es sinnvoll, als weiteres Kriterium die Homogenität der Gruppen zu betrachten. Diese kann man z.B. durch die beiden Netze definieren, die innerhalb der Gruppe minimale stochastische Abhängigkeit haben. Sei $K = |G_1|$ die Anzahl der Elemente in der Gruppe, dann definiert man die Homogenität als

$$Hom(G_1) := \min_{i,j} I(y_i, y_j) \quad \text{mit } y_i, y_j \in G_1. \tag{3.15}$$

Die Homogenität im Schritt k definiert man als die kleinste Homogenität einer der Gruppen

$$Hom_k \quad := \quad \min_i Hom(G_i) \ , \tag{3.16}$$

wobei i alle Gruppen durchläuft, die im aktuellen Schritt der Zusammenlegung existieren, d.h. all jene, aus denen ein Vertreter zur Bildung des Komitees ausgewählt werden soll. Das hier skizzierte Verfahren der Clusteranalyse bezeichnet man auch als Complete-Linkage-Verfahren, das besonders homogene Klassen erzeugt (Kaufmann & Pape, 1996). Prinzipiell könnte man die Homogenitäts- und Heterogenitätskriterien statt mit den minimalen und maximalen Ähnlichkeiten auch mit dem Durchschnitt berechnen (Average-Linkage-Verfahren). Es gibt keine objektiven Kriterien, welches der Verfahren bessere Lösungen liefert, vielmehr ist die Brauchbarkeit für das Untersuchungsziel entscheidend (Kaufmann & Pape, 1996). Sinnvollerweise wählt man also dazu die strengere Variante, da es das Ziel des Verfahrens ist, möglichst unabhängige Netze zu finden.

Letzendlich muß man eine Klasseneinteilung auswählen. Für diesen wesentlichen Aspekt der Clusterung, die Bestimmung der Klassenzahl, gibt es noch keine 'optimale' Lösung. (Kaufmann & Pape, 1996). Bei dem vorgeschlagenen agglomerativen Verfahren wählt man vernünftigerweise die Klassenzahl aus, die das Kriterium (3.14) maximiert (Ragg, 2000). Als Kontrolle sollte man dazu auch das Homogenitätskriterium beachten und sicherstellen, daß man sich bei der Auswahl der Klassenzahl in einem Bereich befindet, in dem die Homogenität noch groß ist. Dies ist deswegen sinnvoll, weil mit abnehmender Klassenzahl die Homogenität stark sinkt. Abbildung 3.1 zeigt die Auswahl an einem Beispiel. Ein starkes Absinken der Homogenität deutet darauf hin, daß zwei sehr heterogene Klassen verschmolzen wurden (Kaufmann & Pape, 1996). Damit sinkt dann auch der Wert des Heterogenitätskriteriums.

Für die letztendliche Güte des Komitees ist die Auswahl der Klassen entscheidend. Soll man aus einer Menge von trainierten Netzen das beste Komitee zusammenstellen, dann ist es sinnvoll, das Kriterium (3.14) zu optimieren. Es ist aber für die Suche nach guten Lösungen, die in Kapitel 4 erläutert wird, weniger wichtig, wieviele Klassen in jedem Suchschritt genau ausgewählt werden. Neue Suchpunkte werden um die Vertreter der Klassen mit größerer Wahrscheinlichkeit gesetzt (Ragg, 2000). Es ist also kein absolutes Kriterium, das alle anderen Netze sofort aus der Population entfernt. Diese besitzen vielmehr einen Selektionsnachteil. Dadurch wird es möglich, nach und nach auch bei verrauschter Qualitätsfunktion ein Optimum der beiden Kriterien - Evidenz einerseits, stochastische Unabhängigkeit andererseits - zu finden (vgl. Abbildung 4.2).

Neben agglomerativen Verfahren gibt es noch weitere Verfahren, die basierend auf einer Ähnlichkeitsmatrix eine Klasseneinteilung liefern. Insbesondere die polythetischen Verfahren wären hier zu nennen, die den umgekehrten Weg nehmen. Ausgehend von einer Klasse

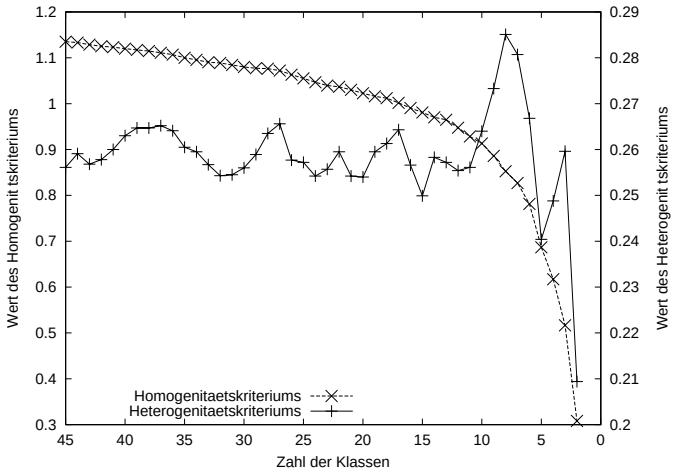


Abbildung 3.1: Die Abbildung zeigt die Bestimmung der Klassenzahl für eine Menge von neuronalen Netzen, die mit den Daten aus Abbildung 1.1 trainiert wurden. Das Heterogenitätskriterium ist eine Kurve mit mehreren lokalen Maxima und einem relativ deutlichen Maximum, hier bei 8 Klassen. Das Homogenitätskriterium ist lange Zeit relativ konstant und nimmt dann mit zunehmend sinkender Klassenzahl stark ab. Eine starke Abnahme in einem Schritt deutet darauf hin, daß sehr heterogene Gruppen verschmolzen wurden.

wird kontinuierlich eine Klasse geteilt. Das Problem hierbei ist, daß der Aufwand wesentlich größer ist. In jedem Schritt muß man für alle Klassen genau den Auftrennpunkt innerhalb der Klassen finden, der lokal optimal ist (Kaufmann & Pape, 1996). Erwähnenswert ist auch das sogenannte *Deterministic Annealing*, das bei der Segmentierung von Bildern oder Sprache gute Ergebnisse gebracht hat und auch auf einem soliden theoretischen Fundament beruht (Pereira *et al.*, 1993, Buhmann, 1997, Hofmann & Buhmann, 1997). Das Verfahren setzt allerdings voraus, daß die Klassenzahl bekannt ist. Diese soll jedoch als Ergebnis der Clusterung ermittelt werden. Indem man das Verfahren über alle Klassenzahlen iteriert, könnte man ein geeignetes Kriterium durch eine gewichtete Summe der Kriterien der einzelnen Ebenen definieren und optimieren (Puzicha *et al.*, 2000). Da die Population sich während der Evolution beständig verändert, macht es jedoch keinen Sinn, unverhältnismäßig viel Rechenaufwand darauf zu verwenden, eine eventuell geringfügig bessere Klasseneinteilung zu erhalten. Das würde den nachfolgenden Selektionsprozeß nicht wesentlich verändern, aber eine wesentlich kleinere Population erforderlich machen, um den Beschränkungen der Rechenzeit Genüge zu leisten.

Eine ähnliche Vorgehensweise schlägt auch Bishop für die Methode des 'Variational Learning' vor: Statt mit viel Rechenaufwand die lokal beste Approximation zu berechnen, verzichtet man auf Genauigkeit und führt in der gewonnenen Rechenzeit mehr Iterationen aus. Siehe dazu (Bishop, 1998), ebenso dort auch der Vergleich mit dem GEM-Algorithums.

Nach der Herleitung des analytischen Optimierungskriteriums wird zum Abschluß dieses Kapitels die Frage behandelt, wie man die Ähnlichkeit von Funktionen messen kann.

3.6 Ähnlichkeit von Funktionen

Zur Berechnung der Ähnlichkeit benötigt man eine Mustermenge. Beim Einsatz von Bootstrapping-Verfahren bietet es sich an, dafür einfach die Trainingsmenge oder eine Stichprobe davon zu nehmen. Jedes Netz wurde nur auf einer Teilmenge der Trainingsmenge trainiert, d.h. auf dem ganzen Datensatz sollten die Unterschiede der Ausgabefunktionen noch deutlicher hervortreten.

Als Ähnlichkeitsmaß kommen zwei Kriterien aus der Statistik in Betracht: eine Schätzung der linearen Abhängigkeit in Form des *Korrelationskoeffizienten* bzw. der nicht-linearen Abhängigkeit auf Basis des *wechselseitigen Informationsgehaltes* (engl.: mutual information).

3.6.1 Korrelationkoeffizienten

Die Korrelation zweier Zufallsvariablen mißt deren lineare Abhängigkeit, oder, mit anderen Worten, der Korrelationskoeffizient ist ein Maß für die lineare Vorhersagbarkeit einer Zufallsvariablen Y durch X. Dieser Koeffizient $\rho(X, Y)$ definiert sich auf Basis Kovarianz $C(X, Y)$ und dem Produkt der Varianzen von X und Y:

$$\rho(X, Y) = \frac{C(X, Y)}{\sqrt{V(X)V(Y)}}. \tag{3.17}$$

Der Korrelationkoeffizient ist die einfachste statistische Kenngröße für Paare von Zufallsvariablen. Sind zwei Funktionen bzw. Merkmale stark korreliert, dann wird man bei der Modellbildung auf eine bzw. eines der beiden verzichten können. Um nicht-lineare Abhängigkeiten zu erkennen, benötigt man eine Aussage darüber, inwieweit zwei Zufallsvariablen X und Y stochastisch abhängig sind, d.h. wieviel Information X über Y enthält (Cover & Thomas, 1991, Henze, 1997).

3.6.2 Mutual Information

Zwei Zufallsvariablen X, Y heißen stochastisch unabhängig, wenn für ihre gemeinsame Wahrscheinlichkeitsdichte gilt:

$$p(X, Y) = p(X)p(Y).$$

Das heißt, die gemeinsame Dichte entspricht genau dem Produkt der einzelnen Dichten (Henze, 1997). In anderer Form:

$$p(X, Y) - p(X)p(Y) \stackrel{!}{=} 0. \tag{3.18}$$

Um ein Maß für die stochastische Abhängigkeit zu erhalten, muß man den Wert der linken Seite von 3.18 schätzen. Basierend auf der Entropie einer Zufallsvariablen kann man Mutual Information als ein Maß für den Informationsgehalt einer Zufallsvariablen X über eine Zufallsvariable Y betrachten (Shannon, 1948, Cover & Thomas, 1991). Dadurch zieht man nicht-lineare Zusammenhänge in Betracht.

Mathematische Definition von Mutual Information

Die Entropie $H(X)$ definiert ein Maß für die Information einer gegebenen Zufallsvariablen X mit Wahrscheinlichkeitsfunktion p:

$$H(X) = -\int p(x) \log p(x)\, dx.$$

Für ein Paar X, Y von Zufallsvariablen mit gemeinsamer Verteilung $p(x, y)$ definiert sich die Entropie in ähnlicher Weise:

$$H(X, Y) = -\int\int p(x, y) \cdot \log p(x, y)\, dxdy.$$

Basierend darauf kann man die bedingte Entropie bei gegebener bedingter Verteilung $p(y|x)$ definieren:

$$H(Y|X) = -\int\int p(x, y) \cdot \log p(y|x)\, dxdy.$$

Im folgenden wird der Zugewinn an Information über eine Zufallsvariable Y durch Kenntnis von X, d.h. der Wert von $H(Y) - H(Y|X)$ betrachtet:

$$\begin{aligned} H(Y) - H(Y|X) &= -\int p(y) \log p(y)\, dy - \left(-\int\int p(x, y) \cdot \log p(y|x) dxdy\right) \\ &= -\int\int p(x, y) \cdot \log p(y)\, dxdy + \int\int p(x, y) \cdot \log p(y|x)\, dxdy \\ &= \int\int p(x, y) \cdot \log \frac{p(y|x)}{p(y)}\, dxdy \\ &= \int\int p(x, y) \cdot \log \frac{p(x, y)}{p(x)p(y)}\, dxdy. \end{aligned} \tag{3.19}$$

Der Ausruck 3.19 ist die Kullback-Leibler Distanz zwischen der gemeinsamen Verteilung $p(x, y)$ und dem Produkt $p(x)p(y)$ und wird als Mutual Information bezeichnet (Cover & Thomas, 1991). Mutual Information mißt also den Grad der stochastischen Abhängigkeit von X und Y. Damit ist die interessante Größe mathematisch definiert. Weiterhin muß sie für einen gegebenen Datensatz noch berechnet werden.

Es sei noch angemerkt, daß Mutual Information, wie andere Informationsmaße auch, monoton wachsend ist in der Zahl der Dimensionen. Das heißt, jedes Merkmal, das man zur Eingabe hinzunimmt, vergrößert den Informationsgehalt bezüglich der Ausgabe. Aufgrund des Fluchs der Dimensionen ist es aber wichtig, die Größe des Eingaberaumes zu beschränken.

Approximation auf Basis der Daten

Die drei Verteilungsfunktionen $p(x,y), p(x)$ und $p(y)$ müssen anhand der Daten geschätzt werden. Wenn keine Kenntnis über die Verteilungsfunktionen vorhanden ist und die Datenmenge ausreichend ist, dann empfiehlt es sich, eine nicht-parametrische Dichteschätzung zu verwenden (Silverman, 1986). Ich verwende hier einen multivariaten Epanechnikov-Kernschätzer basierend auf der euklidischen Metrik.

Sei $\mathbf{z}$ ein d-dimensionaler Vektor, dann definiert man die Kernfunktion

$$K(\mathbf{z}) = \begin{cases} (3/4)^d(1-(z_1^2+z_2^2+\ldots+z_d^2)) & \text{falls } \|\mathbf{z}\|_2 < 1 \\ 0 & \text{sonst.} \end{cases} \tag{3.20}$$

Die Normierungskonstante $(3/4)^d$, die sicherstellt daß $\int K(\mathbf{z})d\mathbf{z} = 1$ gilt, muß für jede Kernfunktion hergeleitet werden. Die ausführliche Rechnung hierzu findet sich im Anhang.

Der Vorteil der euklidischen Metrik besteht vor allem darin, daß die Daten sowohl binäre als auch reellwertige Komponenten enthalten können, ohne daß die Approximation der Verteilung deswegen ungenau wird. Bonnlander und Weigend verwenden im Gegensatz dazu in (Bonnlander & Weigend, 1994) die Maximumsmetrik, um paarweise den Informationsgehalt von Vektoren zu vergleichen. Hierzu ist zuerst anzumerken, daß das Informationsmaß aufgrund der Monotonie nicht direkt verwendbar ist, um einen möglichst optimalen Eingabevektor zu selektieren. In Kapitel 5 wird diese Aussage auch anhand eines Beispiels belegt. Weiterhin entsteht durch Verwendung der Maximumsmetrik für viele reale Anwendungen ein Problem, wenn die Eingabevektoren auch binäre Variablen enthalten. Ändert sich in einem Vektor nur eine binäre Komponente, dann ist das Muster bereits aus der Nachbarschaft aufgrund der speziellen Definition der Metrik hinausgefallen. Dies läßt sich auch nicht durch die *Fensterbreite* h kompensieren, die in Gleichung (3.21) eingeführt wird. Gilt $h > 1$, dann haben die binären Komponenten keinen Einfluß darauf, welches Muster in der Umgebung eines anderen liegt. Andernfalls ist der Einfluß so stark, daß nur solche Muster benachbart sind und zum Informationsgehalt beitragen, für die alle Bits den gleichen Wert haben, völlig unabhängig von den reellwertigen Komponenten. Damit beruht die Schätzung immer auf sehr wenigen Daten und wird im Extremfall unbrauchbar. Abbildung 3.2 illustriert die Problematik.

Für rein binäre Vektoren empfiehlt es sich entsprechend die Hamming-Distanz für den Kernschätzer zu verwenden. Für fast alle in der Praxis auftretenden Probleme ist der hier verwendete Kernschätzer sehr gut geeignet.

Der Wert der Wahrscheinlichkeitsdichte an einem gegebenen Punkt $\mathbf{z}(k)$ wird approximiert durch

$$p(\mathbf{z}(k)) = \frac{1}{N \cdot h^d} \sum_{j=1}^{N} K\Big(\frac{1}{h}(\mathbf{z}(k) - \mathbf{z}(j))\Big). \tag{3.21}$$

wobei N die Zahl der verwendeten Muster und h die Fensterbreite der Kernfunktion ist. Die Schätzung der drei Verteilungen $p(x,y), p(x)$ und $p(y)$ ergibt sich dann jeweils durch Summenbildung über alle Muster.

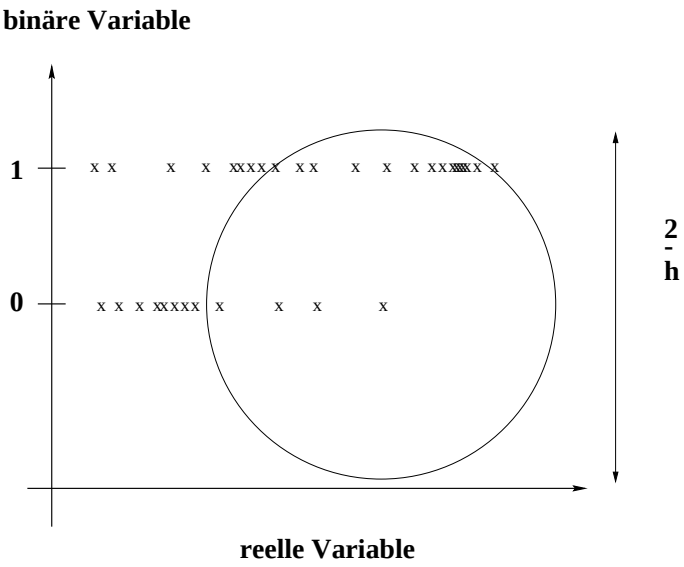


Abbildung 3.2: Die Abbildung zeigt die Nachbarschaftsbeziehungen für ein Beispiel mit einer reellen und einer binären Variablen. Der Kreis deutet an, welche Punkte bei der euklidischen Metrik in der Nachbarschaft des Mittelpunktes liegen. Je enger die Punkte an dem Mittelpunkt liegen, desto größer wird der Wert des Kernschätzers. Im Falle der Maximumsmetrik und $h < 1$ kann einer der oberen Punkte nie in der Nachbarschaft von einem der unteren Punkte liegen, unabhängig vom Wert der reellen Variablen. Im anderen Fall, $h > 1$, hat die binäre Komponente überhaupt keinen Einfluß auf die Nachbarschaftsbeziehungen, da die Norm des Arguments des Kernschätzers in dieser Richtung immer kleiner als 1 ist.

Die Wahl der Fensterbreite h, die eigentlich einen Glättungsparameter darstellt, ist ein wesentlicher Bestandteil der Schätzung. Der Parameter wird häufig subjektiv nach Einschätzung des Nutzers oder experimentell bestimmt (Bonnlander & Weigend, 1994, Ragg & Gutjahr, 1997a), was einen nicht unerheblichen Aufwand darstellt. Es gibt allerdings keine allgemein anerkannte Methode, diesen Glättungsparameter einzustellen. In jedem Fall ist ein Automatismus von wesentlicher Bedeutung, wenn die Dichteschätzung in einer größeren Optimierungsprozedur, wie hier vorgeschlagen, eingesetzt wird. In (Silverman, 1986) werden die Vor- und Nachteile verschiedener Methoden skizziert.

Verwendet man eine radialsymmetrische Kernfunktion und hat die zu schätzende Dichtefunktion stetige und beschränkte zweite Ableitungen, dann läßt sich für die Fensterbreite h eine optimale Größe ableiten (Silverman, 1986) zu:

$$h_{opt} = A(K)\frac{1}{N^{1/d+4}} \tag{3.22}$$

wobei N die Anzahl der Datenpunkte und d die Dimension des Vektors ist. Die Konstante $A(K)$ hängt vom verwendeten Kern ab und hat für den oben angegebenen Kern den Wert

$$A(K) = \left(\frac{8}{c_d}(d+4)(2\sqrt{\pi})^d\right)^{-(d+4)}.$$

Dabei ist c_d das Volumen der d-dimensionalen Einheitskugel. Für die Verwendung eines Kernschätzers ist aufgrund der Metrik zu beachten, daß die Daten in ähnlicher Weise skaliert sein sollten. Durch *Standardisierung* oder *Whitening* läßt sich dieser Anforderung einfach Genüge leisten (Fukunaga, 1990, Silverman, 1986). Für die meisten Algorithmen des maschinellen Lernens sind diese Vortransformationen empfehlenswert, insbesondere auch beim Einsatz von Regularisierungstermen (Bishop, 1995).

Es bleibt abschließend zu bemerken, daß die Berechnung des Mutual Information für einen gegebenen Datensatz einen Aufwand von $O(dN^2)$ hat. Dies ergibt sich aus dem Aufwand $O(d)$ zur Berechnung einer Kernfunktion, dem Aufwand $O(dN)$ zur Berechnung der Wahrscheinlichkeitsdichte für einen festen Punkt und der Wiederholung dieser Berechnung für alle N Punkte. Hat man eine sehr große Datenmenge ($\gg 1000$ Datenpunkte), dann genügt es völlig, eine Stichprobe auszuwählen. Die Approximation wird deswegen nicht viel schlechter, aber der Aufwand wird erheblich reduziert.

3.7 Zusammenfassung

Der wesentliche Gesichtspunkt, den es bei der Bildung von Komitees zu beachten gilt, ist die Unabhängigkeit der einzelnen Netze. Unter bestimmten Annahmen läßt sich zeigen, daß der Fehler eines Komitees drastisch kleiner werden kann als der durchschnittliche Fehler einer Menge von Netzen. Andererseits läßt sich allgemein für alle (konvexen) Fehlerfunktionen mittels der Ungleichung von Jensen zeigen, daß der Fehler eines beliebigen Komitees nie größer wird als der durchschnittliche Fehler der Netze. Um Unabhängigkeit der einzelnen Modelle zu erreichen, kommen verschiedene Möglichkeiten in Betracht. Die wichtigsten Methoden sind, einerseits die Trainingsdaten mehrfach zu sampeln (Bootstrapping, Teilmengenbildung) und andererseits, wenn möglich, verschiedene Merkmale zu verwenden. Weiterhin ist es auch sinnvoll, die Netzkomplexität zu varieren, so daß sowohl 'einfache', d.h. nahezu lineare Lösungen als auch entsprechend nicht-lineare Lösungen im Komitee vorhanden sein können.

Die ganz entscheidene Frage lautete hier, wie man die Komiteemitglieder auswählen kann. Ausgehend von der Bias-Varianz Dekomposition 3.3 des Komiteefehlers habe ich hierzu ein Kriterium entwickelt, das erlaubt, zwei Komitees miteinander zu vergleichen. Basierend auf diesem Kriterium wurde vorgeschlagen, ein möglichst gutes Komitee durch Clusterung der Modelle zu bestimmen. Dabei wird die optimale Klassenzahl mittels des Kriteriums (3.12) bestimmt. An Gleichung (3.3) kann man auch erkennen, daß es keinen Sinn macht, die Unabhängigkeit in die Fehlerfunktion zu integrieren, wenn man gleichzeitig auf weitere Regularisierung, z.B. durch Weight-Decay, verzichtet. Der durchschnittliche Fehler wird sonst ansteigen und damit auch der Komiteefehler.

Bisher gehe ich davon aus, daß eine Menge von trainierten Netzen vorliegt, um dann ein Komitee zu bestimmen. Es drängt sich die Frage auf, ob hierzu nicht ein zielgerichtertes Vorgehen möglich ist. Im nächsten Kapitel werde ich, basierend auf den bisher vorgestellten und entwickelten Verfahren, einen Algorithmus entwickeln, um gezielt nach Netzen zu suchen. Dabei unterscheide ich zwei Stufen: erstens die Suche nach dem Modell mit maximaler Evidenz und zweitens die kombinierte Suche nach stochastisch maximal unabhängigen Modellen mit hoher Evidenz, um basierend auf dem Kriterium (3.12) ein Komitee zu finden.

Kapitel 4

Evolutionäre Suchverfahren

Mit den bisher eingeführten Methoden ist man in der Lage, die freien Parameter eines neuronalen Netzes möglichst optimal einzustellen und die Gewichtung eines Regularisierungstermes zu bestimmen. Darüberhinaus kann man für jedes so bestimmte Modell seine Qualität, wenn auch nur verrauscht, angeben. Hat man für ein bestimmtes Anwendungsproblem viele Modelle auf diese Art entwickelt, dann stellt sich die Frage, welches Modell man nun eigentlich in der Praxis einsetzen möchte. Eine übliche Vorgehensweise ist, das Modell mit dem höchsten Qualitätsmaß auszuwählen (Bishop, 1995) oder über die besten Modelle geeignet zu mitteln, indem man ein Komitee bildet, wie das im vorigen Kapitel ausgeführt wurde. Die Methoden der Parameteroptimierung und auch der Komiteebildung setzen bisher nur voraus, daß man einige Modelle trainiert hat, aus denen man das geeignetste oder auch mehrere nach einer vorgegebenen Vorschrift auswählt. Die Frage, ob man Modelle oder Komitees gezielter entwickeln kann, z.B. durch Einsatz eines Suchverfahrens, wurde bisher noch ausgeklammert. Dieser Frage möchte ich mich nun zuwenden. Hierzu stelle ich zuerst einige grundlegende Überlegungen zur Entwicklung neuronaler Modelle vor und werde argumentieren, daß der Einsatz heuristischer Suchverfahren, wie z.B. evolutionärer Algorithmen, sinnvoll ist. Nach der Darstellung des Verfahrens ENZO (Braun & Ragg, 1996a, Braun, 1997) erläutere ich die Punkte, die für ein integriertes Optimierungskonzept zu beachten sind. Darauf aufbauend stelle ich das Konzept zur Evolution unabhängiger Netze vor.

4.1 Wie findet man 'gute' Modelle?

Die oben skizzierte Vorgehensweise bei der Auswahl eines Modells setzt voraus, daß man vor dem eigentlichen Training einige weitere Stufen durchläuft: Zuerst wird man eine geeignete Kodierung für das Problem festlegen müssen. Auf dieser Stufe muß man oft aus einer unbegrenzten Menge von potentiellen Merkmalen eine Auswahl treffen, welche wiederum die mögliche Güte des Modells erheblich beeinflußt. Betrachtet man neuronale Netze als semiparametrische Modelle in einem wahrscheinlichkeitstheoretischen Kontext (Bishop, 1995), dann entspricht die Wahl der Topologie der Festlegung der Zahl und Art der parametrischen Funktionen, deren Parameter dann mit Hilfe der Daten bestimmt werden.

Sind für ein gegebenes Problem eine Kodierung und eine Topologie gewählt, dann werden die Trainingsdaten ausgewählt und mittels dieser Daten die Parameter des neuronalen Netzes bestimmt. Durch ein Bayes'sches Lernverfahren werden nicht nur die Gewichte berechnet, sondern auch die Zahl der effektiven Parameter festgelegt. Ist die Topologie zu groß gewählt, dann wird die Anzahl der Gewichte die Zahl der effektiven Parameter, wie sie in Gleichung (2.23) definiert wurden, weit übersteigen. Dies hat die Konsequenz, daß die Approximationen, die im Bayes'schen Ansatz gemacht werden, ungenauer sind. Das heißt zum Beispiel, daß viele Eigenwerte der Hessematrix nahe bei Null liegen und damit die Invertierung der Matrix numerisch instabil wird. Ebenso wird die Berechnung der Modellevidenz ungenauer, da dabei das Produkt der Eigenwerte auftritt (Gleichung (2.36)). Offensichtlich werden die Ergebnisse des Verfahrens umso besser, je 'angemessener' die Topologie des Netzes der Struktur des Problems ist. Experimentelle Ergebnisse bestätigen diese Aussage (Thodberg, 1993, Ragg & Gutjahr, 1998). Daraus folgt, daß automatische Regularisierung allein noch nicht ausreicht, um geeignete Modelle zu finden. Zu diesem Ergebnis kommt auch Gutjahr in seiner Dissertation (Gutjahr, 1999). In Abbildung 1.4 war das für den Merkmalsvektor deutlich ersichtlich. In ähnlicher Weise gilt das auch für die Zahl der versteckten Neuronen. Das Problem der Topologiefindung ist insofern einfacher, als eben nur die Größe der versteckten Schichten Bedeutung hat, nicht aber wie bei der Merkmalsselektion die richtige Kombination der Komponenten. Das bedeutet, daß man für diese Optimierung mit weniger mächtigen Methoden auskommt.

Neben der Parameteroptimierung und der Komiteezusammensetzung sind also in grundlegender Weise die folgenden drei Aspekte der Modellentwicklung für ein integriertes Optimierungskonzept zu berücksichtigen.

Problemkodierung: Bisher existiert noch keine allgemeine Theorie, mittels derer man gute Merkmale findet (Schürmann, 1996). Die Entwicklung geeigneter Merkmale oder die Vorverarbeitung der Daten, z.B. durch Hauptachsentransformation, entscheidet der Entwickler aufgrund seines Problemwissens. Um in der realen Anwendung den Konsequenzen des *curse of dimensionality* Rechnung zu tragen, selektiert man aus einer Menge von entwickelten Merkmalen eine geeignete Teilmenge anhand verschiedener Kriterien, wie z.B. Mutual Information, Fisher's lineare Diskriminate oder Kovarianzmatrizen. Durch Kombination mit Suchstrategien schränkt man den kombinatorischen Aufwand einer vollständigen Suche auf ein beherrschbares Maß ein (Fukunaga, 1990, Bishop, 1995, Schürmann, 1996). Der Nachteil dieser statistischen Kriterien ist, daß das Verhältnis der Merkmale zur Anzahl der Muster nicht direkt berücksichtigt wird. Da die genannten Kriterien bezüglich der Zahl der Dimensionen monoton wachsen, verbessert jedes zusätzliche Merkmal den Wert des Kriteriums (Bishop, 1995). An dieser Stelle verwendet man oft Kreuzvalidierung, um die Güte linearer Modelle mit verschiedenen Kodierungen zu vergleichen und so eine Kodierung für das nicht-lineare Modell zu gewinnen. In (Ragg & Gutjahr, 1997a) wurde vorgeschlagen, die Selektion auf Basis statistischer Kriterien mit dem Training der Netze in einem Suchprozeß zu verzahnen.

Topologie: Um eine geeignete Topologie zu finden, verfährt man üblicherweise nach Versuch und Irrtum: Es wird sukzessive für mehrere Topologien eine Anzahl an Netzen trainiert,

und die im Mittel geeignetste Topologie wird ausgewählt. Der Aufwand vervielfacht sich entsprechend. Durch die vorgegebene Anzahl an Trainingsmustern ist die maximale Größe der Topologie beschränkt. In (White, 1989) werden sinnvolle Schranken für die Zahl der Parameter in Abhängigkeit der Größe der Trainingsmenge hergeleitet für den Fall, daß keine Regularisierung angewendet wird. Dadurch läßt sich also eine maximale Topologie festlegen, die den Suchraum begrenzt. Die Optimierung der Topologie mit evolutionären Strategien ist ein häufig und seit längerer Zeit verfolgter Ansatz (Schiffmann *et al.*, 1992, Braun & Weisbrod, 1993, Braun & Ragg, 1996a, Alander, 1996, Yao, 1999), der bisher allerdings immer auf Kreuzvalidierungstechniken basierte.

Datenabhängigkeit: In Kapitel 2.2 wurde gezeigt, daß die Abhängigkeit von den gewählten Trainigsdaten ein grundsätzliches Problem ist, dem man am besten dadurch begegnet, daß man über mehrere Datensätze integriert, die aus den Trainigsdaten *gezogen* werden. Das Ziehen kann dabei mit oder ohne Zurücklegen erfolgen. Diese Problematik legt es nahe, nach dem Training nicht das 'beste' Modell auszuwählen, sondern ein Komitee zu bilden. Die Aufgabe des Optimierungsprozesses ist es also, mehrere lokale Optima zu finden. Auch hier stellt sich die Frage, ob das Sampling ein einmaliger Vorgang zu Beginn der Suche ist, oder ob man durch Variation der Datensätze während der Suche noch signifikante Verbesserungen erreichen kann.

Zusammenfassend kann man sagen, daß die genannten Aspekte einen Raum von möglichen Einstellungen festlegen. Der Entwickler hat die Aufgabe, geeignete Einstellungen zu bestimmen. Bei der gewöhnlichen Vorgehensweise zur Entwicklung neuronaler Netze sucht man nach Festlegung der Designentscheidungen unter einer Menge von Suchpunkten den günstigsten aus, wobei die Suchpunkte durch eine Zufallsauswahl mit anschließender Optimierung der Fehlerfunktion gewonnen wurden. Der Schluß liegt nahe, daß man durch Einsatz eines Suchverfahrens an dieser Stelle bei gleichem Aufwand bessere Lösungen finden kann. Der Einsatz eines evolutionären Suchverfahrens ist jedenfalls dann sinnvoll, wenn die beste Lösung mit klassischen Verfahren nicht direkt gefunden werden kann. Dies ist insbesondere dann der Fall, wenn mehrere Optima vorhanden sind (Multimodalität) und/oder nur ein verrauschtes Qualitätsmaß für die Lösung angegeben werden kann. In der Literatur wird dieser Sachverhalt oft mit einer Gruppe von Bergsteigern im Nebel veranschaulicht (Rechenberg, 1994, Schwefel, 1995).

Diese Eigenschaft evolutionärer Suchverfahren, eine multimodale Suche auch bei verrauschter Qualitätsfunktion zu ermöglichen, läßt als Suchverfahren aber prinzipiell jedes Verfahren in Betracht kommen, das gerade diese Eigenschaften hat, wie z.B. auch tabu search (Glover, 1993) oder andere heuristische Verfahren (Goldberg, 1989, Reeves, 1993). Weiterhin sind es vor allem praktische Gesichtspunkte, auf denen eine Entscheidung basiert:

- Die wichtigste Eigenschaft evolutionärer Algorithmen ist die Verwendung einer Population von Suchpunkten, die es erlaubt - und das ist wichtig -, globale Information über die verschiedenen Suchpunkte in die lokale Ausgestaltung der weiteren Suche einzubeziehen. Wie bereits erwähnt, ist die Eigenschaft, auch bei verrauschter Qualitätsfunktion ein Optimum zu finden, für das hier gestellte Problem notwendig.

- Das Verfahren ist sehr modular und daher programmiertechnisch effizient umsetzbar. Dadurch gestaltet sich sowohl die Pflege als auch die Erweiterung der Software sehr einfach (Braun & Ragg, 1996a).

- Eine Parallelisierung des Algorithmus ist bei großem speed-up einfach und damit auch kostengünstig zu entwickeln (Ragg, 1996). Insbesondere während der Entwicklungsphase eines Verfahrens ist ein Zeitgewinn zur Vermeidung von Irrwegen von Bedeutung.

- Durch die Anlehnung an die natürliche Evolution gewinnt die Methodik einen zusätzlichen Reiz: Übertragung von Konzepten aus der Natur in die Technik und umgekehrt auch ein besseres Verständnis des natürlichen Verfahrens (Rechenberg, 1994). Softwaretechnische Simulationsumgebungen für evolutionäre Prozesse haben die Evolutionstheorie sehr bereichert, insbesondere die Forschung zur Entwicklung von kooperativem Verhalten sehr geprägt (Dawkins, 1994, Badcock, 1999).

Die Qualität eines neuronalen Modells kann durch die Evidenz oder den Fehler auf einer zusätzlichen Testmenge geschätzt werden (vgl. Kapitel 2). Beide Maße enthalten eine Rauschkomponente: Im ersten Fall werden idealisierende Annahmen gemacht und Integrale nur approximativ gelöst, im zweiten Fall wird der Fehler auf endlich vielen Datenpunkten gemessen. Weiterhin sind die Ausgaben meist durch ein unbekanntes Rauschen überlagert, bezüglich derer der Fehler gemessen wird. Die originalen Zielwerte des zugrundeliegenden Prozesses sind in aller Regel unbekannt. Es ist also gerechtfertigt, eine Abhängigkeit des Qualitätsmaßes von den Parametern wie in Abbildung 4.1 anzunehmen.

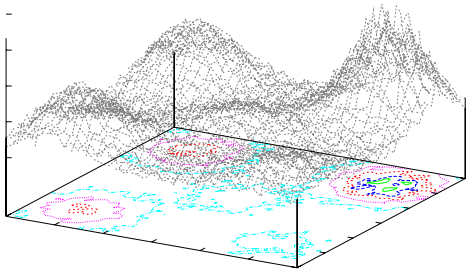


Abbildung 4.1: Abhängigkeit eines Qualitätsmaßes von zwei Parametern. Die Gütefunktion weist mehrere lokale Maxima auf, die aber von Rauschen überlagert sind. Die genaue Parameterkombination, die einem Maximum entspricht, kann nicht genau bestimmt werden. Ein evolutionäres Suchverfahren stellt sicher, daß man sich in der globalen Struktur zu den Bergspitzen hin bewegt.

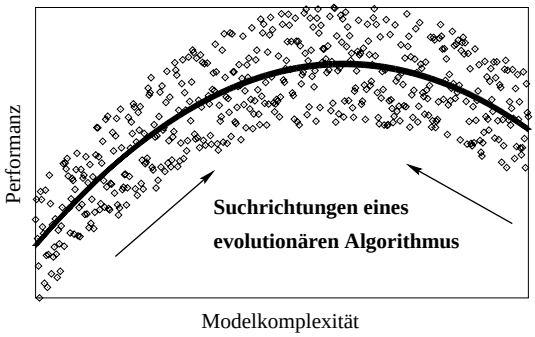


Abbildung 4.2: Die Abbildung zeigt eine hypothetische eindimensionale Projektion der Qualitätsfunktion und die Suchrichtungen, die ein evolutionärer Algorithmus verfolgen würde. Wenn man nur wenige Punkte zur Verfügung hat, dann ist es schwierig, das Optimum zu lokalisieren. Dadurch daß evolutionäre Strategien einem Gradientenpfad folgen, bewegen sie sich von Generation zu Generation weiter nach oben und pendeln dann um das Maximum.

Das bedeutet, daß die Auswahl eines Modells aufgrund seines Qualitätsmaßes einer Suche über den lokal optimalen Einstellungen entspricht. Da die Qualitätsfunktion Stetigkeitseigenschaften hat - kleine Variationen der Parameter werden auch die Qualität nur gering verändern - macht es Sinn, nach 'guten' Modellen gezielt zu suchen. Man bezeichnet das auch als das starke Kausalitätsprinzip - Ähnliches ballt sich zusammen (Rechenberg, 1994). Mit einer evolutionären Strategie kann man nun lokale Maxima der Qualitätsfunktion finden, ohne daß man einen Gradienten berechnen müßte. Im Falle diskreter Parameter (Merkmale, Topologie, Datenauswahl) ist dies auch gar nicht möglich. Die Abbildung 4.2 illustriert die Suchrichtungen der evolutionären Strategie für eine hypothetische eindimensionale Projektion der Qualitätsfunktion. Indem die Suchpunkte jeweils um die Eltern gestreut und Modelle mit höherer Qualität bevorzugt selektiert werden, tendieren die Lösungen mit zunehmender Generationenzahl immer weiter nach oben und gruppieren sich letztendlich um das Maximum.

Jedes heuristische Suchverfahren muß das Dilemma zwischen breiter Suche im gesamten Suchraum, *Exploration*, und verstärkter Suche im Umfeld vielversprechender Suchpunkte, *Exploitation*, lösen. Durch einen höheren Berechnungsaufwand kann man systematisches Suchen durch statistische Verfahren ersetzen und dadurch ungünstige Suchschritte vermeiden (vgl. auch (Rechenberg, 1994)). Das heißt, statt ungerichtet viele Suchschritte durchzuführen und zu vergleichen, bewertet man die möglichen Suchschritte durch statistische Verfahren. Ein Algorithmus ist also auch danach zu beurteilen, wie er dieses Dilemma auflöst.

4.2 Evolutionäre Suchverfahren

Ein neuronales Netz mit all seinen Parametern heißt in der Begriffswelt der evolutionären Algorithmen ein *Individuum*. Im Bergsteigerbeispiel repräsentiert das Individuum eine Position im Gebirge. Jedem Individuum wird über die Zielfunktion ein Fitneßwert zugeordnet. Eine Menge von μ Individuen bildet eine *Population*, die sich unter Verwendung von Operatoren im Laufe der Generationen verändert. In jeder Generation werden λ Nachkommen durch lokale Operatoren generiert: Für die Optimierung neuronaler Netze sind das vor allem die *Mutation* der Topologie und das *Training*, d.h. die lokale Adaption der Gewichte und Hyperparameter. Die *Selektion* bestimmt, welche Individuen in welchem Umfang Nachkommen in der nächsten Generation haben. Bezogen auf das Bergsteigen heißt das, von welchen Punkten aus mit wieviel 'Suchern' weitergesucht wird. Man unterscheidet im wesentlichen zwei Arten der Selektion: *Plus-Selektion* und *Komma-Selektion*. Bei der Plus-Selektion werden die μ besten aller $(\mu + \lambda)$ Individuen zu Eltern in der nächsten Generation. Bei der Komma-Selektion werden die Eltern jeder Generation verworfen und durch die μ besten der λ Nachkommen ersetzt. Sie wird als (μ, λ)-*Strategie* gekennzeichnet.

Die $(\mu+\lambda)$-*Strategie* erhält die Individuen mit guter Fitneß. Dies geht zu Lasten der Durchwanderung von Tälern, da von einem lokalen Minimum aus nicht mehr weitergesucht, sondern dort verharrt wird. Die (μ, λ)-Strategie ist dagegen in der Lage, wandernden Optima zu folgen. Für den Einsatz in dem integrierten Optimierungskonzept ist sie deswegen zu bevorzugen, da die $(\mu + \lambda)$-*Strategie* die Eltern nicht weiter verändert. Die Verzahnung des Bayes'schen Lernens mit dem Suchverfahren macht es aber erforderlich, daß die Eltern wieder trainiert werden. Die Erzeugung von Nachkommen erfolgt gerade bei der Verände-

rung der Fehlerfunktion durch Adaption des Gewichtungsfaktors des Regularisierungstermes. Dieses Vorgehen wird weiter unten noch genauer beschrieben. Weiterhin kommt die (μ, λ)-Strategie auch der Idee, Modelle zu clustern, entgegen. In das Bergsteigerbeispiel übersetzt, selektiert die Clusterung die unabhängigsten Suchrichtungen, in denen gemäß einer (μ, λ)-Strategie verstärkt Suchpunkte definiert werden. Von den so erhaltenen Suchpunkten schreitet der Prozeß in derselben Weise fort.

Eine ausführliche Darstellung von *Evolutionären Strategien* findet sich bei (Rechenberg, 1994, Schwefel, 1995, Bäck, 1996), deren Notation ich hier gefolgt bin. Für die Verwendung des evolutionären Suchverfahrens in dieser Arbeit ist anzumerken, daß der Suchraum zwar reellwertig ist - die Parameter des wahrscheinlichsten Gewichtsvektors zu bestimmen -, die Mutationsoperatoren aber nur einen Teil der Parametersuche abzudecken haben, nämlich die diskrete Suche nach einer günstigen Topologie und einem guten Eingabevektor. Die Einstellung der Gewichte läßt sich wesentlich effizienter durch ein Lernverfahren bewerkstelligen. Mutation und Lernen ergänzen sich also gegenseitig.

4.3 Optimieren durch Lernen und Evolution (ENZO)

Diese Arbeit baut auf einer mehrjährigen Erfahrung zur Optimierung von neuronalen Netzen durch evolutionäre Verfahren auf. In diesem Abschnitt möchte ich diesen Ansatz beschreiben, dann die wesentlichen Nachteile skizzieren, um im folgenden zu konkretisieren, wie eine Suche nach mehreren unabhängigen Lösungen bewerkstelligt werden kann.

Der Ansatz, neuronale Netze durch Lernen und Evolution zu optimieren, wird seit mehreren Jahren an diesem Institut verfolgt (Braun & Weisbrod, 1993, Braun & Ragg, 1996b, Ragg, 1996, Braun, 1997, Ragg *et al.*, 1997, Menzel, 1998). In einer Vielzahl von Diplomarbeiten wurden verschiedene Aspekte des Ansatzes eingehend untersucht: Mutationsoperatoren für Neuronen und Gewichte (Weisbrod, 1992, Zagorski, 1994, Schäfer, 1994, Schubert, 1995), Netze aus *Radiale Basisfunktionen* (Preut, 1995, Sprenger, 1996), Kombination mit *Reinforcement-Lernen* (Pütz, 1997, Hofmann, 1997), multimodale Evolution (Engelmann, 1996) und Selektion von Trainingsmustern (Schmiedle, 1997). Im Rahmen dieser Arbeiten ist das System ENZO (Evolutionärer Netzwerk Optimierer) entstanden (Braun & Ragg, 1996a), das sich durch sein modulares und flexibles Design auszeichnet, und dessen erste Version inzwischen zusammen mit dem Neurosimulator SNNS (Stuttgarter Neuronale Netze Simulator)(Zell, 1994) Verbreitung gefunden hat.

Man kann ENZO als eine Suche auf lokalen Minima charakterisieren. In der Lernphase (Feineinstellung) werden die Parameter so eingestellt, daß ein lokales Minimum erreicht wird. Im Evolutionsschritt werden diese Lösungen dann der Selektion und Mutation (Grobeinstellung) unterworfen. Mutationsoperatoren werden in ENZO nur auf Topologieebene definiert. Das heißt, es werden Neuronen oder Gewichte entfernt oder eingefügt. Abbildung 4.4 zeigt die wesentlichen Schritte des Verfahrens in einer algorithmischen Darstellung. Der Suchraum wird in ENZO durch eine Referenztopologie begrenzt. Diese dient auch dazu, allen Neuronen und Gewichten einen eindeutigen Namen und Platz zu geben. Ein Gewicht oder Neuron kann nur eingefügt werden, wenn dieses gegenüber der Referenztopologie fehlt. Da in der Regel von der Referenztopologie ausgegangen wird, kann man diese Vorgehensweise als Top-Down Entwurf bezeichnen.

4.3.1 Initialisierung

Zu Beginn der Evolution werden die Struktur und Parameter der Netze zufällig initialisiert. Das heißt, es werden beliebige Eingabestrukturen und versteckte Neuronen gewählt, begrenzt durch die Referenztopologie, und die vorhandenen Gewichte initialisiert. Dazu werden in ENZO standardmäßig Zufallszahlen aus dem Intervall $[-1, 1]$ gezogen. Durch die Initialisierung wird erreicht, daß die Suchpunkte über den ganzen Raum verteilt sind.

4.3.2 Mutationsoperatoren

Die Mutationsoperatoren haben eine ähnliche Aufgabe in der evolutionären Optimierung wie die Pruning-Schritte in den sogenannten Pruning-Verfahren (Reed, 1993, Zell, 1994, Bishop, 1995). Diese reduzieren die Komplexität der Modelle dadurch, daß weniger relevante Gewichte oder Neuronen entfernt werden, nachdem der Fehler in ein Minimum konvergiert ist. Danach wird in der Regel nachtrainiert. Die Verfahren unterscheiden sich darin, wie die Relevanz geschätzt wird, z.B. Betrag des Gewichtes, statistische Relevanz, Zunahme des Fehlers, etc. Die Zahl der Parameter sinkt während des Trainings, solange der Fehler unter einer bestimmten Grenze bleibt. In der Regel wird die Grenze mittels einer Validierungsmenge bestimmt und dadurch die Gewichtung von Komplexität zu Fehler vorgenommen. Pruning-Verfahren kann man als einelementige evolutionäre Algorithmen betrachten (Ragg *et al.*, 1997, Braun, 1997).

Mutationsoperatoren sind in ENZO problemspezifisch ausgestaltet. Im Gegensatz zur üblichen Vorgehensweise, auf das Attribut normalverteilte Zufallszahlen zu addieren (Rechenberg, 1994), stellt sich bei der diskreten Topologieoptimierung nur die Frage, ob ein Element entfernt werden soll oder nicht. In ähnlicher Weise wie bei einem Pruning-Verfahren sortiert man die Elemente, d.h. Gewichte oder Neuronen, nach ihrer Relevanz und entfernt dann ein oder mehrere wenig relevante Elemente mit einer gewissen Wahrscheinlichkeit.

Gleiches gilt für das Einfügen von Elementen. Die Art der Mutation bezeichnet man auch als *rangbasiert* (engl.: ranking-based) im Unterschied zur bewertungsbasierten Mutation, die die Wahrscheinlichkeit der Mutation direkt von der Relevanz abhängig macht. Das hat den Vorteil, daß man für alle Mutationsschritte ein einheitliches Schema verwenden kann und sich keine Gedanken, um die Größe der Bewertung machen muß. Abbildung 4.3 zeigt verschiedene Auswahlfunktionen von der Form x^y für die rangbasierte Selektion. Die Rangordnung wird dabei auf das Intervall $[0, 1]$ abgebildet. Um ein Element zu ermitteln wird eine Zufallszahl z aus einer Gleichverteilung zwischen 0 und 1 gezogen. Mit z^y ist der Abschnitt des Intervalls festgelegt, dessen zugehöriges Element selektiert wird. Ist $y > 1$, dann fallen mehr Zahlen in die vorderen Intervalle. Der Parameter y legt also in diesem Fall einen Selektionsvorteil zugunsten der Elemente am Anfang der Liste fest. Für Mutationsoperatoren wird man y groß wählen ($y \geq 4$), damit sinnvolle Suchschritte gemacht werden. Bei der Selektion der Eltern wird man den Selektionsdruck eher klein halten, um eine zu schnelle Konvergenz zu verhindern ($y \leq 2$).

Der Suchraum für die Topologie wird durch eine Referenz-Topologie begrenzt. Das heißt, bei einem Individuum können nur solche Elemente eingefügt werden, die ihm gegenüber der Referenztopologie fehlen.

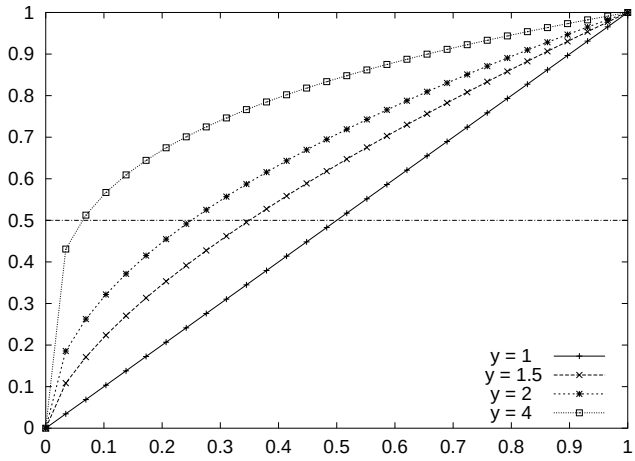


Abbildung 4.3: Die Abbildung zeigt verschiedene Auswahlfunktionen von der Form x^y für die rangbasierte Selektion, die in ENZO bei der Mutation und der Selektion verwendet wird.Der Bevorzugungsfaktor y steuert, um wieviel wahrscheinlicher Elemente am Anfang der Liste selektiert werden. Wählt man einen Punkt auf der x-Achse und geht dann senkrecht nach oben, bis man die Auswahlfunktionen schneidet, dann gibt der Wert auf der y-Achse an, wie wahrscheinlich es ist, daß ein Element links vom x-Wert selektiert wird.

In ENZO werden für die verschiedenen Elemente - Gewichte, versteckte Neuronen und Eingabeneuronen - eigene Mutationsoperatoren bereitgestellt, die frei kombiniert werden können. Tabelle 4.3.2 stellt die verschiedenen Sortierkriterien zusammen und bewertet den Zeitaufwand für die Mutation. Für die Herleitung der Aufwandsabschätzung sei auf die angegebenen Referenzen und auf (Ragg *et al.*, 1997, Braun, 1997) verwiesen. Für die Bewertung ist noch zu beachten, daß die Trainingsphase bereits einen Aufwand von $O(eNW)$ hat (Bishop, 1995, Ragg *et al.*, 1997), wobei e die Zahl der Epochen ist, N die Zahl der Muster und W die Zahl der Gewichte im Netz. Für das Bayes'sche Lernen gilt hier $O(eNW^2)$, da die Hessematrix berechnet und invertiert werden muß.

Methode	Kriterium	Aufwand	Typ	Referenz
MbP	$\lvert w_q \rvert$	$O(W)$	W	-
OBD	$\frac{H_{qq} w_q^2}{2}$	$O(NW)$	W	(LeCun *et al.*, 1990)
OBS	$\frac{w_q^2}{2H_q^{-1}q}$	$O(NW^2)$	W	(Hassibi & Stork, 1992)
random	-	$O(1)$	U	-
unit-MbP	$\sum_j \lvert w_{ji} \rvert + \sum_k \lvert w_{ik} \rvert$	$O(W\sqrt{W})$	U	-
unit-OBS	$-H^{-1}M(M^TH^{-1}M)^{-1}M^T\mathbf{w}$	$O(NW^2)$	U	(Stahlberger, 1996)

Tabelle 4.1: Die Tabelle stellt einige Pruning-Methoden, die verwendeten Auswahlkriterien und deren Aufwand zusammen. Der Aufwand bezieht sich nur auf die Auswahl eines Elementes. Um den Aufwand der gesamten Evolution anzugeben, muß man berücksichtigen, daß Mutationen von Neuronen wesentlich seltener durchgeführt werden als reine Gewichtsmutation. H bezeichnet wieder die Hessematrix, H_{qq} das q-te Diagonalelement derselben und $M = (e_{q1}, \ldots, e_{qn})$ eine Auswahlmatrix analog zu dem Auswahlvektor bei OBS, die angibt, welche Gewichte gelöscht werden sollen.

Für die auf dem Betrag der Gewichte basierenden Verfahren wird vorausgesetzt, daß die Eingabe korrekt skaliert ist, d.h. daß alle Eingabekomponenten gleichen Mittelwert und Varianz haben. Ansonsten werden durch die verschiedenartige Skalierung bestimmte Elemente gegenüber anderen benachteiligt, ohne daß das ihrer eigentlichen Relevanz entspricht (Bishop, 1995), vgl. auch (Silverman, 1986, Ramsay & Silverman, 1997).

Rekombinationsoperatoren sind in ENZO zwar implementiert (Braun & Ragg, 1996a), konnten aber bisher nicht mit Erfolg eingesetzt werden. Zufällige Rekombination aus mehreren Eltern führt dazu, daß die Netzfunktion oder auch die Hyperparameter so sehr verändert werden, daß die Rekombination einer Zufallsinitialisierung entspricht. Der Grund dafür liegt darin, daß Rekombinationsoperatoren mit mehreren Eltern voraussetzen, daß die Parameter unabhängig voneinander variiert werden können. In der Natur werden deshalb zusammengehörige Informationsblöcke mittels sogenannter Crossover-Points als Ganzes vererbt. Man geht davon aus, daß durch Rekombination sich 'gute' Gene, die sich in verschiedenen Individuen entwickelt haben, in einem Individuum zusammenfinden können (Dawkins, 1994). Für die Evolution neuronaler Netze käme hier vor allem die Rekombination von Merkmalen der Eingabeschicht oder von Trainingsdaten (Schmiedle, 1997) in Betracht. Die Zeit, die für die Suche zur Verfügung steht, ist allerdings so begrenzt, daß es keinen Sinn macht, mehrere hundert Generationen zu rechnen. Bei der Anwendung dieses Suchverfahrens auf neuronale Netze liegt man weit unter der Generationenzahl, die bei der natürlichen Evolution zur Verbreitung eines Merkmals angenommen wird (vgl. (Wilson, 1995, Dawkins, 1994)).

4.3.3 Lernen

Durch die Verzahnung der beiden Optimierungsprobleme, Topologie einerseits und Gewichtsparameter andererseits, ergeben sich Synergieeffekte, die es erst erlauben, den Suchraum intensiver zu explorieren (Ragg *et al.*, 1997, Braun, 1997). Dies ist vor allem damit zu begründen, daß die Eltern ihre Gewichte an die Nachkommen vererben, und dadurch die Trainingsdauer im Vergleich zu einer Zufallsinitialisierung erheblich verkürzt wird (vgl. die Ausführungen bei (Braun, 1997)). Als Lernverfahren wird Rprop mit Weight-Decay eingesetzt. Der Gewichtungsfaktor für den Strafterm muß dabei experimentell vor Beginn der Evolution bestimmt werden. Die globale Suche beschränkt sich in diesem Fall darauf, diskrete Veränderung durch Mutation vorzunehmen, d.h. Einfügen oder Löschen von Elementen, während lokal bei fester Topologie ein Gradientenabstieg durchgeführt wird. Die Suche verzahnt also den diskreten Raum der Topologie des neuronalen Netzes mit dem reellwertigen Raum der Paramter des Netzes.

4.3.4 Selektion der Eltern

Jede Lösung wird in ENZO durch die Fitneßfunktion bewertet. Dabei werden mittels Kreuzvalidierung der geschätzte Generalisierungsfehler und die Größe der Topologie berücksichtigt. Die Gewichtung der einzelnen Terme ist durch den Benutzer vorzugeben. Für die Selektion wird eine 'elitäre' Strategie verwendet, d.h. die beste Lösung bleibt immer erhalten. Konsequenterweise wird dafür eine $(\mu + \lambda)$-*Evolutionsstrategie* eingesetzt, die aus den λ Nachkommen die Besten unter Verdrängung schlechterer Eltern in die Population einsortiert. Gebräuchliche Werte sind $\mu \in [30, 50]$ und $\lambda \in [10, 20]$, wobei zwischen 30 und 50 Generationen gerechnet wird. Während der Evolution werden also etwa 1000 Netze trainiert. Geht man davon aus, daß man üblicherweise 10 Topologien vergleicht und 50 bis 100 Netze trainiert, aus denen man ein Modell auswählt, dann entsteht durch die Anwendung des evolutionären Verfahrens prinzipiell kein höherer Aufwand. Durch Synergieeffekte läßt sich vielmehr einiges an Aufwand einsparen.

Aus den Eltern werden dann zufällig Individuen ausgewählt, aus denen durch Mutation Nachkommen generiert werden. Fittere Eltern erhalten ein höheres Gewicht, d.h. sie werden wahrscheinlicher ausgewählt. Die Wahrscheinlichkeit hängt dabei nur von ihrem Rang in der Population ab, wie in Abbildung 4.3 dargestellt. Am Ende der Evolution wird das Netz mit der höchsten Fitneß als Modell ausgewählt.

4.3.5 Parameterbestimmung

Ein evolutionärer Algorithmus ist ein heuristisches Verfahren, bei dem einige Parameter vom Benutzer festgelegt werden müssen. Die Wahl der Parameter beeinflußt die Güte der Lösung insofern, als man die zur Verfügung stehende Zeit mehr oder weniger gut ausnutzen kann. Die mittlere Güte der Lösungen, die gefunden werden können, verhält sich in weiten Bereichen des Parameterraumes robust. Optimale Werte lassen sich für die Parametereinstellungen nicht angeben, allerdings einige grundsätzliche Schranken aufzeigen.

Durch die Übertragung von Elterngewichten sinkt die Trainingsdauer der Nachkommen erheblich. Ähnlich wie bei den Pruning-Verfahren ist nach der Mutation nur ein kurzes Nachtrainieren nötig, um wieder in ein lokales Minimum zu gelangen. Bei realen Problemen sinkt der Trainingsaufwand auf ca. 1/10 (Ragg *et al.*, 1997, Braun, 1997). In derselben Zeit, die man benötigt, um 100 Netze konventionell zu trainieren, kann man bei ENZO etwa 1000 Topologien und Parameterbelegungen gegeneinander vergleichen. Daraus lassen sich Schranken für die Größe μ der Population, die Zahl λ der Nachkommen und die Zahl der Generationen ableiten.

Für eine $(\mu + \lambda)$-Evolutionsstrategie tastet man gewissermaßen ausgehend von den in der Population vorhandenen Suchpunkten die Umgebung ab. Dabei macht es keinen Sinn, einen Suchpunkt ohne Mutation in die Nachkommenschaft zu transferieren, da dieser sonst mit seinem Elter identisch ist. Vielmehr empfiehlt es sich im Gegensatz zur natürlichen Evolution, eine hohe Mutationswahrscheinlichkeit anzusetzen. Diese ergibt sich folgendermaßen aus dem Produkt der einzelnen Mutationswahrscheinlichkeiten:

$$p_{Mut} \;=\; 1 - ((1 - p_{Mut-Input})\;\; (1 - p_{Mut-Hidden})\;\; (1 - p_{Mut-Weights}))\,,$$

wobei $p_{Mut-Input}$ die Wahrscheinlichkeit angibt, daß ein Eingabeneuron mutiert wird, $p_{Mut-Hidden}$ und $p_{Mut-Weights}$ entsprechend für versteckte Neuronen und Gewichte. Falls man vor allem den Eingabevektor optimieren möchte, wird man der Mutationswahrscheinlichkeit $p_{Mut-Input}$ einen größeren Wert einräumen als der Mutation der versteckten Neuronen bzw. der Gewichte. Es hat sich auch gezeigt, daß es sinnvoll ist, wenn man von einer großen Topologie ausgeht, zuerst verstärkt Neuronen zu mutieren und erst dann Gewichte. Man kann die beiden Phasen als Grob- und Feinevolution bezeichnen (Braun, 1997, Ragg *et al.*, 1997). In gleicher Weise erhält man für das unitOBS-Verfahren gute Ergebnisse (Stahlberger & Riedmiller, 1997).

In (Ragg *et al.*, 1997) haben wir einige Kriterien, die zur Topologieoptimierung eingesetzt werden können, verglichen bzgl. Komplexität und Leistung. Insbesonders zeigt sich, daß die Vorteile der komplexeren Verfahren wie Optimal Brain Surgeon (OBS) verlorengehen, wenn man Pruning in Kombination mit Regularisierungsverfahren anwendet. Dieser Umstand läßt

sich damit begründen, daß dann sehr viele Gewichte nahe bei Null liegen, die Relevanz also bereits gut mit einem einfachem Kriterium wie dem Betrag des Gewichtes geschätzt wird. Man vergleiche weiter auch die Überlegungen zur Zeitkomplexität von Pruning-Verfahren in (Ragg *et al.*, 1997) und (Braun, 1997), insbesondere die Folgerung, daß sich die Anwendung komplexerer Kriterien zur Topologieoptimierung erst empfiehlt, wenn die Topologie des Netzes hinreichend klein ist bzw. einfachere Methoden ausgeschöpft sind.

Abbildung 4.4 zeigt das Verfahren zusammenfassend in einer algorithmischen Darstellung.

ENZO's Main Loop

```
pre_evolution();          // Initialisierung der Elternnetze
repeat
  selection();            // Auswahl von λ Individuen
  mutation();             // Veränderung zu λ neuen Suchpunkten
                          // durch heuristische Mutation
  optimization();         // Gradientenabstieg bis zum nächsten
                          // lokalen Minimum (mit Weight-Decay)
  evaluation();           // Fitnessberechnung über CV-Fehler
                          // und Größe
  survival();             // Auswahl der μ besten Individuen
until stop_evolution();
```

Abbildung 4.4: Die Abbildung zeigt den evolutionären Algorithmus ENZO mit seinen wesentlichen Komponenten als Hauptschleife eines Programms.

4.4 Ansatzpunkte für ein integriertes Konzept

Drei wesentliche Nachteile der beschriebenen Methodik sind hier zu nennen: Zum einen wird der Weight-Decay Parameter zu Beginn der Evolution fest eingestellt. Aufgrund der Abhängigkeit des Regularisierungstermes von der Größe der Topologie führt das fortschreitende Entfernen von Neuronen und Gewichten dazu, daß der Parameter überschätzt ist. Das heißt, die Netze werden überregularisiert, und somit findet man nur suboptimale Lösungen. Zweitens kombiniert die Fitneßfunktion zwei wesentliche Aspekte: den durch Kreuzvalidierung geschätzten Generalisierungsfehler und die Größe der Topologie. Für das Ergebnis ist die Balance der beiden Terme von Bedeutung. Diese Gewichtung wird vom Benutzer vorgegeben, basiert also auf Erfahrungswerten. Drittens setzt sich im Laufe der Evolution aufgrund des eindimensionalen Fitneßkriteriums eine Lösung in der Population durch. Die Suche kollabiert also in einem ganz bestimmten Punkt des Lösungsraumes, von dem aus dann immer wieder neue Variationen gebildet werden. Im Experiment läßt sich beobachten,

daß nach einer gewissen Zahl an Generationen nur noch selten ein Nachkomme zum Elter wird. Alle anderen 'Entwicklungslinien' werden also abgeschnitten, obwohl sie durchaus zu gleichwertigen oder sogar besseren Lösungen führen könnten. Bezogen auf das Exploration-Exploitations-Dilemma bedeutet das, daß verschiedene Suchpunkte am Anfang mit gleicher Intensität verfolgt werden, aber mit zunehmender Generationenzahl keine Exploration mehr stattfindet. Alle Suchpunkte sind in einem Bereich konzentriert. Viele Lösungen in der Population sind nahezu identisch, d.h. sie berechnen dieselbe Funktion. Eine Suche nach mehreren Optima, die die Datenabhängigkeit geeignet behandelt, ist so nicht möglich.

Zur Lösung des ersten Problems wird in der vorliegenden Arbeit eine effiziente Kombination der evolutionären Suche mit dem Bayes'schen Lernen vorgeschlagen und eingehend untersucht. Effizient heißt dabei, daß durch die Suche kein höherer Zeitaufwand für das Training der Modelle nötig wird. Weiterhin wird in der Bayes'schen Theorie die Grösse der Topologie als sogenannter *Occam-Faktor* bereits berücksichtigt (Bishop, 1995, Gutjahr, 1999). Damit läßt sich die Fitneßfunktion auf das Qualitätsmaß, in diesem Fall die Evidenz, beschränken. Mit dem Wissen um das Bias-Varianz Dilemma läßt sich die Balance zwischen Exploration und Exploitation elegant finden, indem man den Algorithmus so ausrichtet, daß er verschiedenartige Lösungen suchen soll. Es wird eine Methodik vorgeschlagen, die das Kollabieren auf eine Lösung dadurch zu verhindern sucht, daß man die Ähnlichkeit der neuronalen Netze berechnet und die Individuen in der Population bezüglich ihrer Ähnlichkeit in Gruppen zusammenfaßt. Aus der Lösung dieses Problems zieht man dann noch den weiteren Nutzen, verschiedenartige Modelle evolviert zu haben, aus denen in venünftiger Weise ein Komitee gebildet werden kann.

4.5 Evolution unabhängiger Netze

Nachdem alle wesentlichen Bausteine für die Arbeit bereitgestellt sind, soll in diesem Abschnitt das Konzept der Arbeit aufgezeigt und in den nächsten Kapiteln anhand von Experimenten konkretisiert werden. Abbildung 1.5 zeigte die nötigen Schritte ausgehend von den Grundlagen zum Ziel der Arbeit: Problemlösungen durch Komitees neuronaler Netze.

Im folgenden werde ich den evolutionären Algorithmus, wie ich ihn in der Arbeit verwendet habe, entwickeln. Hierzu werde ich in einem konzeptionellen Teil alle Operatoren vorstellen und die einzelnen Aspekte der Operatoren im experimentellen Teil (Kapitel 5 und 6) eingehend untersuchen.

Zuerst wird der oben eingeführte evolutionäre Algorithmus ENZO dahingehend weiterentwickelt, daß die grundlegenden Veränderungen der Nachkommen - Mutation der Topologie und Adaption der Gewichte - auf statistischen Verfahren aufgebaut werden. Bei der Mutation steigt die Zeitkomplexität gegenüber den völlig ungerichteten Mutationsoperatoren, bleibt aber für die meisten praktischen Probleme geringer als die bekannter konnektionistischer Kriterien wie OBS bzw. unitOBS, die in ENZO bereits verwendet wurden. Für OBS ist noch anzumerken, daß es mit einem Regularisierungterm nicht ohne weiteres kombinierbar ist (Ragg *et al.*, 1997). Der Gewichtungsfaktor λ muß bei der Initialisierung des OBS-Verfahrens zur Gewichtung der Einheitsmatrix berücksichtigt werden, was nur im Falle einer Gewichtsgruppe unter Verzicht der Anpassung der Hyperparameter möglich ist.

Bei der Adaption der Gewichte ist in jeder Trainingsphase noch zusätzlich der Hyperparameter λ zu bestimmen. Die zielgerichtete Mutation erspart hier wiederum unnötigen Aufwand, Modelle zu trainieren, die mit nicht geringer Wahrscheinlichkeit verworfen werden. Anhand der Experimente soll aufgezeigt werden, wie man für ein gegebenes Problem eine möglichst optimale Topologie und einen optimalen Gewichtsvektor im Bayes'schen Sinne gleichzeitig bestimmen kann. Die Abbildung 4.1 legt hier nahe, eine (μ, λ)-Strategie einzusetzen. Modelle hoher Evidenz werden zu den Eltern der nächsten Generation und legen damit die Suchpunkte fest, von denen aus weitergesucht wird. Dabei soll verhindert werden, daß man durch beständiges Verwerfen der Nachkommen auf einem lokalen Maximium verharrt, obwohl noch wesentlich bessere Lösungen gefunden werden könnten. Ein weiteres Argument, diese Strategie einzusetzen, ist die Tatsache, daß die Evidenz eines Modells während des Trainings im Bayes'schen Ansatz nicht unbedingt monoton steigt, sondern meist nur tendenziell zunimmt (Abbildung 2.8).

Nach der Optimierung eines neuronalen Modells werde ich das Problem der Balance zwischen Exploration und Exploitation in neuartiger Weise behandeln. Durch die Berechnung der stochastischen Abhängigkeit zwischen zwei Funktionen, genauer den Ausgabefunktionen zweier neuronaler Netze, ist ein Ähnlichkeitsmaß definiert, mittels dessen die Individuen durch ein hierarchisches agglomeratives Clusterverfahren in Gruppen zusammengefaßt werden können. Weiterhin ist es naheliegend, die Individuen innerhalb der Gruppe bezüglich ihrer Evidenz zu ordnen, so daß höhere Evidenz einen Selektionsvorteil bedeutet. Die unabhängigen Modelle können sich in ihrer Güte durchaus unterscheiden.

Aufbauend auf den unabhängigen neuronalen Netzen und dem Varianzkriterium des Bayes'schen Ansatzes (Kapitel 2.43) wird in Kapitel 6 aufgezeigt, wie durch die Evolution Komiteemitglieder gewonnen und in adaptiver Weise durch ihre Varianz für jedes Eingabemuster neu gewichtet werden können. Die für das integrierte Konzept notwendigen Operatoren werden in den folgenden Abschnitten vorgestellt.

4.5.1 Initialisierung

Die einzelnen Sucher der evolutionären Strategien finden jeweils das lokal nächste Maximum der Evidenz. Je besser also die Startbedingungen desto günstiger wird die Suche verlaufen. Abbildung 1.4 zeigte, daß die Güte der Modelle sehr davon abhängt, welche Merkmale man verwendet. Bei der zufälligen Initialisierung der Merkmalsvektoren, d.h. jedes Merkmal hat die gleiche Wahrscheinlichkeit ausgewählt zu werden, verschenkt man sein Wissen darüber, daß bestimmte Merkmale wichtiger sind als andere. Mittels Mutual Information kann man die Merkmale sortieren und auf diese Liste wieder eine Selektionsfunktion wie in Abbildung 4.3 anwenden. Die Sortierung erhält man, indem man eine Top-Down oder Bottom-Up Suche wie in Abbildung 4.5 vor der Evolution durchführt und die Reihenfolge des Löschens bzw. die umgekehrte Reihenfolge des Einfügens als Sortierung übernimmt. Ein Experiment in Kapitel 5 wird zeigen, daß dieses Vorgehen den Suchprozeß stark verbessert, indem zu Anfang überdurchschnittlich gute Suchpunkte gewählt werden.

Die Größe der versteckten Schichten wählt man wie bisher zufällig. Ebenso wird für jedes Netz zufällig eine Teilmenge der Trainingsdaten gezogen. Die Zahl der Muster in einer Teilmenge beträgt dabei 80% bis 90% der gesamten Trainingsmuster.

4.5.2 Mutationsoperatoren

Ein wesentlicher Aspekt bei heuristischen Suchverfahren betrifft die Auswahl der Suchrichtungen. Möchte man beispielsweise die geeignetste Teilmenge aus n Merkmalen auswählen, dann ist es unmöglich, alle 2^n Möglichkeiten durchzuprobieren. Vielmehr kann man mit Einsatz von statistischen Verfahren den Suchbaum enorm verkleinern (vgl.Abbildung 4.5).

Neuronen: Bei der Mutation von Neuronen sollte man zwischen Eingabeneuronen und versteckten Neuronen unterscheiden. Entfernt man ein verstecktes Neuron, dann paßt sich das Netz im allgemeinen in der Lernphase durch die Adaption der Gewichte der neuen Situation an. Es kann sozusagen umlernen, weil die Funktion der versteckten Neuronen nicht von vornherein festgelegt ist. Im Falle eines Eingabeneurons sollte man sehr viel mehr Vorsicht walten lassen, da die Information eines entfernten Neurons immer verloren geht. Anders gesagt: Eine Veränderung der Eingabestruktur ist ein schwerwiegender Eingriff. Im umgekehrten Fall gibt es für versteckte Neuronen beim Einfügen kein Relevanzkriterium.[1] Für Eingabeneuronen kann man aber die linearen und nicht-linearen Abhängigkeiten verwenden, um den vielversprechendsten Kandidaten zu selektieren.

In meinem Ansatz kommen zwei Kriterien zur Bewertung der Neuronen zum Einsatz. Mittels des Korrelationskoeffizienten werden lineare Abhängigkeiten und mittels Mutual Information nicht-lineare Abhängigkeiten gemessen. Das lineare Kriterium ist einfach und schnell zu berechnen, während im anderen Fall zur Bestimmung des (wechselseitigen) Informationsgehaltes eine nicht-parametrische Dichteschätzung, wie in Kapitel 3.6 dargelegt, durchgeführt werden muß. Es sei hier angemerkt, daß eine rechenzeitintensive Optimierung der versteckten Schicht in aller Regel keine Vorteile bringt, da die Regularisierung hier bereits die Löwenarbeit leistet. Es ist vielversprechender, die Zeit für die Optimierung des Eingabevektors oder eine größere Population von Netzen zu verwenden.

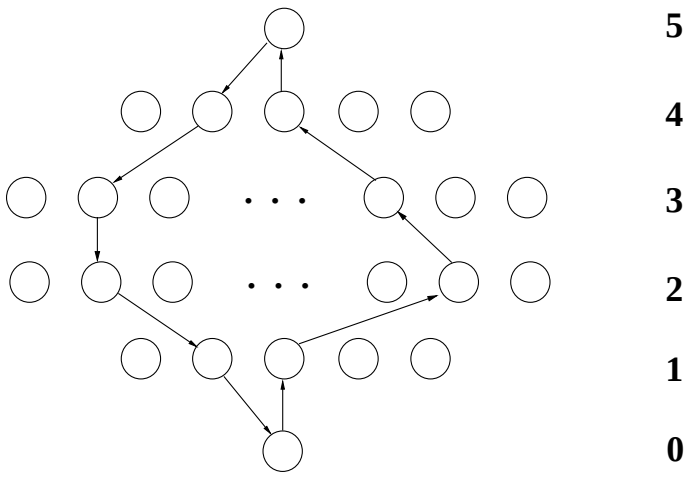


Abbildung 4.5: Die Abbildung zeigt den Suchbaum für alle $\binom{n}{k}$ möglichen Kombinationen von Eingabemerkmalen für $k = 0, \ldots, 5$. Jeder Kreis repräsentiert also eine mögliche Merkmalskombination. Die Pfeile illustrieren ein Top-Down und ein Bottom-Up Suchverfahren, bei denen auf jeder Ebene der lokal günstigste Schritt bezüglich eines Suchkriteriums durchgeführt wird. Dieses Vorgehen garantiert nicht, daß auf jeder Ebene der Vektor mit maximalem Informationsgehalt gefunden wird. Die mangelnde Vorausschau kann durch ein Mischverfahren, z.B. einen evolutionären Algorithmus, ausgeglichen werden, indem Vorwärts- und Rückwärtsschritte möglich sind.

[1] Eine Möglichkeit wäre die Größe der Hyperparameter im Bayes'schen Ansatz für die Auswahl zwischen verschiedenen Gruppen.

Die wesentliche Aufgabe bei der Mutation besteht darin, die möglichen Alternativen geeignet zu sortieren, so daß ein Element gemäß der Funktionen aus Abbildung 4.3 gewählt werden kann. Ich unterscheide das Einfügen und Entfernen von Elementen. Sei jetzt

$$X = (X_1, \cdots, X_{k-1}, X_{k+1}, \cdots, X_{i-1}, X_i, X_{i+1}, \cdots, X_d)$$

der zu betrachtende Vektor von Neuronen der Eingabeschicht oder einer versteckten Schicht, und Y die Ausgabe. X_i, X_j, X_n bezeichnen Komponenten, die in X vorhanden sind, X_k, X_l solche, die gegenüber der Referenztopologie fehlen.

Das einfachere Sortierkriterium basiert auf den Korrelationskoeffizienten zwischen den Komponenten von X. Im Falle des Löschens berechnet man die Korrelation zwischen allen Komponenten von X und entfernt von dem Paar maximaler Korrelation die Komponente, deren Korrelation zur Ausgabe kleiner ist. Im Falle des Einfügens bestimmt man das Paar (X_i, X_k) minimaler Korrelation zwischen den existierenden und fehlenden Komponenten. Damit ergeben sich die folgenden Sortierkriterien:

$$\begin{array}{lll} \text{Löschen:} & X_i \prec X_j & :\Longleftrightarrow \quad (1)\ \max_n \rho(X_i, X_n) > \max_n \rho(X_j, X_n) \\ & & \quad\ (2)\ \text{bei Gleichheit: } \rho(X_i, Y) < \rho(X_j, Y) \\ \\ \text{Einfügen:} & X^k \prec X^l & :\Longleftrightarrow \quad (1)\ \max_i \rho(X_k, X_i) < \max_i \rho(X_l, X_i) \text{ für alle } i \\ & & \quad\ (2)\ \text{bei Gleichheit: } \rho(X_k, Y) > \rho(X_l, Y) \end{array}$$

Definiert man

$$X_{(i)} := (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_d)$$

und

$$X^{(k)} := (X_1, \ldots, X_{k-1}, X_k, X_{k+1}, \ldots, X_d),$$

dann geht beim Übergang zu $X_{(i)}$ die i-te Komponente von X verloren, während $X^{(k)}$ die k-te Komponente einfügt, die X gegenüber der Referenztopologie fehlte. Berechnet man jetzt für alle Vektoren $X_{(i)}$ bzw. $X^{(k)}$, die aus X gebildet werden können, ihren Informationsgehalt bezüglich des Ausgabevektors Y mittels Mutual Information (siehe Kapitel 3.6), dann erhalten wir die folgenden Sortierkriterien für den Mutationsoperator:

$$\begin{array}{lll} \text{Löschen:} & X_i \prec X_j & :\Longleftrightarrow \quad I(X_{(i)}; Y) > I(X_{(j)}; Y) \\ \text{Einfügen:} & X^k \prec X^l & :\Longleftrightarrow \quad I(X^{(k)}; Y) > I(X^{(l)}; Y) \end{array}$$

Das heißt, im Falle des Löschens betrachtet man alle möglichen Vektoren, die eine Komponente weniger haben und wählt einen günstigen aus, wie in Abbildung 4.5 illustriert. Umgekehrt betrachtet man im Falle des Einfügens alle möglichen Vektoren, die eine Komponente mehr haben.

Das Mutual Information Kriterium in ein evolutives Suchverfahren einzubetten wurde erstmalig in (Ragg & Gutjahr, 1997a, Ragg & Gutjahr, 1997b) vorgeschlagen. Es zeigte sich, daß

man auf diese Weise kleinere Eingaberäume finden kann, die das zugrundeliegende Problem immer noch adäquat beschreiben. Die Generalisierungsfähigkeit der neuronalen Netze, die mit Regularisierung trainiert wurden, stieg zudem noch an. In Kapitel 7 wird anhand der Prognose von Absatzzahlen der Nachweis erbracht, daß sich die beiden Kriterien zur Reduktion hochdimensionaler Eingaberäume geeignet einsetzen lassen (vgl. dazu auch (Menzel, 1999, Ragg *et al.*, 2000)).

Muster: Die Suche nach unabhängigen Modellen beinhaltet das Bemühen, die Abhängigkeit von der speziellen Wahl der Trainingsdaten zu minimieren. Hierzu kann man jedem Netz zu Beginn der Evolution zufällig eine Teilmenge aus den Trainingsdaten zuordnen. Im Gegensatz zum Bagging-Verfahren wähle ich ein Ziehen ohne Zurücklegen, wie es in der Bias-Varianz Dekomposition (2.4) bei der Erwartungswertbildung angenommen wird. Der Grund für diese Vorgehensweise liegt darin, daß das Bootstrap-Verfahren voraussetzt, daß über alle Stichproben gemittelt wird. Die Clusterung der Netze läßt aber nur bestimmte Vertreter und damit Stichproben zu. Weiterhin ist es offensichtlich, daß Netze, die nur auf wenigen Mustern trainiert werden, welche dafür aber mehrfach vorkommen, einen kleinen Trainingsfehler und eine große Evidenz haben. Je weniger Muster vorhanden sind, desto eher erhält man eine lineare Lösung. Mutiert man jetzt die Zusammenstellung des Datensatzes, dann besteht ein Selektionsdruck dahingehend, möglichst wenige verschiedene Muster zum Training zu verwenden, da diese Netze eine höhere Evidenz haben werden.

Diese Zusammensetzung des Datensatzes kann man während der Evolution ebenso Mutationen aussetzen, indem man zufällig Muster hinzunimmt und andere streicht. Die Trainingsdaten werden dazu für jedes Netz zusätzlich in einen Genstring bestehend aus Bits abgebildet, bei dem 1 bedeutet, daß das Muster zum Training des Netzes verwendet wird. Man definiert jetzt eine Mutationswahrscheinlichkeit P_{Flip}, die die Häufigkeit angibt, mit der eine Mutation des Mustersatzes auftritt. Einige Vorteile und Eigenschaften der Evolution von Mustermengen wurden hier am Institut in einer Studienarbeit untersucht (Schmiedle, 1997). Dabei wurden allerdings wesentlich weitergehende Operatoren betrachtet, als ich sie in meiner Arbeit aus Aufwandsgründen einsetzen kann.

Gewichte: Gewichte werden im hier vorgeschlagenen Konzept weder entfernt noch hinzugefügt, um zu verhindern, daß Gewichtsgruppen im Bayes'schen Ansatz durch starkes Ausdünnen zu klein und damit die Schätzungen der Hyperparameter ungenau werden. Trotzdem ist es sinnvoll, die Gewichte einer Mutation zu unterwerfen, wie ich das weiter unten begründen werde. Die Mindestanzahl an Gewichten macht es auch erforderlich, eine Minimaltopologie zu definieren, die den Suchraum nach unten begrenzt. Diese Topologie wird so gewählt, daß in jeder Gewichtsgruppe mindestens 4 Gewichte vorhanden sind. Das soll gewährleisten, daß die Approximation der Evidenz für α und β (Gleichungen (2.31) und (2.33) in Verbindung mit (2.37)) auf genügend Datenpunkten basiert.

Bei Verwendung einer (μ, λ)-Strategie besteht ein möglicher Mutationsoperator darin, auf die Gewichte normalverteilte Zufallszahlen z_i zu addieren, um die neuen Suchpunkte im Bereich des bisherigen Punktes zu streuen. Die zugrundeliegende Verteilung wählt man mit Mittelwert 0 und Varianz σ. Den neuen Gewichtsvektor erhält man, indem man zu jeder

Komponente das z_i-fache ihres eigenen Wertes addiert:

$$w_i^{Neu} = w_i + z_i w_i$$

Je nach Problem kann man die Varianz dadurch geeignet einstellen, daß man ausgehend von einem Netz Nachkommen generiert und die Streuung der Nachkommen betrachtet. Sind die Lösungen nahezu identisch, dann ist die Varianz zu klein gewählt, sind sie ähnlich gestreut wie Netze, die aus verschiedenen Zufallsinitialisierungen gewonnen wurden, dann ist die Varianz zu groß gewählt. Ein guter Erfahrungswert ist $\sigma = 10^{-4}$.

Der zweite Mutationsoperator, der später bei der Evolution unabhängiger Netze betrachtet wird, hält die Topologie fest und initialisiert den Gewichtsvektor neu. Das heißt, es wird ein neuer Suchpunkt im Gewichtsraum definiert. Das Ziel dabei ist, eine weiter möglichst unabhängige Lösung zu finden, die die bisherigen Suchpunkte sinnvoll ergänzt.

Große Mutationen, wie z.B. das Entfernen von Neuronen, haben eine geringe Wahrscheinlichkeit, damit der Clusterungsprozeß über die Generationen hinweg einen stabilen Zustand erreicht. In Kapitel 6 wird dies ausgeführt. Daher ist diese Modifikation der Gewichte notwendig, damit man in der nächsten Generation keine identischen Netze betrachtet. Sie ist sinnvoll, da beim Bayes'schen Ansatz einige Approximationen gemacht werden, um die Parameter einstellen zu können. Man denke z.B. an den iterativen Prozeß zur Bestimmumg der Hyperparameter aus Kapitel 2.5.2. Es lohnt sich also unter diesen Umständen, auch in der Umgebung zu suchen. Im nächsten Abschnitt wird auf die Integration des Bayes'schen Lernens in das globale Suchverfahren eingegangen und gezeigt, das die iterative Berechnung der Hyperparameter eine effiziente Lösung zuläßt.

4.5.3 Bayes'sches Lernen und Evolution

Die Umsetzung der Bayes'schen Methode für neuronale Netze wurde in Kapitel 2.5.2 als iterativer Prozeß dargestellt. Die Iteration war notwendig, weil die Forderung, einen optimalen Gewichtsvektor und optimale Hyperparameter gleichzeitig zu bestimmen, nicht analytisch zu erfüllen ist. Durch das iterative Verfahren versucht man, diesen Werten möglichst nahe zu kommen. Für ein integriertes Optimierungskonzept ist das die geeignete Schnittstelle, um die Suche mit dem Lernprozeß zu verzahnen. Trainiert man eine Population von Netzen, dann verfolgt man nicht alle Entwicklungslinien bis zum Ende, um dann das Netz mit der höchsten Evidenz auszuwählen, sondern generiert in jedem Iterationsschritt Kopien mit leichten Mutationen. Netze mit höherer Evidenz bekommen eine höhere Wahrscheinlichkeit, sich fortzupflanzen - auf Kosten der schlechteren Netze.

Abbildung 4.6 illustriert die Idee an einem kleinen Beispiel. Für den Datensatz aus Abbildung 1.1a wurden 10 Netze mit Bayes'schem Lernen trainiert. Im Vergleich dazu wurde eine Mini-Evolution durchgeführt, die nach jedem zweiten Iterationsschritt neue Suchpunkte für den nächsten Schritt erzeugt hat, mit einem Selektionsdruck zugunsten der Netze mit höherer Evidenz. Insgesamt wurden die Netze 1000 Epochen trainiert, wobei nach jeweils 100 Epochen eine Anpassung der Hyperparameter erfolgte. Der Evolutionäre Algorithmus beschneidet sozusagen den Suchbaum, in dem er vielversprechenden Initialisierungen mehr Gewicht gibt und Netze mit kleiner Evidenz bereits am Anfang verwirft.

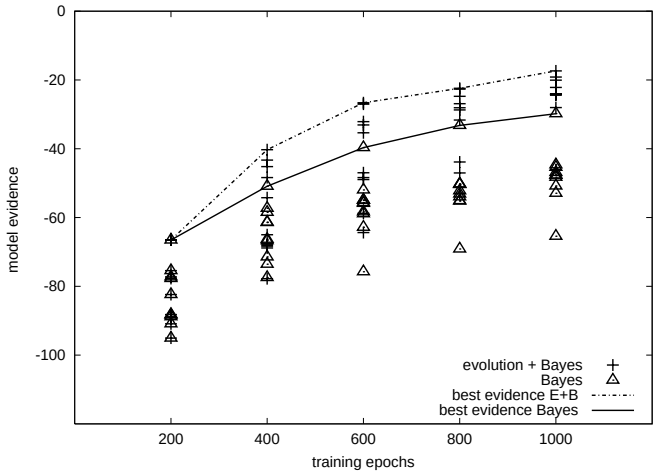


Abbildung 4.6: Die Abbildung zeigt die Entwicklung der Evidenz für 10 neuronale Netze. Nach jeder zweiten Anpassung der Hyperparameter, in diesem Fall alle 200 Epochen, werden Nachkommen gebildet. Ausgehend von der gleichen Initialisierung favorisiert die evolutionäre Strategie Lösungen mit höherer Evidenz, während das normale Verfahren alle Lösungen mit gleicher Intensität weiterverfolgt. Die maximale Evidenz liegt für das einfache Bayes'sche Verfahren deutlich unter der verzahnten Optimierung. Ist die Evidenz negativ mit dem Generalisierungsfehler korreliert, dann werden mit dem selben Aufwand durch die Evolution bessere Netze gefunden.

Es bleibt als zentrale Frage zu beantworten, wie die Verzahnung im Detail zu leisten ist. Betrachten wir zunächst die Extremfälle, daß nach jedem Iterationsschritt Nachkommen erzeugt werden, bzw. daß erst nach Konvergenz der Hyperparameter zur nächsten Generation übergegangen wird. Im letzten Fall steigt der Aufwand der Evolution gegenüber dem normalen Training unvertretbar stark an. Auch zieht man keinen Nutzen daraus, daß man schon vorher Information über die Güte der Lösung hat. Ein Iterationsschritt besteht immer aus zwei Teilschritten: das Anpassen der Hyperparameter und die anschließende Optimierung des Gewichtsvektors. Ist ein lokales Minimum der Fehlerfunktion erreicht, dann liefert uns das Evidenzkriterium ein Maß dafür, wie gut unsere Hyperparameter waren (Abbildung 4.7). Im ersten Iterationsschritt des Bayes'schen Verfahrens sind die Hyperparameter vorgegeben, entweder durch Initialisierung in der ersten Generation oder später von den Eltern überliefert. Durch den Mutationsschritt könnten sie verrauscht sein. Die Evidenz, die am Ende berechnet wird, basiert aber auf den Hyperparametern, mit denen trainiert wurde.

Da die Evidenz als Gütekriterium verwendet wird und erst am Ende der Fehlerminimierung vorliegt, scheint es sinnvoll, wenigstens zwei Iterationen durchzuführen, bevor die Nachkommen für die nächste Generation erzeugt werden. Nochmal: Die berechnete Evidenz des Netzes am Ende der Fehlerminimierung bezieht sich auf die Hyperparameter, mit denen die Fehlerminierung durchgeführt wurde. Erst nach der zweiten Fehlerminimierung liegt also eine Schätzung der Güte vor, die sich tatsächlich auf die Hyperparameter bezieht, die durch das Bayes'sche Verfahren 'optimal' eingestellt wurden. Abbildung 4.7 illustriert den Sachverhalt.

Daraus folgt, daß das Generieren von Nachkommen nach jedem Anpassungsschritt mit einem stärkeren Rauschen behaftet ist und zu schlechteren Ergebnissen bei gleichem Aufwand führen sollte. Im nächsten Kapitel wird durch Experimente belegt, daß die vorgeschlagene Art der Verzahnung die geeignete ist.

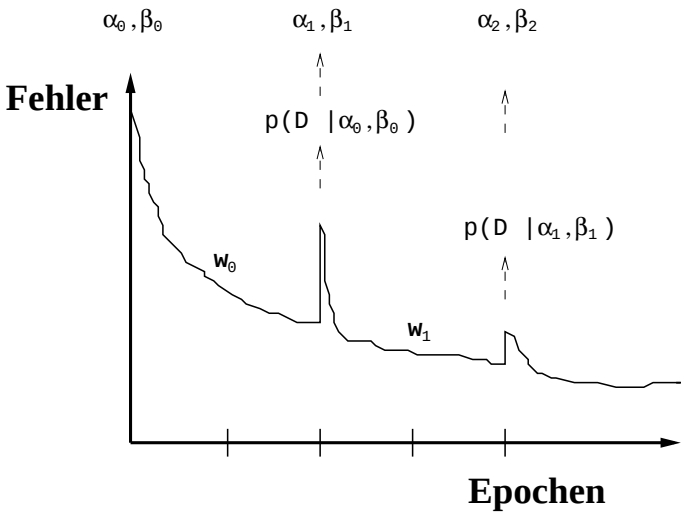


Abbildung 4.7: Illustration der iterativen Anpassung der Hyperparameter durch das Bayes'sche Verfahren. Die Abbildung zeigt die Entwicklung des Fehlers während des Trainings. Zu Beginn werden die Hyperparameter mit α_0 und β_0 initialisiert. Anschließend wird ein Gewichtsvektor $\mathbf{w}_0$ durch Training bestimmt. Man beachte, daß die Form der Fehlerfunktion durch die Parameter α_0 und β_0 bestimmt wird (Gleichung 2.12). Liegt dieser Gewichtsvektor vor, dann kann die Evidenz $p(D|\alpha_0, \beta_0)$ und damit auch die Modellevidenz bezüglich dieser Werte berechnet werden. Nach der zweiten Anpassung liegt mit $p(D|\alpha_1, \beta_1)$ erstmals ein Wert der Evidenz vor, der auf Hyperparametern basiert, die durch das Bayes'sche Verfahren eingestellt wurden.

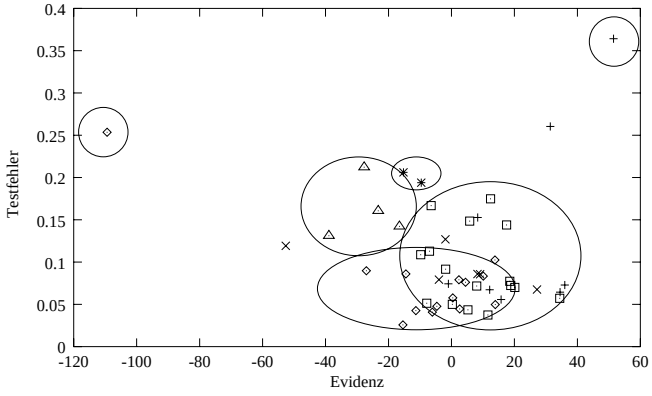
4.5.4 Selektion unabhängiger Eltern

Der Algorithmus, wie ich ihn bisher vorgestellt habe, ist ein Optimierungsverfahren für die Evidenz. Am Ende des Prozesses bildet sich eine Lösung heraus - das Netz mit der höchsten Evidenz. Das Ziel dieser Arbeit ist es, ein Komitee aus mehreren Netzen zu synthetisieren. Im vorigen Kapitel wurde dargelegt, daß es dazu notwendig ist, den Suchprozeß so zu gestalten, daß verschiedenartige Funktionen gelernt werden können. Diese Vielfalt soll während der Evolution erhalten bleiben. Mit dem Kriterium (3.12) habe ich ein analytisches Kriterium hergeleitet, das in jedem Evolutionsschritt optimiert werden kann. Das heißt, in jedem Schritt können wir die Suchrichtungen bestimmen, die maximal voneinander unabhängig sind. Im folgenden werde ich eine Selektionsstrategie angeben, die diese Suchrichtungen möglichst in die nächste Generation transportiert. Dazu sind zwei Aspekte unter einen Hut zu bringen: Die Optimierung der Evidenz der einzelnen Netze und die Unabhängigkeit dieser Netze.

Sind in einer Generation alle Netze trainiert, dann wählt man zuerst die μ besten Individuen aus, d.h. die Netze mit der höchsten Evidenz und bestimmt dann sinnvollerweise die beste Klasseneinteilung, wie in Abbildung 3.1 im letzten Kapitel gezeigt (Ragg, 2000). Die Klassen kann man als Suchrichtungen auffassen, die maximal voneinander unabhängig sind im Sinne des Kriteriums (3.12). Das Ergebnis dieser Klassenbildung ist in Abbildung 4.8 für ein Beispiel gezeigt. Jedes neuronales Netz ist durch einen Datenpunkt im Evidenz/Testfehler Raum repräsentiert. Die Darstellung gibt die Ähnlichkeitsbeziehung nur verzerrt wieder. Netze mit der gleichen Ausgabefunktion haben einen ähnlichen Testfehler. Die Evidenz kann aber stark variieren, da sie auch von der Netzkomplexität abhängt. Andererseits können unterschiedliche Netze trotzdem einen ähnlichen Testfehler haben.

Hat man die Klasseneinteilung bestimmt, dann ergeben sich einige Möglichkeiten, die Eltern der nächsten Generation zu gewinnen.

Abbildung 4.8: Die Abbildung zeigt die Clusterung neuronaler Netze bei dem Datensatz aus Abbildung 1.1b und der Klasseneinteilung wie in Abbildung 3.1. Die Daten sind über der Evidenz und dem Testfehler aufgetragen und in Gruppen zusammengefaßt. Diese Darstellung gibt die Nachbarschaftsrelation nur verzerrt wieder. Im allgemeinen haben Netze mit ähnlicher Ausgabefunktion auch einen ähnlichen Fehler, die Evidenz kann aber je nach Komplexität durchaus streuen. Umgekehrt gibt es natürlich auch verschiedenartige Netze, die einen ähnlichen Fehler haben können. Sechs der acht Cluster sind zur besseren Veranschaulichung durch Ellipsen markiert. Die Abbildungen 4.9 und 4.10 zeigen die zugehörigen Ausgabefunktionen.



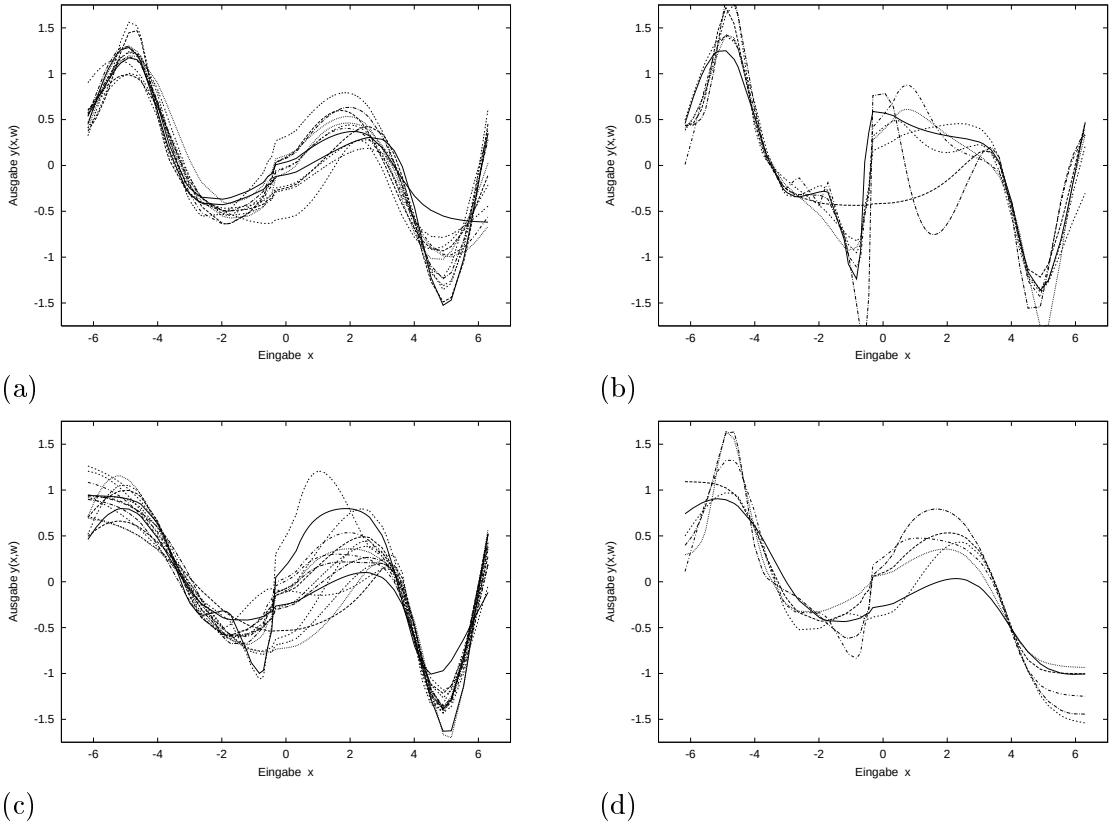


Abbildung 4.9: Die Abbildung zeigt die Ausgabefunktionen der neuronalen Netze der ersten vier Cluster. Jeweils alle Netze eines Clusters sind in einem Bild zusammengefaßt. Die Gruppen in a)-d) entsprechen in Abbildung 4.8 den Netzen rechts unten. Alle Netze haben eine relativ große Evidenz und tendentiell einen kleineren Fehler. Abbildung 4.10 zeigt die anderen vier Cluster.

1. Innerhalb jeder Klasse wählt man das Netz mit der höchsten Evidenz aus und hat

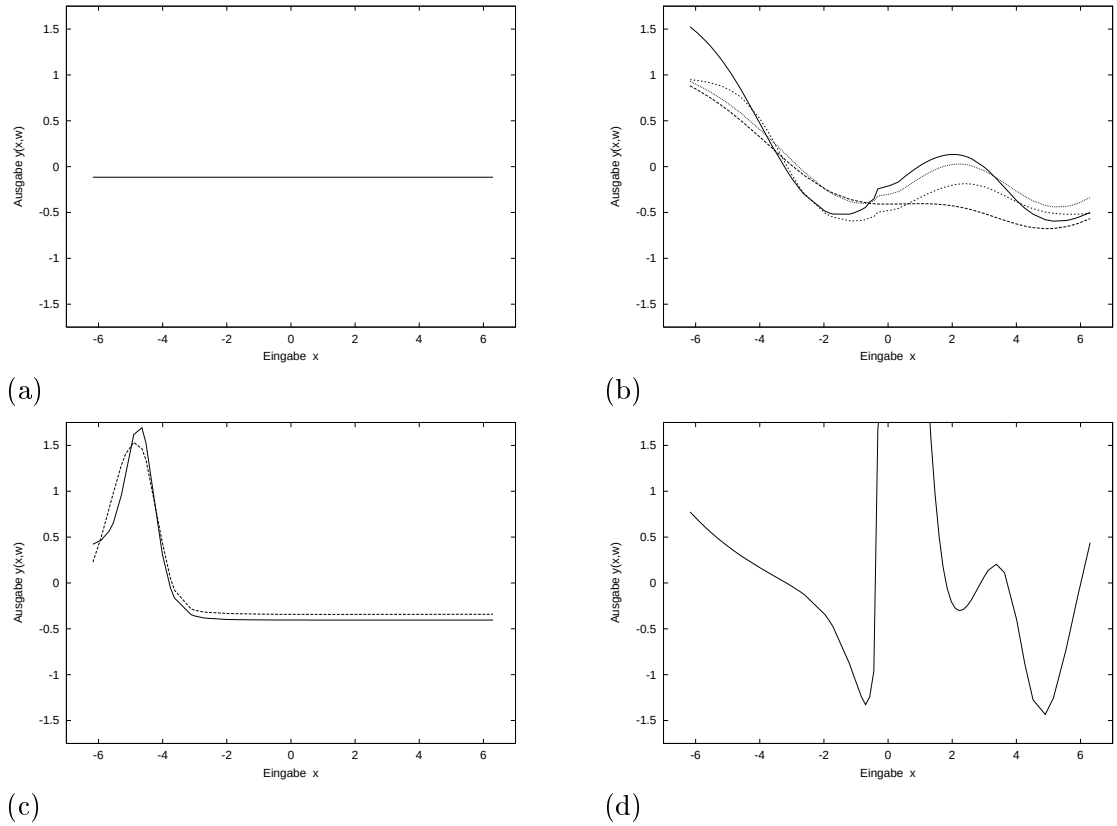


Abbildung 4.10: Die Abbildung zeigt die Ausgabefunktion der restlichen Cluster aus Abbildung 4.8. Jeweils alle Netze eines Clusters sind zusammengefaßt. Unter a) ist die Funktion des Netzes, das in Abbildung 4.8 rechts oben eingezeichnet ist. In diesem Fall sind die Hyperparameter extrem groß, so daß die Ausgabefunktion konstant ist. Dies ist auf den hohen Rauschanteil und eine ungünstige Initialisierung zurückzuführen. Da solche Netze eine hohe Evidenz aufweisen, ergibt sich ein Problem, auf das ich nochmals zu sprechen komme. b) und c) zeigen Netze mit größerem Fehler, die jeweils einen linearen Anteil aufweisen. d) zeigt ein Netz mit großem Fehler und kleiner Evidenz, das insbesondere in dem Bereich, in dem keine Trainingsdaten vorhanden waren, stark overfittet. Insgesamt ist gut zu erkennen, daß die Klasseneinteilung vernünftig bestimmt worden ist.

damit genauso viele Eltern wie Klassen. Der Parameter μ hat bei dieser Variante also nur eine untergeordnete Bedeutung. Er legt fest, welche Netze zur Klassenbildung überhaupt betrachtet werden und sorgt für einen Selektionsdruck in Richtung höherer Evidenz. Ohne diesen Selektionsdruck könnten sich auch die schlechtesten Netze dauerhaft als eigene Klassen in der Popuation etablieren.

2. Man sortiert die Netze in jeder Klasse gemäß ihrer Evidenz. Für jede Klasse wählt man jetzt anteilig Netze aus, wobei diejenigen mit hoher Evidenz, sich mit größerer Wahrscheinlichkeit fortpflanzen gemäß der Auswahlfunktion von Abbildung 4.3.

3. Führt man die Clusterung durch, bis alle Klassen zusammengelegt sind, dann kann man zu jeder Klasse noch angeben, wie 'unabhängig' sie von den anderen ist. Die beiden Klassen, die am Ende verschmolzen werden, geben zwei Suchrichtungen mit maximaler Unabhängigkeit an. Mit der Reihenfolge der Zusammenlegung ist also eine

Reihenfolge der Klassen festgelegt. Abbildung 4.11 veranschaulicht in einem Dendogramm, wie die Reihenfolge zustande kommt. Auf diese sortierte Liste kann man jetzt wieder eine Auswahlwahrscheinlichkeit so festlegen, daß 'größere Unabhängigkeit' eine höhere Fortpflanzungswahrscheinlichkeit bedeutet.

Die erste Variante ist die restriktivste Auswahl. Bilden sich in einem Schritt wenige, große Klassen heraus, dann sind die beiden anderen Strategien robuster gegen ein Kollabieren auf wenige Lösungen, da sich aus einem Cluster mehrere Netze fortpflanzen können.

Für Evolutionsstrategien kennt man das Prinzip *der großen Variation und der kleinen Vererbung*. Dieses besagt, daß man eine erfolgreiche Variation eines Parameters gedämpft vererben sollte (Rechenberg, 1994). Der Sinn davon ist, beim Optimieren im Rauschen, einen Störpegel nicht als erfolgreiche Parametereinstellung zu übernehmen. Man bewegt sich also nur langsam in die Richtung der erfolgreichsten Schrittweite und nicht sprunghaft. Übertragen auf die Festlegung der verschiedenen Suchrichtungen durch Klassenbildung bedeutet das, daß die optimale Klassenzahl ebenso langsam verändert werden sollte, da sie einem Rauschen unterliegt. Das Rauschen kommt durch die zufällige Auswahl der Nachkommen zustande. So verschiebt sich in jeder Generation die optimale Klassenzahl. Es ist intuitiv verständlich, daß beispielsweise ein Wechsel von 15 auf 3 Suchrichtungen und dann wieder zurück auf 9 für Instabilitäten in der Evolution sorgen wird. Es macht also Sinn zu untersuchen, ob eine 'gedämpfte' Bestimmung der Klassenzahl zu einem günstigeren Fortschreiten der Suche führt.

Im experimentellen Teil der Arbeit werden die Strategien nochmals näher beleuchtet. In jedem Fall wird man bei der multimodalen Suche die Mutationswahrscheinlichkeiten so einstellen, daß die Zahl der Klassen sich im Laufe der Zeit stabilisiert und somit die Suche konvergiert. Im nächsten Abschnitt führe ich dazu einige grundsätzliche Überlegungen aus.

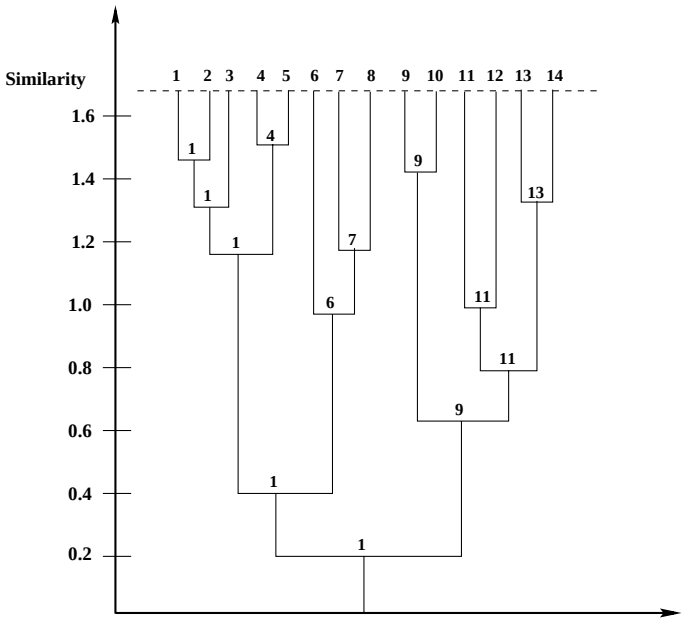


Abbildung 4.11: Die Abbildung zeigt ein hypothetisches Beispiel für die Zusammenlegung von Klassen. In jedem Schritt werden die beiden Klassen zusammengelegt, die sich am ähnlichsten sind, und mit der Nummer der kleineren Klasse versehen. Die Ähnlichkeit läßt sich auf der vertikalen Achse ablesen. Angenommen man bricht die Clusterung bei einer Ähnlichkeit von ca. 1.0 ab, so daß die Klassen 1,6,9 und 11 übrig sind. Dann definiert der Zeitpunkt der Zusammenlegung eine Rangfolge auf den Klassen. In dem Beispiel ist die Reihenfolge 1,9,11 gefolgt von 6.

4.5.5 Gewichtung der Komiteemitglieder

Am Ende der Evolution liegen die Vertreter der zuletzt gefundenen Klasseneinteilung als Kandidaten für ein Komitee vor. Während der Suche war es nur das Ziel, möglichst unabhängige Netze zu finden, die auch gleichzeitig eine große Evidenz haben. Kombiniert man diese nun zu einem Komitee, dann bestehen mehrere Möglichkeiten, die einzelnen Mitglieder zu gewichten.

Zum einen kann man die Korrelationen der Fehlerfunktionen nutzen, um einen Gewichtungsvektor $\gamma_1, \ldots, \gamma_L$ zu finden unter der Nebenbedingung $\sum_i \gamma_i = 1$ (Hashem, 1999, Bishop, 1995). Im Bayes'schen Ansatz ergeben sich Komitees in natürlicher Weise, indem man die Gewichte gemäß ihrer Evidenz und des gewählten Priors berechnet (Bishop, 1995). Bishop führt allerdings auch aus, daß bei praktischen Anwendungen die Gewichtung auf Basis der Evidenz oft zu schlechteren Ergebnissen führt (Bishop, 1995), siehe auch (Thodberg, 1993). Stattdessen schlägt er vor, die Evidenz nur als Auswahlkriterium zu verwenden, aber die Netze alle gleich zu gewichten. Oft erreicht man mit einer einheitlichen Gewichtung wie bei Bagging die besten Ergebnisse.

In dieser Arbeit werde ich mich auf die einfache Gewichtung der Netze beschränken. Das erlaubt es, die Fehlerbalken für die Ausgabe, wie sie die Bayes'sche Methode berechnet, sinnvoll in den Komiteebildungsprozeß einzubringen. Je größer die Varianz, desto unsicherer ist sich das Netz bei dem gegebenen Datenpunkt und desto weniger sollte seine Ausgabe berücksichtigt werden. Man vergleiche dazu nochmals die Abbildung 2.3.

Für jedes Netz berechnet man die Varianz σ_i^2 der Ausgabefunktion $y_i(\mathbf{x})$ für jedes Muster $\mathbf{x}$ gemäß Gleichung (2.43). Der Gewichtungsvektor $\mathbf{v}$ für das Komitee ergibt sich dann wie in Gleichung (3.1), indem man für die Werte v_i die Kehrwerte der Varianz einsetzt. Man beachte, daß der Vektor noch zu 1 normiert werden muß.

$$y_{COM}(x) := \frac{1}{\sum_i v_i} \sum_{i=1}^{L} v_i y_i(x). \tag{4.1}$$

Für Klassifikationsprobleme lohnt es sich, hier den Ansatz von Gutjahr zur Berechnung von Verlustwahrscheinlichkeiten aufzugreifen und diese als Gewichtung zu verwenden (Gutjahr, 1999). Im folgenden setze ich der Einfachheit halber voraus, daß der Erwartungswert der Zielwerte gerade 0 ist. Ansonsten sind die Formeln durch eine Transformation des Mittelwertes entsprechend anzupassen.

Sei T_x eine bedingte Zufallsvariable mit der Dichtefunktion $p(t|x, \alpha, \beta, D)$, so interessieren wir uns für die Wahrscheinlichkeit $P(T_x < 0)$, falls die Ausgabe des neuronalen Netzes $y_{opt} = y(\mathbf{x}, \mathbf{w}_{opt})$ größer als 0 ist und $P(T_x > 0)$ im umgekehrten Fall. Man berechnet also für jede Ausgabe die Wahrscheinlichkeit, daß diese auf der falschen Seite des Mittelwertes liegt. Im Falle einer Absatzprognose entspricht das der Wahrscheinlichkeit, daß das Netz einen steigenden Absatz vorhersagt, obwohl dieser tatsächlich fällt.

Für $y_{opt} > 0$ ergibt sich

$$P(T_x < 0) = P\left(\frac{T_x - y_{opt}}{\sigma_t} < -\frac{y_{opt}}{\sigma_t}\right) = \Phi\left(-\frac{y_{opt}}{\sigma_t}\right) \tag{4.2}$$

und für $y_{opt} < 0$ gilt entsprechend

$$P(T_x > 0) = 1 - P(T_x < 0) = 1 - \Phi\left(-\frac{y_{opt}}{\sigma_t}\right). \tag{4.3}$$

$\Phi(x)$ ist dabei der Wert der Verteilungsfunktion der Standard-Normalverteilung an der Stelle x, der tabelliert vorliegt (Büning & Trenkler, 1994, Henze, 1995). Abbildung 4.12 illustriert die Idee.

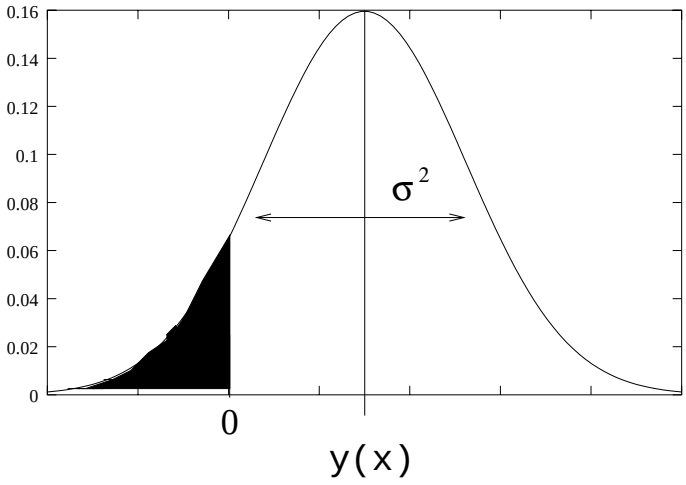


Abbildung 4.12: Die Abbildung zeigt die bedingte Wahrscheinlichkeit für ein festes Eingabemuster x, die ein neuronales Netz berechnet. Die Ausgabe des Netzes y_{opt} entspricht dem Maximum der Funktion. Die Verlustwahrscheinlichkeit entspricht gerade der schwarz gefärbten Fläche unter der Kurve links von dem Mittelwert der Zielwerte, d.h. im Bild links von 0. Die Abbildung wurde von Gutjahr übernommen.

In Gleichung (4.1) werden dann für $1/v_i$ die für jedes Netz berechnete Verlustwahrscheinlichkeit eingesetzt.

4.5.6 Parameterbestimmung

Wie bereits oben ausgeführt, hat ein evolutionärer Algorithmus als heuristisches Verfahren einige Parameter, die festgelegt werden müssen. Ein ganz wesentlicher Vorteil der hier dargelegten Vorgehensweise ist, daß der Benutzer das Optimierungskriterium des Verfahrens nicht mehr selber definieren muß durch Gewichtung der Komplexität gegen den Kreuzvalidierungsfehler. Vielmehr ist das Ziel durch analytische Kriterien genau bestimmt. Die Möglichkeit, daß zufälllig eine gute Wahl der Parameter zu Overfitting führt, ist also nicht mehr gegeben. Für die Konvergenz des Verfahrens sind trotzdem einige grundsätzliche Schranken für die Parameter zu beachten.

Die Größe μ der Population, die Zahl λ der Nachkommen und die Zahl der Generationen leiten sich aus dem Aufwand ab, der für das Training mit Bayes'schem Lernen normalerweise nötig wäre. Geht man davon aus, daß man für eine Modellentwicklung 100 Netze trainiert und jeweils 10 Iterationen zur Adaption der Hyperparameter durchführt, dann ergibt das 500 Nachkommen in der Evolution, wenn man nach jeder zweiten Iteration Nachkommen generiert. Setzt man für das Optimieren der Zahl der versteckten Neuronen und der Eingabestruktur jeweils noch einen Faktor vier an, dann ergibt das einen Umfang von 8000

Suchpunkten. Wählt man beispielsweise λ zu 100, dann beträgt die maximale Generationenzahl 80. Die Populationsgröße entspricht im Falle der restriktiven multimodalen Evolution gerade der Klassenzahl. Im anderen Fall wähle ich $\mu \in [\frac{\lambda}{2}; \lambda]$. Bei kleinerer Generationenzahl sollte auch μ kleiner gewählt werden. Grundsätzlich sollte μ mit wachsendem Rauschen in der Qualitätsfunktion zunehmen (Rechenberg, 1994). Es empfiehlt sich, μ etwas kleiner zu wählen als λ. Das sorgt dafür, daß in jeder Generation die schlechtesten Suchpunkte mit Sicherheit aussortiert werden, wodurch die Konvergenz beschleunigt wird. In den meisten Experimenten war $\mu = 0.9 \cdot \lambda$.

Für eine (μ, λ)-Evolutionsstrategie streut man in jeder Generation ausgehend von den in der Population vorhandenen Individuen neue Suchpunkte in deren Umgebung. Auf jedes Individuum wird eine normalverteilte Zufallszahl auf den Parametervektor addiert. Zusätzlich werden noch größere Mutationsschritte, d.h. auf Neuronenebene oder Musterebene, durchgeführt. Damit die Clusterbildung sich im Laufe der Evolution stabilisiert, empfiehlt es sich, große Mutationen selten durchzuführen, da durch große Schritte im Suchraum mit höherer Wahrscheinlichkeit eine neue Klasse entsteht. Die Wahrscheinlichkeit einer großen Veränderung

$$p_{Mut} = 1 - ((1 - p_{Mut-Input})(1 - p_{Mut-Hidden})(1 - p_{Mut-Pattern}))$$

sollte also klein bleiben, z.B. bei $p_{Mut} \in [0; 0.3]$. Je nach Problemstellung kann man einem der Operatoren ein größeres Gewicht verleihen. Im allgemeinen ist es sinnvoll, die Mutation von Eingabeneuronen höher zu gewichten als die von versteckten Neuronen, da dieser Suchraum größer ist. Günstige Merkmalskombinationen zu finden erfordet mehr Suchschritte.

Damit ist die Beschreibung des integrierten Optimierungskonzeptes vollständig. In den folgenden Kapiteln werden einzelne Gesichtspunkte des Algorithmus an ausgewählten Beispielen verdeutlicht und dann die Leistung des Verfahrens anhand realer Problemstellungen untersucht.

4.6 Zusammenfassung

Soll eine Funktion mittels neuronaler Netze approximiert werden, dann beeinflussen mehrere Designentscheidungen die Qualität des Ergebnisses: Problemkodierung, Topologie, Auswahl der Trainingsdaten, Parameter des Netzes und im Falle, daß ein Komitee verwendet wird, dessen Zusammensetzung. Für keines dieser Teilprobleme läßt sich eine optimale Lösung mit klassischen Methoden finden. Mehr noch, die Einstellungen einzelner Parameter beeinflussen sich gegenseitig.

Evolutionäre Strategien eignen sich, um Optima einer verrauschten Qualitätsfunktion zu finden. Dadurch daß die Nachkommen um die Eltern gestreut werden, bewegen sich die Individuen tendenziell eher in Richtung lokaler Optima. Sie folgen auch bei diskreten Parametern einer Art Gradientenweg. Damit bieten sie einen geeigneten Rahmen, die beim Training neuronaler Netze auftretenden Probleme zu lösen.

Der evolutionäre Algorithmus ENZO setzt die Suche nach guten Modellen um, indem Lernen und Topologieoptimierung kombiniert werden. Die Qualitätsfunktion wird dabei wie

in vergleichbaren Ansätzen auch durch Kreuzvalidierung bestimmt. Mit der selben Technik wird die Gewichtung des Regularisierungstermes vor Beginn der Evolution festgelegt und während der Suche beibehalten. Die Suchschritte werden mittels heuristischer oder basierend auf konnektionistischen Kriterien festgelegt.

Das in diesem Kapitel vorgestellte integrierte Optimierungskonzept baut auf der Grundidee von ENZO auf. Die Kombination mit dem Bayes'schen Lernen erlaubt es, auf Kreuzvalidierungstechniken zu verzichten, indem die Evidenz als Optimierungskriterium verwendet wird. Dabei wird die Erzeugung von Nachkommen mit dem Iterationsverfahren zur Adaption der Hyperparameter verzahnt. Erst dadurch ist eine effiziente Suche möglich. Diese Suche nach dem wahrscheinlichsten Modell läßt sich auf die Anzahl der versteckten Neuronen und den Eingabevektor ausdehnen. Durch Einsatz von statistischen Verfahren, z.B. Mutual Information, kann man geeignete Suchschritte berechenen und spart dadurch systematischen Suchaufwand ein. Dadurch daß die statistischen Kriterien nur auf den Daten operieren, kann man vor Start der Evolution auch eine Reihenfolge der Wichtigkeit aller Eingaben erstellen und damit die Population besser initialisieren. Damit ist ein wesentlicher Teil des Trainings neuronaler Netze vollständig automatisiert.

Die Suche nach einem Optimum der Evidenz hängt von den Anfangsbedingungen ab, insbesondere der Gewichtsinitialisierung und der zum Training verwendeten Daten. Eine evolutionäre Strategie, die die Evidenz optimiert, kollabiert nach einer gewissen Zahl an Generationen. Alle anderen Lösungen gehen damit verloren. Ausgehend von der Bias-Varianz Dekomposition wurde gezeigt, daß es aus mehreren Gründen sinnvoll ist, über mehrere Netze zu integrieren. Es wurde ein Kriterium hergeleitet, mit dem sich Komitees auf Basis der Ähnlichkeit der einzelnen Netze vergleichen lassen. Mit einem Cluster-Verfahren läßt sich dieses Kriterium optimieren, um aus einer Menge von trainierten Netzen ein möglichst gutes Komitee zusammenzustellen. Diese Klassenbildung läßt sich wiederum in die evolutionäre Suche einbringen, indem die Klassen nach dem Zeitpunkt ihres Zusammenlegens sortiert werden. Man kann eine Klasse als desto unabhängiger von den anderen Suchpunkten betrachten, je später sie mit einer anderen verschmolzen wurde. Bevorzugte Selektion findet nun auf zwei Ebenen statt: (I) innerhalb jeder Klasse bezüglich der Evidenz, (II) unterhalb der Klassen bezüglich der Unabhängigkeit und der Evidenz

Am Ende des integrierten Optimierungskonzeptes liegen mehrere Netze mit lokal maximaler Evidenz vor, die ein geeignetes Komitee bilden. Die Gewichtung der einzelnen Mitglieder kann adaptiv für jedes neue Muster berechnet werden mittels der Konfidenzwerte, die durch den Bayes'schen Ansatz geliefert werden.

Kapitel 5

Optimierung eines neuronalen Modells

Aus der Bias-Varianz Dekomposition des Komiteefehlers (Gleichung (3.3)) kann man zwei Folgerungen ziehen: Um ein möglichst gutes Komitee von Netzen zu erhalten, sollten zum einen die einzelnen Komiteemitglieder einen möglichst kleinen Generalisierungsfehler haben. Zweitens sollten die Netze möglichst breit um die Komiteelösung streuen. Die zweite Bedingung kann man erfüllen, in dem man möglichst unabhängige Netze als Komiteemitglieder auswählt. Diese Fragestellung, wie man solche Netze findet, ist Gegenstand des nächsten Kapitels. Die erste Bedingung, möglichst gute Netze zu finden, wird im folgenden behandelt.

Auf der Ebene der Modelloptimierung ist der grundlegende Aspekt, den es zu betrachten gilt, das Suchen nach Modellen mit einer hohen a posteriori Wahrscheinlichkeit, d.h. mit einem günstigen Gewichtsvektor und Hyperparametern bzw. mit einer günstigen Netzkomplexität. Anhand von Experimenten soll gezeigt werden, daß die Kombination von Bayes'schem Lernen und evolutionärer Suche bei gleichem Aufwand bessere Modelle liefert. Voraussetzung dafür ist die (negative) Korrelation zwischen Evidenz und Generalisierungsfehler, d.h. die Evidenz kann in diesem Fall als Leitfaden durch den Suchraum verwendet werden. Hierzu wird in Abschnitt 5.1 zuerst das Problem aus Abbildung 1.1 betrachtet. Durch das Trainieren vieler Modelle stelle ich einen empirischen Zusammenhang zwischen Zahl der versteckten Neuronen und Testfehler bzw. Evidenz her. Die Prozedur zur Optimierung der Evidenz kann dann hierzu ins Verhältnis gesetzt werden, d.h. es wird nachgewiesen, daß das globale Maximum der Evidenz in dem Sinne, wie das in Abbildung 4.2 illustriert wurde, gefunden wird. Weiterhin wird noch die Optimierungsprozedur für ein künstliches Klassifikationsproblem betrachtet, das ebenso auf einer Sinusfunktion basiert. Die eindimensionalen Probleme haben den Vorteil, daß man einerseits die Ausgabefunktion gut visualisieren kann (vgl. Abbildungen 4.9 und 4.10). Andererseits ist aber auch keine zusätzliche Abhängigkeit von der Eingabestruktur vorhanden, was die Optimierung entsprechend erleichtert.

In Abschnitt 5.2 verwende ich dann vier bekannte Probleme aus Benchmarksammlungen (Neal, 1998, Merz & Murphy, 1998, Prechelt, 1994). Bei den Experimenten wird vor allem die Optimierung der Eingabe betrachtet, d.h. es ist nachzuweisen, daß die Optimierungsprozedur auch hier ein Optimum findet. Dieses Problem ist wesentlich schwieriger, da es

nicht nur auf die richtige Zahl an Merkmalen ankommt, sondern vor allem auf die richtige Kombination. Die Zahl an Kombinationen steigt exponentiell an (vgl. auch Abbildung 4.5). Für verschiedene Merkmalskombinationen wird auch die optimale Größe der versteckten Schicht unterschiedlich sein.

Aus dem Beispiel in der Einleitung, Abbildung 1.4, wurde bereits deutlich, daß die Dimensionalität des Eingabevektors für die Generalisierungsfähigkeit des neuronalen Modells eine wichtige Rolle spielt. In diesem Kapitel wird dieser Aspekt anhand ausgewählter Beispiele konkretisiert und eine Methode vorgeschlagen, mittels statistischer Kriterien (Korrelationskoeffizienten und Mutual Information) die Eingabestruktur eines neuronalen Modells zu optimieren. Die Kriterien wurde zum einen aufgrund ihrer mathematischen Fundierung gewählt, zum anderem aber auch aufgrund ihrer Unabhängigkeit von dem Modelltyp oder Lernverfahren, wodurch die einfache Übertragung auf andere Ansätze gewährleistet ist. Andere konnektionistische Kriterien zur Optimierung der Topologie, wie z.B. Optimal Brain Surgeon, sind nur für neuronale Netze einsetzbar. Die Experimente sollen zeigen, daß die Methoden einerseits in einem integrierten Optimierungskonzept eingesetzt werden können, andererseits aber auch in gewissen Grenzen die Möglichkeit besteht, trotz der Monotonie der Informationsmaße die Eingabestruktur 'offline' vor dem Training der Netze zu optimieren. Für die Schilddrüsen-Klassifikation in diesem Kapitel und für die Prognose von Absatzzahlen in Kapitel 7 wird die Voroptimierung eingesetzt. Für beide Anwendungen steht eine Vielzahl an Merkmalen zur Verfügung, aber nicht die dafür erforderliche Menge an Daten. So gibt Silverman die Anzahl der Muster, die man benötigt, um eine multivariate Normalverteilung im Nullpunkt genau zu schätzen, bei 10 Dimensionen mit 842.000 an (Tabelle 5.1). Das neuronale Modell schätzt eine bedingte Wahrscheinlichkeit. Aus diesem Grund kann bei einer zunehmenden Zahl an Merkmalen die Leistung beständig abnehmen. Die Experimente werden dies belegen.

Tabelle 5.1: Zahl der Muster die man benötigt, um eine multivariate Normalverteilung im Nullpunkt bis auf einen (quadratischen) Fehler von 0.1 genau zu schätzen. Entnommen aus (Silverman, 1986).

Dimensionen	benötigte Muster	Dimensionen	benötigte Muster
1	4	6	2.790
2	19	7	10.700
3	67	8	43.700
4	223	9	187.000
5	768	10	842.000

Die durch die Voroptimierung gewonnene Information kann auch dazu verwendet werden, die Population von Netzen geeignet zu initialisieren, d.h. Suchpunkte günstiger im Parameterraum zu plazieren. Konkreter: Merkmale werden nach ihrer Wichtigkeit gemäß der statistischen Kriterien sortiert und unwichtigere Merkmale häufiger bereits zu Anfang entfernt. Die Suche konzentriert sich damit eher auf Bereiche, die 'guten' Merkmalskombinationen entsprechen.

Aufbauend auf diesen Optimierungsstufen kann man bereits Komitees von Netzen betrachten, wenn der Entwickler den Entwurfsprozeß für unterschiedliche Modelle mehrmals durchführt, siehe dazu etwa (Ragg & Gutjahr, 1997b).

Das Bayes'sche Verfahren und die Optimierung der Evidenz stößt dann an seine Grenzen, wenn die Anzahl der Datenpunkte sehr klein ist, die Muster aber mit starkem Rauschen behaftet sind, das anhand der spärlichen Datenlage nicht mehr richtig einzuschätzen ist. Das liegt darin begründet, daß in diesem Fall auch andere Erklärungen bzw. Modelle für die Daten gefunden werden können, die ebenso ihre Berechtigung haben. Abschließend werde ich in diesem Kapitel genau diese Grenzfälle betrachten.

Einstellung allgemeiner Parameter

Für alle im folgenden durchgeführten Experimente sind einige Parameter des Lernverfahrens und der evolutionären Strategie einzustellen.

Die Parameter des Lernverfahrens werden wie folgt eingestellt. Die Anpassung der Hyperparameter erfolgt für Regressionsprobleme alle 100 Epochen, für Klassifikationsprobleme alle 70 Epochen. Für die Gewichtsgruppen wird für Netze mit mehr als einer versteckten Schicht immer die Standard-Einteilung gewählt, d.h. die Gewichte zwischen zwei Schichten bilden jeweils eine Gruppe und die Biase eine eigene. Im Falle von Klassifikationsproblemen wähle ich meistens Netze mit einer versteckten Schicht und fasse alle Gewichte in einer Gruppe zusammen. Die Biase werden für Klassifikationsprobleme nicht regularisiert. Die Anfangsschrittweite von Rprop beträgt immer 10^{-3} und die maximale Schrittweite 0.1. Die Hyperparameter α_k werden mit 0.02 initialisiert und die Gewichte für die Initialisierung aus der entsprechenden Normalverteilung mit $\sigma \approx 7$ gezogen. β wird immer mit 1 initialisiert.

In einem Versuch werden maximal 100 Netze betrachtet von denen bei der evolutionären Suche im allgemeinen 90% die Eltern der nächsten Generation bilden. Dabei werden je nach Versuch zwischen 10 und 25 Generationen gerechnet. Der Bevorzugungsfaktor y für die Selektion der Eltern beträgt 1.2. Für die Mutationswahrscheinlichkeiten werden folgende Werte eingestellt: 0.2 für die Mutation von versteckten Neuronen, falls die Eingabe nicht optimiert wird, sonst 0.1 für beide Werte. Pro Mutationsschritt wird pro Netz maximal ein Eingabeneuron und ein verstecktes Neuron eingefügt oder gelöscht. Die Auswahl zwischen Einfügen und Löschen wird als gleich wahrscheinlich gewählt. Die Schranken für die Initialisierung der Zahl der Neuronen jedes Netzes zu Beginn der Evolution werden für jeden Datensatz gesondert angegeben. Die Varianz der Normalverteilung von Zufallszahlen, die benutzt wird, um die Werte des Gewichtsvektors und die Hyperparameter zu mutieren, beträgt ca. 10^{-4}.

Netze, die eine konstante Funktion berechnen, werden aus der Population entfernt. Der Grund dafür ist, wie in Abbildung 4.10a zu sehen, daß konstante Funktionen aufgrund ihrer nicht vorhandenen Varianz eine große Evidenz haben können. Solche Funktionen sind aber generell nicht von Interesse, da das bedeutet, daß der Datensatz nicht prognostizierbar ist. Diese Eigenschaft sollte mit einem geeigneten statistischen Test vorab überprüft werden, da die Modellierung mittels maschineller Lernverfahren dann dafür allgemein keinen Sinn macht. Solche Netze treten beispielsweise auf, wenn die Hyperparameter divergieren, weil die a priori Annahmen (zufällig) verletzt waren. Abbildung 2.9 illustrierte diesen Sachverhalt.

5.1 Die Evidenz als Optimierungskriterium

Da beim Bayes'schen Lernen die Korrelation der Evidenz eines Netzes mit der Generalisierungsleistung nicht perfekt ist, bieten Evolutionsstrategien einen eleganten Rahmen, um die lokale Optimierung der Netzparameter und die globale Suche nach guten Lösungen zu verzahnen. Modelle mit höherer Evidenz produzieren mehr Nachkommen, es besteht also ein Selektionsdruck hinsichtlich dieses Kriteriums. Die Gefahr, die Parameter wie bei der Kreuzvalidierung zu sehr an eine zusätzliche Datenmenge anzupassen, ist hier nicht gegeben, da die Evidenz der Modelle während des Lernens ausschließlich auf Trainingsdaten berechnet wird. In diesem Abschnitt sollen verschiedene Aspekte der evolutionären Suche anhand von Experimenten beleuchtet werden.

5.1.1 Beschreibung der Problemstellungen

Sinus-Regressionsproblem

Die grundlegenden Fragestellungen der evolutionären Optimierung der Evidenz sollen an dem Sinusproblem aus Abbildung 1.1a gezeigt werden. Insgesamt besteht der Datensatz aus 40 Datenpunkten, zu denen normalverteiltes Rauschen addiert wurde. Die Daten genügen der Vorschrift $\sin(x) + 0.2 \cdot rand(x)$. Ein Histogramm für den Rauschanteil $rand(x)$ ist in Abbildung 5.25b gezeigt. Der Generalisierungsfehler wird bezüglich des erzeugenden Prozesses berechnet. Als Maximaltopologie wird ein Netz mit 30 versteckten Neuronen, davon 20 in der ersten Schicht, gewählt.

Sinus-Klassifikationsproblem

Die Daten für dieses Klassifikationsproblem wurden generiert, indem zu einer Sinusfunktion gleichverteiltes Rauschen aus dem Intervall $[0, 1]$ addiert wurde (Abbildung 5.1).

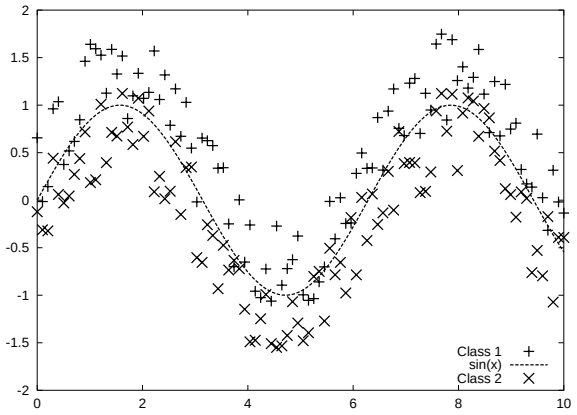


Abbildung 5.1: Die Abbildung zeigt zwei Klassen, die aus einer Sinusfunktion durch Addition von gleichverteiltem Rauschen generiert wurden. Die Mittelwerte beider Klassen bezüglich der y-Achse unterscheiden sich um 0.5, so daß im Grenzbereich eine deutliche Überlappung vorhanden ist. Für die Trainingsmenge wurden aus jeder Klasse jeweils 25 Muster zufällig gezogen, d.h. 50 Muster insgesamt. Die Testmenge besteht aus 150 Mustern.

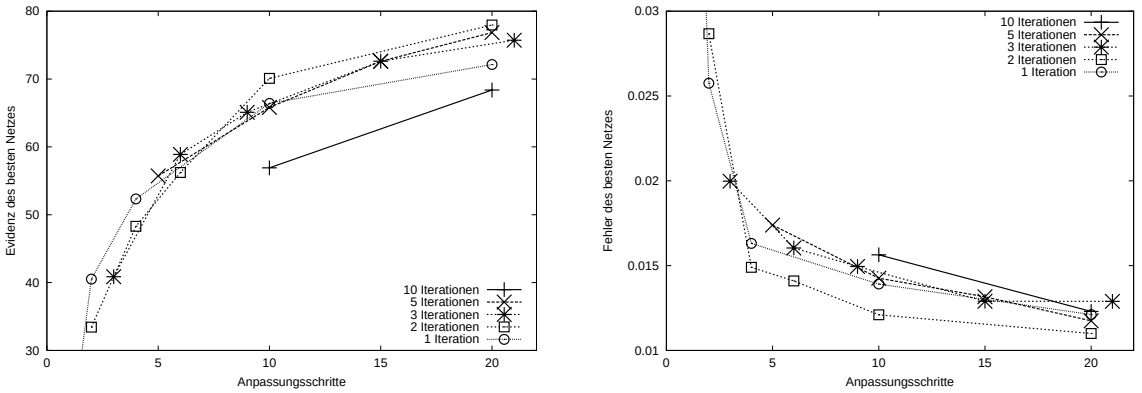
Für jede Klasse wurden 100 Punkte erzeugt, von denen je 25 zum Training und 75 zum Testen verwendet werden. Die erste Klasse genügt der Vorschrift $\sin(x) + rand(x) - 0.25$, und die zweite $\sin(x) + rand(x) - 0.75$. Im Grenzbereich kommt es zu Überlappungen, so daß ein

Modell des Datensatzes nie perfekt sein kann in dem Sinne, daß es keine Fehlklassifikationen macht. Dieser Datensatz ist deutlich schwieriger zu lernen als einige der realen Klassifikationsprobleme, die im nächsten Abschnitt betrachtet werden. Damit läßt sich vergleichen, inwieweit die Struktur erkannt wird oder nur die Grenzfälle auswendig gelernt werden.

5.1.2 Verzahnung von systematischem Suchen und Bayes'schem Lernen

In Kapitel 4.5 habe ich mit einer Plausibilitätsüberlegung begründet, warum es günstig ist, mindestens zwei Anpassungsschritte der Hyperparameter durchzuführen, bevor die Eltern auf Basis ihrer Evidenz selektiert und durch Mutation die Nachkommen für die nächste Generation erzeugt werden.

Anhand des Sinus-Regressionsproblems werde ich im folgenden vergleichen, ob und wie das Ergebnis von der Zahl der Iterationen des Bayes'schen Verfahrens pro Generation abhängt. Betrachtet werden 1,2,3,5 und 10 Anpassungsschritte pro Generation. Für jede Variante wurde der Algorithmus 20 mal durchgeführt. Die Zahl der Anpassungsschritte betrug mit einer Ausnahme[1] immer 20, d.h. der Aufwand ist für alle Verfahren gleich. Wird also ein Anpassungsschritt pro Generation gemacht, dann werden 20 Generationen insgesamt gerechnet, im anderen Extremfall, wenn 10 Anpassungsschritte gemacht werden, entsprechend nur 2 Generationen. Für jede Generation wird das beste Netz, d.h. das mit der größten Evidenz, ausgewählt und der Mittelwert über alle zwanzig Versuche gebildet. Die Abbildung 5.2 zeigt die Ergebnisse sowohl bezüglich des Testfehlers als auch bezüglich der Evidenz.



(a) (b)

Abbildung 5.2: Die Abbildung zeigt die Entwicklung der Evidenz und des Testfehlers in Abhängigkeit der Zahl der Anpassungsschritte für die Hyperparameter. Die fünf Linien entsprechen den fünf Varianten, Nachkommen zu generieren. Die fünf Strategien unterscheiden sich darin, wieviele Schritte pro Generation gerechnet werden, d.h. ob die Verzahnung von Suche und Bayes'schem Lernen stark oder eher schwach ist. Betrachtet werden 1,2,3,5 und 10 Schritte pro Generation. Die Werte sind über 20 Versuche gemittelt. Die Strategie, nach jedem zweiten Anpassungsschritt Nachkommen zu generieren, schneidet am besten ab. Die Ergebnisse sind signifikant (siehe auch Text). (a) Höchster Wert der Evidenz in jeder Generation für die fünf untersuchten Möglichkeiten der Verzahnung. (b) Entwicklung des Testfehlers für das Netz mit der höchsten Evidenz. Das heißt, in jeder Generation wird jeweils nur das fitteste Netz betrachtet.

Die Selektion und Mutation nach zwei Anpassungsschritten durchzuführen schneidet am besten ab. Vergleicht man die Endergebnisse des Testfehlers dieser Strategie mit den vier

[1] Bei drei Anpassungsschritten pro Generation werden insgesamt 21 Anpassungsschritte gemacht.

anderen, dann wird der t-Test auf Gleichheit der Mittelwerte bei einer Schwelle von $t_{0.95;20} = 1.73$ für alle abgelehnt und für den Fall von 5 Iterationen pro Generation angenommen. Für die Evidenz wird der Test für 3 und 5 Iterationen angenommen. Bei gleichem Aufwand erziehlt man also für 2 bis 5 Iterationen pro Generation die besten Ergebnisse. Je weniger Iterationen man allerdings pro Generation durchführt, desto breiter kann man den Raum durchsuchen. Im folgenden werden alle Experimente so durchgeführt, daß nach jeweils zwei Iterationsschritten des Bayes'schen Verfahrens Nachkommen generiert werden.

5.1.3 Maximum der Evidenz bezüglich der Netzgröße

Evolutionäre Algorithmen sollten in der Lage sein, die optimale Anzahl von versteckten Neuronen für ein gegebenes Problem zu finden. Gutjahr hat in seiner Dissertation für mehrere Probleme empirisch gezeigt, daß es ein Maximum der Evidenz in Abhängigkeit der Netzgröße gibt (Gutjahr, 1999). Das Maximum findet man, indem man über die Größe des Netzes iteriert und für jede Zahl an versteckten Neuronen mehrere Initialisierungen trainiert. Die Ausführungen in Kapitel 4 legten nahe, daß man dieses Maximum durch Einsatz eines Suchverfahrens schneller und mit weniger Aufwand finden sollte (vgl. Abbildung 4.2). Im folgenden will ich für das Sinus-Regressionsproblem diesen empirischen Zusammenhang herstellen und zeigen, daß die Suche nach einem Maximum der Evidenz erfolgreich ist. Im Gegensatz dazu schneidet die Auswahl des Netzes mit der höchsten Evidenz aus einer Menge von trainierten Netzen deutlich schlechter ab.

Abbildung 5.3 zeigt Netze, die standardmäßig mit Bayes'schem Lernen trainiert, sowie Netze, die evolutiv optimiert wurden. Die Modelle sind über ihrer Evidenz und ihrem Testfehler aufgetragen. Der Testfehler ist für alle evolutiv optimierten Netze klein (vgl. auch Abbildung 5.4d). Das Risiko, ein falsches Modell zu wählen, sinkt also erheblich. Dies ist umso wichtiger, da die Korrelation zwischen Testfehler mit $\rho = -0.35$ zwar negativ, aber nicht allzu deutlich ist, so daß man sich bei der Modellauswahl ausschließlich auf sie stützen könnte. Das liegt zum einen daran, daß der Datensatz einen starken Rauschanteil hat, zum anderen machen sich aber auch die fehlenden Daten im Intervall $[0, 2]$ bemerkbar (vgl. Abbildung 1.1), die die Aufgabe zusätzlich erschweren. Solange die Korrelation gegeben ist, findet die evolutive Suche immer gute Netze, da sich im Laufe der Evolution sehr viel mehr Netze gegeneinander bewähren müssen.

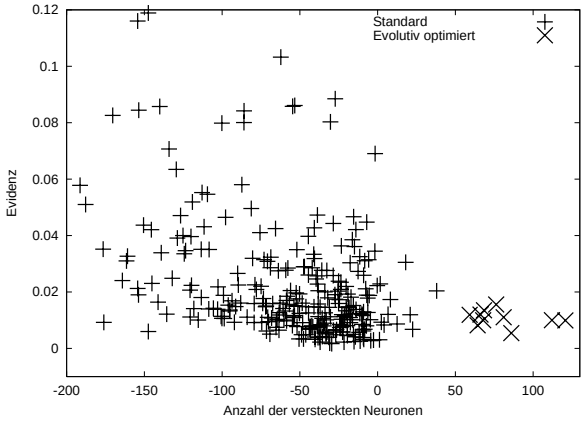


Abbildung 5.3: Vergleich der evolutiv optimierten Netze mit einer Auswahl an normal trainierten Modellen. Im rechten unteren Bereich finden sich die optimierten Netze, d.h. sie haben eine hohe Evidenz und einen kleinen Fehler. Dagegen finden sich bei den normal trainierten Netzen etliche mit einer hohen Evidenz aber großem Fehler. Das Risiko, ein falsches Modell auszuwählen, sinkt also durch die evolutive Optimierung erheblich.

Die Abbildungen 5.4a-d zeigen den empirischen Zusammenhang zwischen Netzgröße und Evidenz bzw. Generalisierungsfehler für das Sinus-Regressionsproblem. Insgesamt wurden in 10 Versuchen je 100 Netze, d.h. insgesamt 1000, mit unterschiedlichen Topologien trainiert. Die maximale Topologie war 1-20-10-1 und die minimale 1-4-4-1, d.h. ein Netz hat mindestens vier Neuronen pro versteckter Schicht. Die beiden oberen Abbildungen zeigen den Verlauf der Evidenz. Im Bereich zwischen 16 und 24 Neuronen zeichnet sich ein deutliches Maximum ab. In der rechten oberen Abbildung ist zu erkennen, daß das evolutionäre Verfahren das Maximum findet. Alle Netze gruppieren sich in diesem Bereich. Die evolutiv optimierten Modelle erreichen einen signifikant niedrigeren Testfehler als die Netze mit der höchsten Evidenz der zehn Versuche. Tabelle 5.2 stellt die Ergebnisse für den Testfehler zusammen.

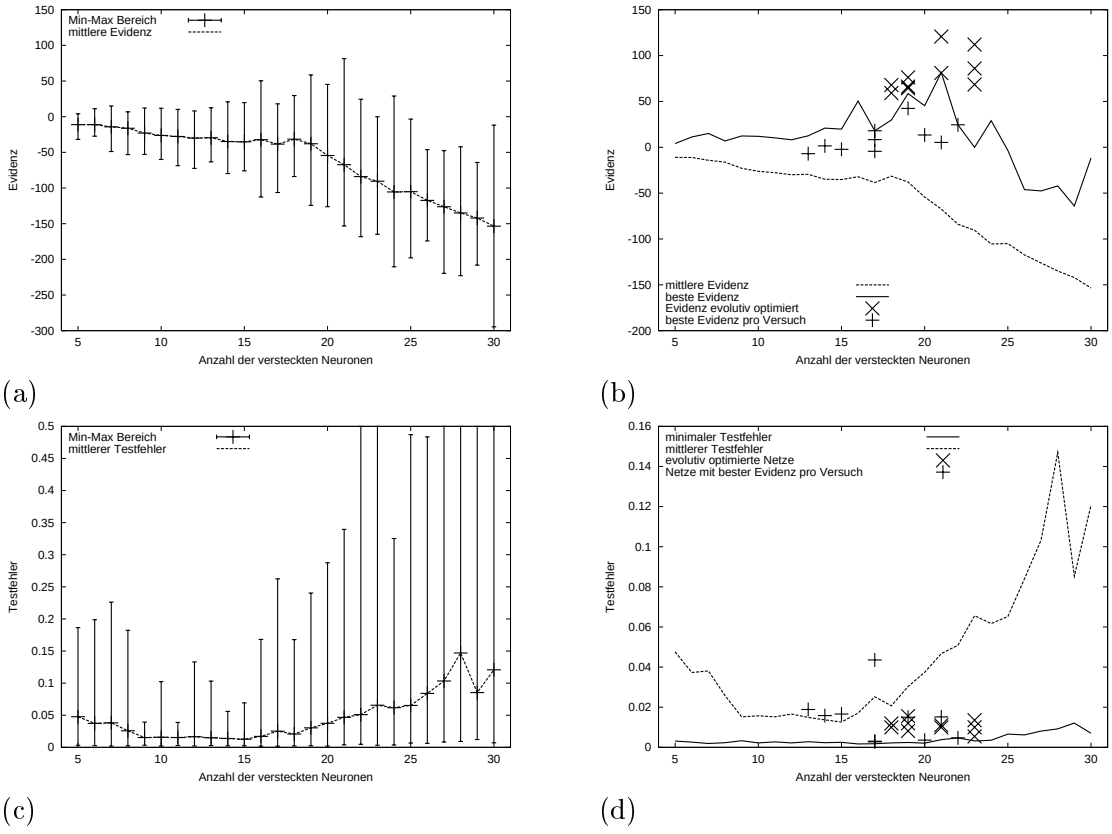


Abbildung 5.4: Die Abbildung zeigt die empirische Abhängigkeit der Evidenz bzw. des Testfehlers von der Zahl an versteckten Neuronen für das Sinus-Regressionsproblem. Um den Zusammenhang zu ermitteln, wurden insgesamt 1000 Netze trainiert. a) Mittlere, minimale und maximale Evidenz. Die Netze mit der größten Evidenz haben zwischen 16 und 24 Neuronen in der versteckten Schicht. b) Bereich der mittleren und maximalen Evidenz. Die Kreuze (+) entsprechen jeweils dem Netz mit der höchsten Evidenz pro Versuch. Die anderen Symbole (×) kennzeichnen die Ergebnisse der evolutiven Optimierung. Sie konzentrieren sich alle im Bereich des Maximums. Die absoluten Werte liegen sogar etwas höher. c) Mittlerer, minimaler und maximaler Testfehler. Der mittlere Testfehler steigt ab ca. 15 Neuronen wieder an, der minimale Fehler etwa ab 23 Neuronen. d) Bereich des mittleren und minimalen Testfehlers. Die Symbole sind genauso verwendet wie in (b). Die evolutive Optimierung erreicht einen mittleren Fehler von 0.011. Die Netze mit der höchsten Evidenz aus den 10 Versuchen haben im Mittel einen Fehler von 0.014. Der Unterschied ist signifikant bei einer Schwelle von $t_{0.95;10} = 1.82$.

Tabelle 5.2: Mittlerer, minimaler und maximaler Generalisierungsfehler für Netze, die wie folgt ausgewählt wurden: (I) 100 Initialisierungen mit Bayes'schem Lernen. (II) Netze mit höchster Evidenz aus 10 Versuchen. (III) Netze, die aus 10 Versuchen mit evolutiver Optimierung gewonnen wurden. Der t-Test zeigt, daß der Unterschied signifikant ist (Schwelle $t_{0.95;10} = 1.82$).

Methode	Mittelwert	σ	Minimum	Maximum
(I)	0.0429	0.08	0.0017	1.049
(II)	0.0139	0.012	0.0027	0.044
(III)	0.011	0.003	0.0054	0.016

Tabelle 5.3: Mittlere, minimale und maximale Trefferquote für Netze, die wie folgt ausgewählt wurden: (I) 100 Initialisierungen mit Bayes'schem Lernen. (II) Netze mit höchster Evidenz aus 10 Versuchen. (III) Netze die aus 10 Versuchen mit evolutiver Optimierung gewonnen wurden. Der t-Test zeigt, daß der Unterschied signifikant ist (Schwelle $t_{0.95;10} = 1.82$). Zum Vergleich: Ein lineares Modell kommt auf eine Trefferquote von 64.6%.

Methode	Mittelwert	σ	Maximum	Minimum
(I)	76.4%	2.2	80.7%	66.0%
(II)	78.2%	0.9	79.3%	76.7%
(III)	78.9%	1.2	81.3	77.3%

Bemerkenswert ist weiterhin, daß die mittlere Evidenz beständig abnimmt, ab etwa 17 versteckten Neuronen auch sehr deutlich. Das bedeutet, wenn man nur wenige Netze mit dem Bayes'schen Verfahren trainiert (und keine evolutive Optimierung anwendet), dann ist es günstiger, weniger Neuronen zu verwenden. Dieser Umstand wird in Kapitel 7 nochmals angesprochen.

Für das Klassifikationsproblem ist der Zusammenhang unübersichtlicher (Abbildungen 5.5a-d). Auch hier wurden 1000 Netze trainiert. Die maximale Topologie war 2-15-10-1 und die minimale 2-4-4-1. Hier zeigen die beiden oberen Abbildungen, daß es mehrere Maxima der Evidenz gibt. Die evolutiv optimierten Netze liegen jetzt auch verstreut über dem ganzen Bereich, d.h. es werden verschiedene Maxima gefunden. Die Evidenz der optimierten Modelle fällt wieder höher aus als die der konventionell trainierten. Die Leistung auf den Testdaten ist wieder signifikant besser. Sie steigt im Mittel von 78.2% (Netze mit höchster Evidenz) auf 78.9% für die evolutiv optimierten Netze. Tabelle 5.3 stellt die Ergebnisse zusammen.

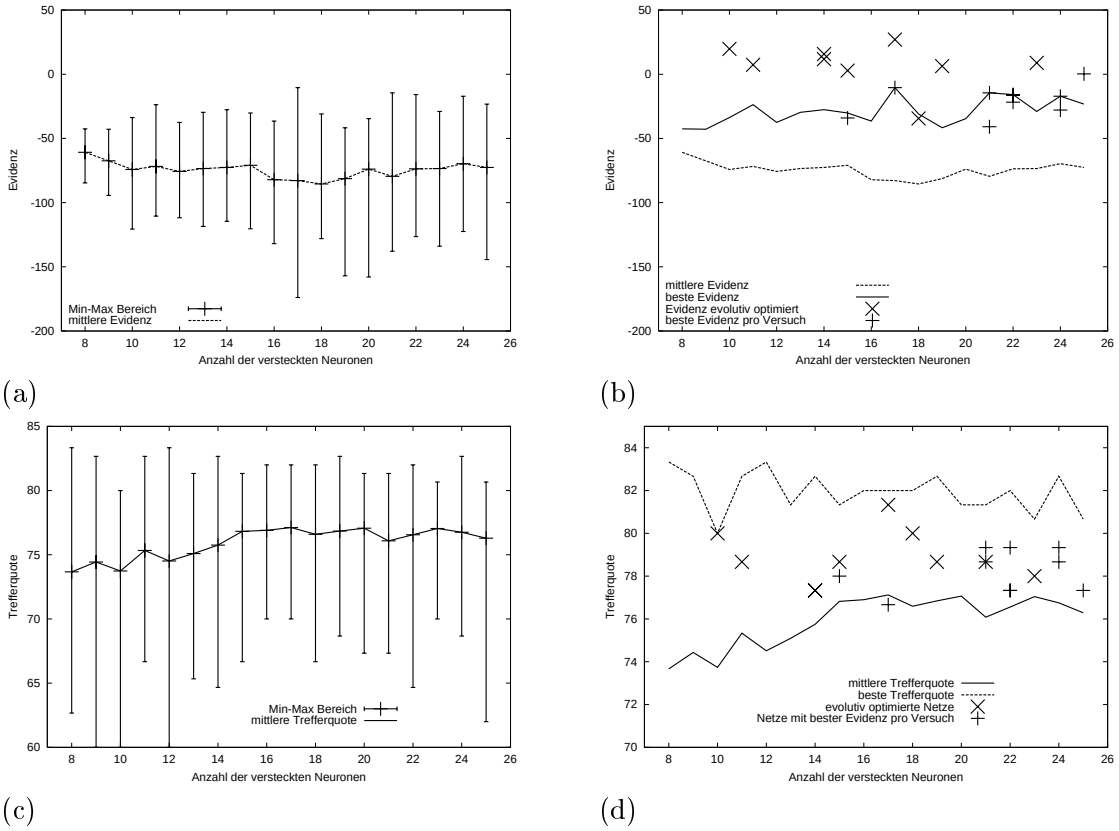
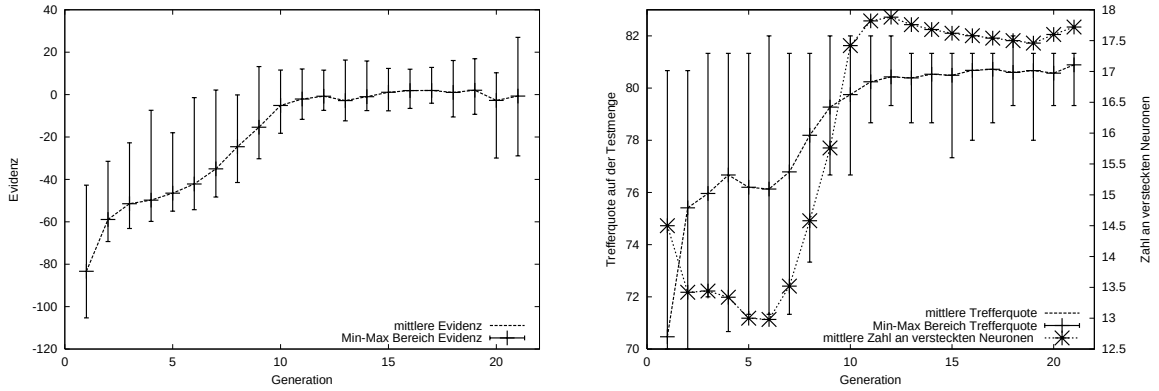


Abbildung 5.5: Die Abbildung zeigt die empirische Abhängigkeit der Evidenz bzw. der Trefferquote auf der Testmenge von der Zahl an versteckten Neuronen für das Sinus-Klassifikationsproblem. Um den Zusammenhang zu ermitteln, wurden insgesamt 1000 Netze trainiert. Die Trefferquote wird auf einer Testmenge gemessen und nicht bezüglich des erzeugenden Prozesses, wie das bei der Regressionsaufgabe der Fall war. Bei diesem Beispiel ist die Leistung auf der Testmenge also auch eine mit Rauschen behaftete Größe. a) Mittlere, minimale und maximale Evidenz. Es sind mehrere Maxima der Evidenz zu beobachten. Der höchste Wert liegt bei 17 Neuronen. b) Bereich der mittleren und maximalen Evidenz. Die Kreuze (+) entsprechen jeweils dem Netz mit der höchsten Evidenz pro Versuch. Die anderen Symbole (×) kennzeichnen die Ergebnisse der evolutiven Optimierung. Im Gegensatz zum Regressionsproblem werden bei diesem Datensatz mehrere Optima gefunden. Die absoluten Werte der evolutiv gefundenen Netze liegen wieder deutlich über denen der standardmäßig trainierten Netze. c) Mittlere, minimale und maximale Trefferquote auf der Testmenge. Wie bei der Evidenz gibt es auch hier mehrere Maxima. d) Bereich der mittleren und maximalen Trefferquote auf den Testdaten. Die Symbole sind genauso verwendet wie in (b). Die evolutive Optimierung erreicht eine mittlere Trefferquote von 78.9%. Die Netze mit der höchsten Evidenz aus den 10 Versuchen haben im Mittel eine Leistung von 78.2%. Der Unterschied ist signifikant bezüglich des t-Tests bei einer Schwelle von $t_{0.95;10} = 1.82$.

5.1.4 Verlauf der Evolution

Nachdem wir gesehen haben, daß die Evolution Maxima der Evidenz im Parameterraum findet, ist es interessant, die Entwicklung einmal für ein Beispiel zu betrachten. Abbildung 5.6 zeigt für das Sinus-Klassifikationsproblem den Verlauf der Mittelwerte für Evidenz, Trefferquote und Zahl der versteckten Neuronen in Abhängigkeit der Generationen. Zu Beginn liegen die Werte für die Evidenz noch deutlich niedriger und streuen stärker, bis sie sich dann zunehmend um das Maximum einpendeln, das nach etwa zehnten Generationen bereits erreicht ist (Abbildung 5.6a). In der rechten Abbildung erkennt man, wie sich die Trefferquote auch nach etwa 10 Generationen auf hohem Nievau stabilisiert. Interessant ist die Entwicklung für die mittlere Anzahl an versteckten Neuronen. Am Anfang sinkt der Wert zuerst ab, bis dann nach sechs Generationen ein Minumum erreicht ist. Dann steigt die Zahl kontinuierlich an, bis auf einen Wert zwischen 17 und 18 Neuronen. Dort ist offenbar ein Maximum der Evidenz erreicht, das im folgenden dann auch nicht mehr verlassen wird.



(a) (b)

Abbildung 5.6: Die Abbildung zeigt die Entwicklung der Evidenz und der Trefferquote in Abhängigkeit der Zahl der Generationen. (a) Mittlere, minimale und maximale Evidenz. Bis zur zehnten Generation nimmt die Evidenz deutlich zu und pendelt dann um das Maximum. (b) Entwicklung des Trefferquote auf der Testmenge und der mittleren Netzkomplexität, d.h. der Zahl an versteckten Neuronen. Die mittlere Leistung der Netze wird zunehmend größer. Auch hier pendelt die Leistung ab der 10 Generation um eine feste Größe. Erstaunlich ist, daß die Zahl an versteckten Neuronen zuerst abnimmt, ab der sechsten Generation sprunghaft zunimmt und sich dann auf einen Wert zwischen 17 und 18 Neuronen einpendelt.

5.2 Evolutive Merkmalselektion und Topologieoptimierung

Aus der Statistik sind verschiedene Methoden bekannt, die den Infomationsgehalt einer Kodierung berechnen. In aller Regel sind diese Maße monoton wachsend, d.h. je größer der Merkmalsvektor, desto höher der Informationsgehalt der Eingabe bezüglich der Ausgabe (Fukunaga, 1990, Bishop, 1995). Gleichzeitig stehen aber pro Dimension, die hinzugefügt wird, immer weniger Datenpunkte zur Verfügung, um die Parameter zu bestimmen. In niedrigdimensionalen Räumen läßt sich deswegen mit weniger Information die gesuchte Funktion besser approximieren (Silverman, 1986, White, 1989, Bishop, 1995). Im folgenden werde ich darlegen, daß man mit diesen Kriterien in gewissen Schranken eine Vorverarbeitung durchführen und damit den Suchraum für die evolutionäre Suche sinnvoll eingrenzen kann.

Dazu werde ich für jeden Datensatz eine Top-Down Suche, wie in Abbildung 4.5 illustriert, durchführen.

5.2.1 Iterative Merkmalselektion

Der Korrelationskoeffizient ist die einfachste statistische Kenngröße für Paare von Zufallsvariablen. Sind zwei Funktionen bzw. zwei Merkmale stark korreliert, dann wird man bei der Modellbildung auf eines der beiden verzichten können. Eine einfache Rückwärtssuche kann bei manchen praktischen Problemen die Zahl der Eingabemerkmale unter Umständen bereits deutlich reduzieren. Zumindest erhält man aber eine Aussage darüber, wie groß der lineare Anteil in der Problemstellung ist.

Algorithmus: *Rückwärtssuche mit Korrelationskoeffizienten*

1. Berechne paarweise für alle Eingabeneuronen ihren Korrelationskoeffizienten, sowie für alle Eingabeneuronen und das Ausgabeneuron.
2. Bestimme das Paar von Eingabeneuronen mit der größten Korrelation und lösche das Neuron der beiden, das die geringere Korrelation zur Ausgabe besitzt.
3. Wiederhole, bis Abbruchkriterium erfüllt ist, z.B. ein Schwellwert für die Korrelation.

Analog dazu kann man ebenso mit dem Nullvektor beginnen und solange Neuronen einfügen, bis eine bestimmte Korrelation überschritten ist (Vorwärtssuche).

Neben der Einschränkung, daß der Korrelationskoeffizient nur lineare Zusammenhänge erkennt, hat der angegebene Algorithmus auch den Nachteil, daß damit nur eindimensionale Zufallsvariablen paarweise verglichen werden können. Der nun folgende Algorithmus basiert auf dem Mutual Information Kriterium aus Kapitel 3.6.2. Dieses ist für Zufallsvariablen beliebiger Dimensionalität definiert. Ausgehend von der maximalen Eingabestruktur werden fortlaufend Komponenten gelöscht. Dabei wird jeweils der lokal optimale Schritt ausgeführt, d.h. das Merkmal weggelassen, das den geringsten Informationsverlust verursacht.

Algorithmus: *Rückwärtssuche mit Mutual Information*

1. Berechne paarweise für alle Eingabevektoren, die aus dem aktuellen Vektor durch Weglassen einer Komponente gebildet werden können, den Informationsgehalt bezüglich der Ausgabe.
2. Bestimme den Vektor mit dem höchsten Informationsgehalt und lösche das entsprechende Neuron.
3. Wiederhole, bis Abbruchkriterium erfüllt ist, z.B. ein Schwellwert für den Informationsgehalt oder dessen prozentuale Veränderung.

Auch hier kann man ein analoges Verfahren angeben, das ausgehend vom Nullvektor kontinuierlich Merkmale einfügt. Die angegebenen Algorithmen kann man als Grenzfälle einer

evolutionären Strategie mit einer einelementigen Population auffassen. Die Mutationswahrscheinlichkeit für das Löschen resp. Einfügen beträgt dann 1. Es liegt nahe, daß man für das kombinatorische Suchproblem am besten eine Mischung aus Vorwärts- und Rückwärtssuche verwendet, da die lokal optimalen Schritte nicht global optimal sein müssen. Verwendet man mehrere Suchpunkte gleichzeitig, dann erhält man ein klassisches evolutionäres Suchverfahren.

In der Praxis gehe ich so vor, daß ich bereits vor dem Training der Modelle starke lineare Abhängigkeiten in der Eingabestruktur und Merkmale, die keine Information tragen, beseitige. Hat der verbleibende Vektor eine Dimensionalität, die weit über die in Tabelle 5.1 empfohlene hinausgeht, dann werden weitere Merkmale basierend auf der Rückwärtssuche mit Mutual Information entfernt. Als Schranke für die Zahl der Merkmale wähle ich üblicherweise einen Wert, der vier oder fünf Merkmale zuviel beinhaltet. Das hat den Zweck, der evolutionären Suche genügend Spielraum einzuräumen. Die letztendliche Optimierung der Eingabe wird dann verzahnt mit der Parameteroptimierung durchgeführt (Ragg & Gutjahr, 1997a, Ragg & Gutjahr, 1997b). Daß die Verzahnung sinnvoll und in der Praxis erfolgreicher ist, wird an den Beispielen deutlich werden. Ein Grund dafür liegt darin, daß die Kriterien mit zunehmender Dimensionalität des Eingabevektors monoton wachsen. Das heißt, man kann die möglichen Eingabevektoren nur partiell anhand des Kriteriums anordnen. Das Wissen, daß Vektor $\mathbf{x}$ einen höheren Informationsgehalt hat als Vektor $\mathbf{y}$ ist für die Ordnung wertlos, wenn $\mathbf{x}$ ebenfalls mehr Komponenten hat.

Die Ergebnisse der Rückwärtssuche nutze ich weiterhin dazu aus, die neuronalen Netze geeignet zu initialisieren. Die Reihenfolge, in der die Merkmale gelöscht wurden, wird dazu verwendet, diese in eine Liste zu sortieren. Das Merkmal, das zuerst gelöscht wurde, steht am Anfang der Liste, usw. Auf diese Liste kann man nun wieder eine Selektionsfunktion aus Abbildung 4.3 anwenden. Dadurch erreicht man, daß vermeintlich wichtigere Merkmale mit höherer Wahrscheinlichkeit verwendet werden. Anhand von Experimenten wird der Nutzen dieser Initialisierung nachgewiesen.

In den nun folgenden Experimenten, werde ich zuerst den Datensatz beschreiben und dann die Ergebnisse der obigen Algorithmen präsentieren. Danach werden wieder verschiedene Aspekte der evolutionären Suche betrachtet. Im ersten Abschnitt war die optimale Anzahl der versteckten Neuronen zu bestimmen. Hier ist die Suche schwieriger, da nun sowohl die Eingabestruktur als auch die versteckten Schichten verändert werden.

5.2.2 Beschreibung und Voruntersuchung der Problemstellungen

Insgesamt werden in diesem Abschnitt vier Probleme betrachtet, drei Klassifikationsprobleme aus der UCI-Benchmarksammlung (Merz & Murphy, 1998, Prechelt, 1994) und ein Regressionsproblem aus der DELVE-Benchmarksammlung (Neal, 1998). Die ersten beiden Datensätze, Diabetes und Brustkrebs, gehören zu den kleineren der oft verwendeten Klassifikationsprobleme, d.h. es liegen wenige Muster vor (< 1000) und die Zahl der Merkmale ist nicht zu groß (< 10). Das Schilddrüsenproblem hat eine 1-aus-N Kodierung, viele Merkmale (> 20), aber auch weit über 1000 Muster.

Das Add10-Problem ist ein künstliches Regressionsproblem, bei dem die optimale Eingabe bekannt ist, genügend Daten vorliegen und die Nicht-Linearität gesichert ist. Tabelle 5.4

zeigt die Struktur der Datensätze in einer Übersicht.

Tabelle 5.4: Die Tabelle faßt die Strukturen der verwendeten Probleme zusammen. Bei den beiden letzteren werden 10% der Daten zum Training genutzt und auf dem Rest getestet. Damit sind aussagekräftige Ergebnisse gewährleistet. In den ersten beiden Fällen wurden die Vorgaben der Literatur beibehalten, um auch hier Vergleiche zu ermöglichen. Die letzte Spalte zeigt, wie gut man mit einer konstanten Ausgabe bereits werden kann, wenn man einfach immer die Klasse ausgibt, die am häufigsten auftritt.

Datensatz	Training	Test	Merkmale	Ausgaben	Muster pro Klasse
Diabetes	576	192	8	1	34,9% / 65,1%
Brustkrebs	525	174	9	1	34,5% / 65,5%
Schilddrüsen	720	6480	21	3	2,3% / 5,1% / 92,6%
Add10	500	4500	10	1	—

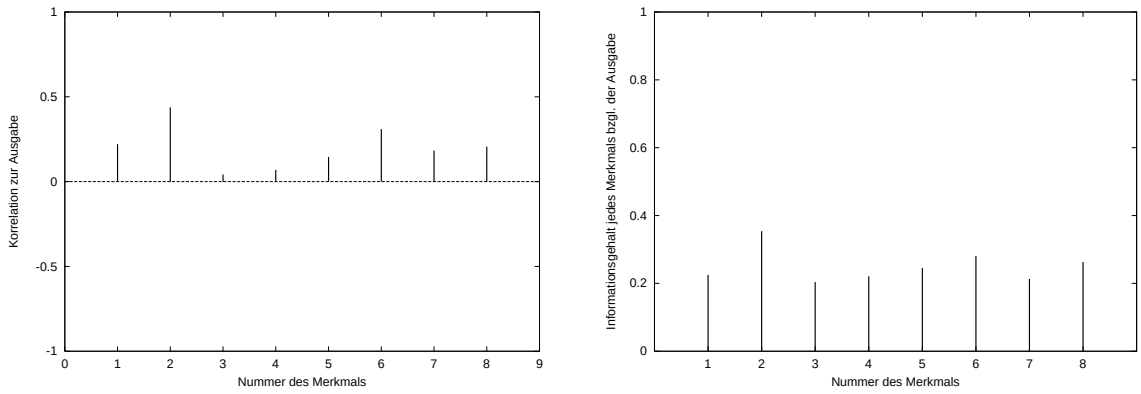
Für die Klassifikationsprobleme wurden die Eingabedaten standardisiert, indem für jedes Merkmal der Mittelwert abgezogen und durch die Standardabweichung geteilt wurde. Die Größe dieser Werte wurde auf den Trainingsdaten ermittelt. Das Add10-Problem wurde unverändert aus (Gutjahr, 1999) übernommen. Ich werde für die Ergebnisse bei Klassifikationsproblemen immer die Trefferquote angeben. Da beim Schilddrüsenproblem eine konstante Ausgabe bereits eine Leistung von über 92% hat, ist es unerläßlich, die Veränderungen zu vergleichen. Das heißt eine Steigerung von 98% auf 99% entspricht einer Reduktion der Fehlklassifikationen um die Hälfte.

Klassifikation von Diabetes

Dieses Klassifikationsproblem stammt von Vincent Sigillito von der Johns Hopkins University und ist über die Datenbank der University of Irvine erhältlich (Merz & Murphy, 1998). Die Aufgabe besteht darin, aus persönlichen Daten und medizinischen Meßwerten zu entscheiden, ob Frauen (indianischer Abstammung) an Diabetes leiden. Erfaßt wird zum Beispiel die Anzahl der Schwangerschaften, das Alter, Blutdruck oder Gewicht.

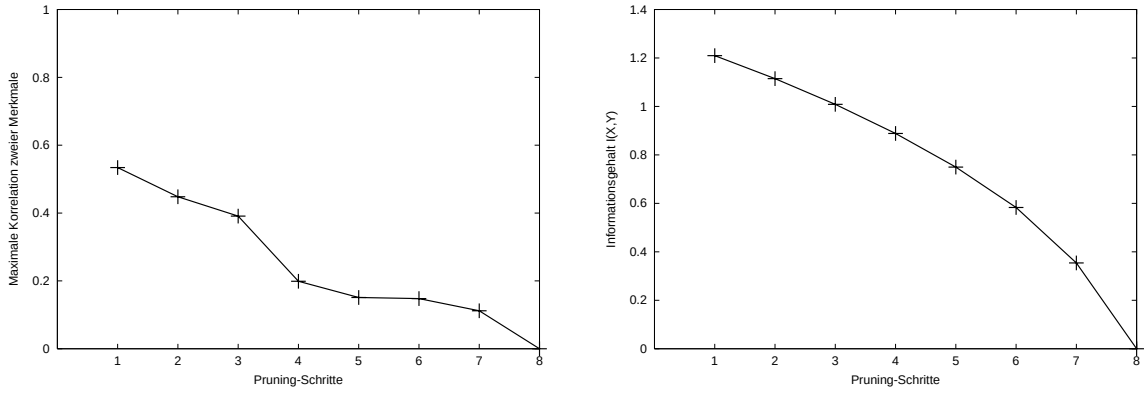
Im ersten Schritt untersuche ich für alle Probleme die linearen und nicht-linearen Abhängigkeiten der Ausgabe von den einzelnen Merkmalen. Diese Werte werden auf den gesamten Trainingsdaten berechnet. Abbildung 5.7a zeigt für zwei Merkmale eine mittelstarke Korrelation mit der Ausgabe. Das sind Merkmal 2 (Wert eines oralen Blutzuckertests) und Merkmal 6 (Körpergewicht). Im rechten Bild ist zu erkennen, daß der Informationsgehalt weniger stark schwankt. Die Aufgabenstellung hat also eine nicht-lineare Komponente.

Die Merkmale für dieses Problem sind relativ unabhängig voneinander. Abbildung 5.8a zeigt, daß bei der Merkmalselektion die maximale Korrelation zweier Merkmale schnell abnimmt und nach drei Schritten auf einen Wert von 0.2 fällt. Im rechten Teil der Abbildung 5.8 ist der Informationsgehalt der Eingabevektoren aufgetragen, die entstehen, wenn man die Rückwärtssuche mit Mutual Information anwendet. Ein gleichmäßiges Absinken deutet darauf hin, daß die Merkmale unabhängig sind und alle wichtige Information beitragen.



(a) (b)

Abbildung 5.7: Linearer und nicht-linearer Zusammenhang zwischen den einzelnen Merkmalen X_i und der Ausgabe Y für das Diabetes-Klassifikationsproblem. (a) Korrelation $\rho(X_i, Y)$. Einige Merkmale zeigen deutlich einen linearen Zusammenhang mit der Ausgabe, während andere nahezu unkorreliert sind. (b) Informationsgehalt $I(X_i, Y)$. Gegenüber der linken Abbildung ist zu erkennen, daß die Merkmale 3,4 und 5 einen nicht-linearen Zusammenhang mit der Ausgabe aufweisen.



(a) (b)

Abbildung 5.8: Anwendung der Rückwärtssuche mit Korrelationskoeffizienten und Mutual Information auf das Diabetes-Klassifikationsproblem. (a) Maximale Korrelation zweier Merkmale in jedem Pruningschritt: Die Korrelationen sind relativ niedrig und sinken nach drei Schritten deutlich ab. Pro Schritt wird für das entfernte Merkmal die entsprechende Zeile und Spalte der Korrelationsmatrix gestrichen. (b) Informationsgehalt $I(X, Y)$ des Eingabevektors nach jedem Pruning-Schritt: Pro Schritt wird ein Merkmal entfernt. Der Informationsgehalt des Eingabevektors sinkt relativ gleichmäßig. Die Reihenfolge der Merkmale beim Pruning ist: 5, 3, 4, 1, 7, 8, 6, 2.

Vergleichswerte aus der Literatur für dieses Klassifikationsproblem sind in Tabelle 5.5 zusammengestellt. Auf dem wie oben angegeben standardisierten Datensatz erreicht ein lineares neuronales Netz bereits 79.1%! Damit liegt man bereits über dem besten in der Literatur verzeichneten Wert. Man muß erwähnen, daß die beiden Klassen sich im Eingaberaum stark überlappen. Es ist schwierig, die Komplexität so einzustellen, daß die nicht-lineare Information ausgenutzt wird. Wie wir sehen werden, muß man die geeignete Merkmalskombination finden. Nach Tabelle 5.1 sollte man nicht mehr als 4 oder 5 Merkmale benutzen. Ohne ein automatisches Verfahren zur Gewichtung des Regularisierungsterms ist es unmöglich, die

Komplexität des Modells richtig zu bestimmen, da man das bei heuristischer Einstellung für viele Merkmalskombinationen und Topologien tun müßte.

Tabelle 5.5: Die Tabelle stellt einige Ergebnisse aus der Literatur für das Diabetes-Problem zusammen.

Referenz	Methode	Modelltyp	Leistung
(Breiman, 1999)	Bagging	Entscheidungsbaum	76.1%
	Boosting	Entscheidungsbaum	73.4%
(Drucker, 1999)	-	Neuronales Netz	75.1%
	Boosting	Neuronales Netz	78.3%
(Freund & Schapire, 1996)	Boosting	Entscheidungsbaum	75.6%

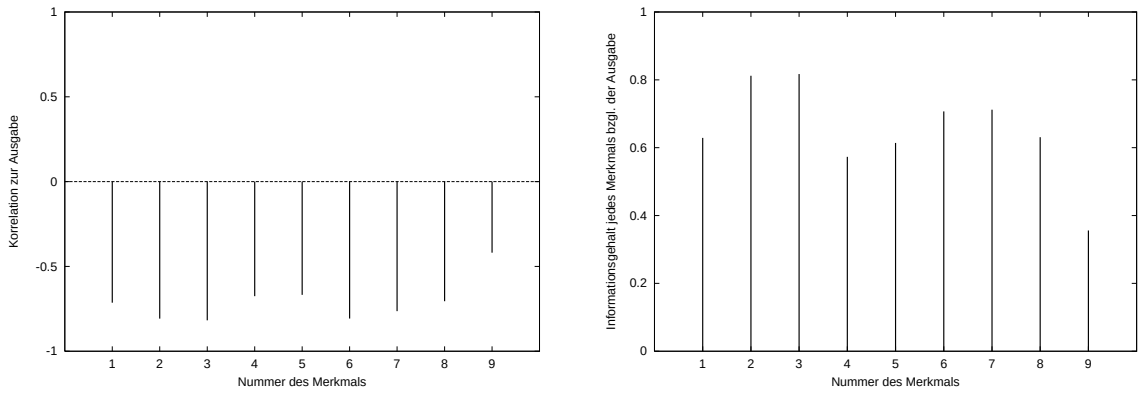
Klassifikation von Brustkrebs

Dieses Klassifikationsproblem stammt von dem Mediziner Holberg von der University of Wisconsin (Mangasarian & W.Wolberg, 1990) und wurde in der Datenbank der University of Irvine veröffentlicht (Merz & Murphy, 1998). Dort findet sich auch eine genauere Beschreibung des Datensatzes. Das Problem besteht darin, einen Tumor als gutartig oder bösartig zu klassifizieren. Die Eingabeinformation besteht aus beschreibenden Merkmalen der Zellen, z.B. die Verteilung der Größe bzw. Form, oder biochemische Informationen. Die Daten sind diskret auf einer Skala von 1 bis 10 erfaßt, was dieses Problem von allen anderen unterscheidet.

Abbildung 5.9 zeigt für drei Merkmale eine starke negative Korrelation mit der Ausgabe ($\rho > 0.8$). Auch die anderen sechs Merkmale weisen deutliche Korrelationen auf. Interessant ist in der rechten Abbildung, daß der Informationsgehalt der einzelnen Merkmale sich ganz ähnlich verhält. Die Werte sind aber zum Teil etwas geringer.

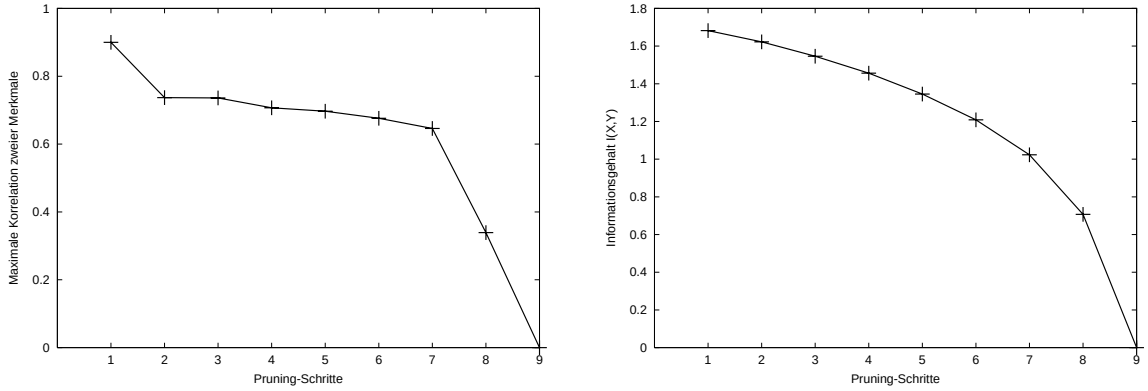
Bei diesem Datensatz sind die Merkmale stark miteinander korreliert. Abbildung 5.10a zeigt, daß bei der Merkmalselektion die maximale Korrelation zweier Merkmale kaum abnimmt. Solange noch drei Merkmale vorhanden sind, weisen zwei davon noch eine Korrelation von ca. 0.65 auf, die erst danach stark abfällt. Im rechten Teil der Abbildung 5.8 ist wieder der Informationsgehalt der Eingabevektoren aufgetragen, die entstehen, wenn man die Rückwärtssuche mit Mutual Information anwendet. Im Vergleich zum Diabetes-Datensatz sinkt der Informationsgehalt am Anfang deutlich langsamer ab.

Vergleichswerte aus der Literatur für dieses Klassifikationsproblem sind in Tabelle 5.6 zusammengestellt. Für diesen Benchmark sind auch Werte angegeben, die wir in früheren Studien mit ENZO erreicht haben. In der zweiten angegebenen Studie wurde die Merkmalsselektion basierend auf Mutual Information durchgeführt. Auf dem wie oben angegeben standardisierten Datensatz erreicht ein lineares neuronales Netz bereits 97.7%. Auch hier gilt, daß man die geeignete Merkmalskombination bestehend aus 4 bis 5 Merkmalen finden muß. Allerdings



(a) (b)

Abbildung 5.9: Linearer und nicht-linearer Zusammenhang zwischen den einzelnen Merkmalen X_i und der Ausgabe Y für das Krebs-Klassifikationsproblem. (a) Korrelation $\rho(X_i, Y)$. Alle Merkmale zeigen einen starken linearen Zusammenhang mit der Ausgabe. (b) Informationsgehalt $I(X_i, Y)$. Die Werte für Mutual Information entsprechen nahezu den Korrelationen, d.h. jede Eingabe hängt nur linear mit der Ausgabe zusammen.



(a) (b)

Abbildung 5.10: Anwendung der Rückwärtssuche mit Korrelationskoeffizienten und Mutual Information auf das Krebs-Klassifikationsproblem. (a) Maximale Korrelation zweier Merkmale in jedem Pruningschritt: Wenn noch drei Merkmale vorhanden sind (im 7. Schritt), liegt die maximale Korrelation noch bei ca. 0.7 und sinkt erst dann deutlich ab. (b) Informationsgehalt $I(X, Y)$ des Eingabevektors nach jedem Pruning-Schritt: Im Vergleich zum Diabetes-Klassifikationsproblem sinkt der Informationsgehalt zuerst deutlich langsamer. Das deutet darauf hin, daß für dieses Problem der optimale Eingabevektor kleiner ist als für das Diabetes-Klassifikationsproblem. Die Reihenfolge der Merkmale beim Pruning ist: 9, 3, 2, 5, 8, 7, 4, 1, 6.

erreicht man bei diesem Problem auch mit großen Netzen gute Werte. Dies hängt mit der ganz speziellen Struktur der Eingabedaten zusammen und läßt sich nicht verallgemeinern. Auf die Eigenheit des Datensatzes wird später noch eingegangen.

Das Add10-Regressionsproblem

Das Add10-Problem ist ein künstlich erzeugter Datensatz, der von Friedmann vorgeschlagen wurde (Friedman, 1988) und Teil der DELVE-Benchmarksammlung ist (Neal, 1998). Er

Tabelle 5.6: Die Tabelle stellt einige Ergebnisse aus der Literatur für das Krebs-Klassifikationsproblem zusammen.

Referenz	Methode	Modelltyp	Leistung
(Breiman, 1999)	Bagging	Entscheidungsbaum	96.3%
	Boosting	Entscheidungsbaum	96.8%
(Drucker, 1999)	-	Neuronales Netz	96.99%
	Boosting	Neuronales Netz	97.58%
(Freund & Schapire, 1996)	Boosting	Entscheidungsbaum	96.8%
(Ragg *et al.*, 1997)	ENZO	Neuronales Netz	98%
(Ragg & Gutjahr, 1997a)	ENZO	NN + Mutual Inf.	98.9%

besteht aus einem zehndimensionalen Eingabevektor und einer Ausgabe. Gesucht ist die Funktion f mit

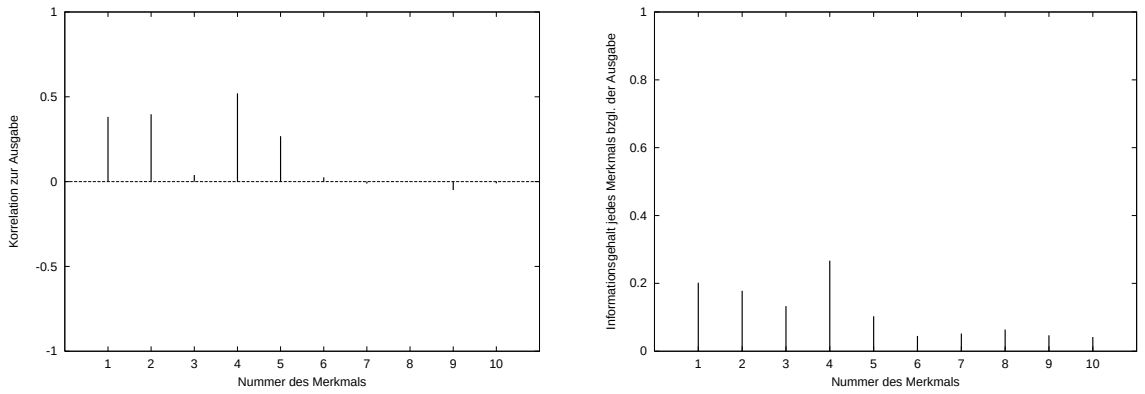
$$f(x_1, \ldots, x_{10}) = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5. \tag{5.1}$$

Die Daten enthalten noch fünf zusätzliche Merkmale, die in einer Rauschkomponente $R(x_6, \ldots, x_{10})$ zusammengefaßt sind. Die Werte der Merkmale sind aus einer $\mathcal{N}(0,1)$-Verteilung gezogen. Die Eingaben werden dabei gleichverteilt aus dem Intervall $[0,1]$ gewonnen.

Bei diesem Problem gibt es also einen relevanten linearen und nichtlinearen Anteil. Aufgrund der Betrachtung irrelevanter Information in einem Teil des Eingabevektors könnte ein neuronales Netz aufgrund der Endlichkeit des Datensatzes nichtlineare Strukturen in den Daten erkennen, die eigentlich nicht vorhanden sind. Hier ist mir unter anderem wichtig, daß sich damit Methoden zur Optimierung der Eingabestruktur bewerten lassen, da die optimale Eingabe bekannt ist. Das Problem wurde auch von Gutjahr in seiner Dissertation verwendet (Gutjahr, 1999), dessen Ergebnisse mir somit als Vergleichsmaßstab dienen können. Der Datensatz besteht aus 5000 Mustern, von denen 500 für das Training und die restlichen 4500 zum Testen verwendet werden. Die Ein- und Ausgabedaten sind symmetrisch in das Intervall [-1,1] skaliert.

Abbildung 5.11 zeigt links die Korrelation der einzelnen Merkmale mit der Ausgabe und rechts wieder den Informationsgehalt. Die Korrelation mancher Rauschkomponenten mit der Ausgabe ist sogar größer als für eines der nichtlinearen Merkmale. Bei diesem Datensatz hilft das Mutual Information Kriterium die Merkmale die Information tragen, von den anderen zu unterscheiden. Die Eingaben 6 bis 10 haben deutlich geringere Werte, allerdings von 0 verschieden, was durch die Endlichkeit des Datensatzes bedingt ist.

Abbildung 5.12 gibt das Resultat der Rückwärtssuche wieder. Es zeigt sich deutlich, daß unterhalb der Eingabekomponenten kaum lineare Abhängigkeiten vorhanden sind und daß



(a) (b)

Abbildung 5.11: Linearer und nicht-linearer Zusammenhang zwischen den einzelnen Merkmalen X_i und der Ausgabe Y für das Add10-Regressionsproblem. (a) Korrelation $\rho(X_i, Y)$. Von den ersten fünf Komponenten haben vier einen deutlichen linearen Anteil. (b) Informationsgehalt $I(X_i, Y)$. Man erkennt beispielsweise, daß der nicht-lineare Zusammenhang des dritten Merkmals mehr Information trägt als das fünfte Merkmal. Bemerkenswert auch, daß in dem Rauschanteil Information vorhanden ist. Dies ist durch die Endlichkeit des Datensatzes bedingt.

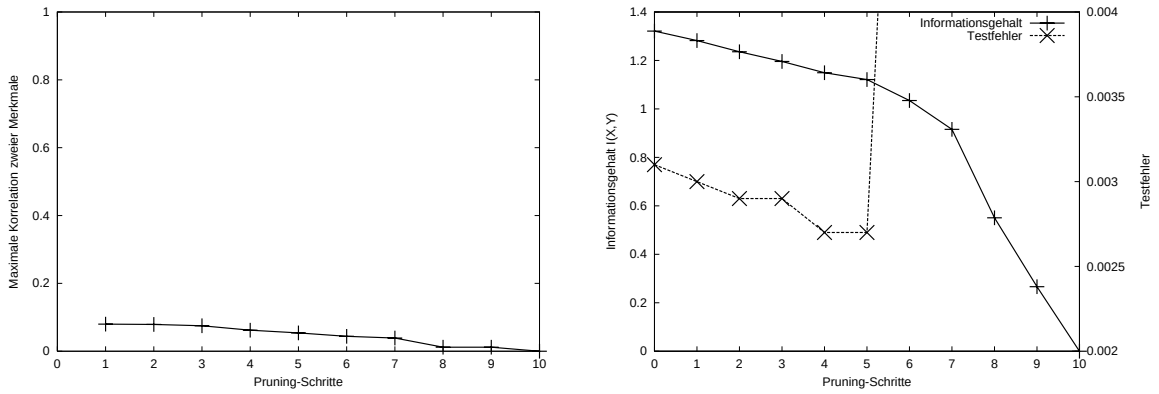
auf diese Weise der Eingabevektor nicht verkleinert werden kann. Die Anwendung des Mutual Information Kriteriums ist in der rechten Abbildung gezeigt. Das Verfahren erkennt sehr schön, daß die Komponenten sechs bis zehn weniger Information enthalten und entfernt sie zuerst. Man erkennt hier deutlich, daß man sechs bis sieben Schritte durchführen kann, ohne daß der Informationsgehalt nennenswert stark sinkt. Ebenfalls eingetragen ist der mittlere Generalisierungfehler von Netzen, die den zugehörigen Eingabevektor verwenden. Die Werte sind gemittelt über 50 Initialisierungen. Die Modelle hatten eine versteckte Schicht mit 15 Neuronen. Obwohl der Informationsgehalt im sechsten Pruning-Schritt nicht auffällig stark sinkt, steigt der Generalisierungsfehler sprunghaft auf etwa das Dreifache an ($\bar{E}_{Schritt-6} = 0.008$), um dann bis auf den 13-fachen Wert zu klettern, wenn nur noch Komponente 4 verwendet wird.

Dieses Beispiel zeigt deutlich, daß die statistischen Kriterien allein noch nicht genügen, um einen optimalen Eingabevektor zu bestimmen. An der Form der Kurve ist die optimale Zahl von 5 Merkmalen nicht abzulesen. Ein integriertes Optimierungskonzept ist aus diesem Grunde unerläßlich. Es sei auch betont, daß der Fehler, wenn man nur 5 Merkmale verwendet, um bis zu 15% kleiner ist als bei Verwendung des gesamten Eingabevektors.

Gutjahr erreichte für dieses Problem einen mittleren Testfehler von 0.00307 für den Fall einer Gewichtsgruppe und 0.00279 für den Ansatz mit mehreren Gewichtsgruppen und variablen Mittelwerten derselben (Gutjahr, 1999).

Schilddrüsen-Klassifikationsproblem

Dieses Klassifikationsproblem ist ebenfalls in der UCI-Datenbank enthalten (Merz & Murphy, 1998) und geht zurück auf die Arbeit von (Schiffmann *et al.*, 1992), die den Datensatz verfügbar gemacht haben. Das Problem besteht darin, aus Patientendaten eine Über-,



(a) (b)

Abbildung 5.12: Anwendung der Rückwärtssuche mit Korrelationskoeffizienten und Mutual Information auf das Add10-Regressionsproblem. (a) Maximale Korrelation zweier Merkmale in jedem Pruningschritt: Bereits zu Anfang liegt die maximale Korrelation zweier Komponenten unter 0.1. Die Zahl der Eingabekomponenten läßt sich damit auf Basis der linearen Abhängigkeiten nicht verkleinern. (b) Informationsgehalt $I(X, Y)$. Die linke y-Achse zeigt den Informationsgehalt des verbliebenen Eingabevektors im jeweiligen Pruningschritt. Auf der rechten y-Achse ist der über 50 Netze gemittelte Generalisierungsfehler für den entsprechendne Eingabevektor eingetragen. Die versteckte Schicht bestand aus 15 Neuronen. Sobald man eine der ersten fünf Komponenten entfernt, steigt der Generalisierungfehler sprunghaft an, ohne daß das am Wert des Mutual Information Kriteriums zu erkennen wäre. Erst nach Entfernen der siebten Komponente sinkt der Informationsgehalt deutlich. Die Reihenfolge der Merkmale beim Pruning ist: 7, 6, 8, 10, 9, 3, 5, 2, 1, 4.

Unter- oder Normalfunktion der Schilddrüse zu bestimmen. Insgesamt stehen 7200 Muster zur Verfügung, von denen ursprünglich ca. die Hälfte zum Training verwendet wurde. Die Daten sind in 21 Merkmalen kodiert, 15 davon sind binär. Die Ausgabe ist als 1-aus-N Kodierung gewählt. Zu Klasse 1 gehören 2.3% aller Fälle, Klasse 2 enthält 5.1%, und Klasse 3 besteht aus dem Löwenanteil von 92.6%. Die einfachste Lösung, eine konstante Ausgabe, hat also eine Fehlerrate von nur 7.4%.

In der Literatur wird eine 21-10-3 Topologie mit sogenannten *shortcut*-Verbindungen verwendet. In dieser Arbeit verwende ich nur Netze ohne shortcut-Verbindungen. Als Klassifikationsleistung wird bei (Schiffmann *et al.*, 1992) 98.4% angegeben. In früheren Studien erreichten wir mit ENZO bis zu 99% (Ragg *et al.*, 1997). Dabei wurden immer 3772 Muster zum Training verwendet.

Der Vorteil dieses Benchmarks ist, daß ausreichend Daten vorhanden sind, um die Ergebnisse zu bewerten. In meinen Untersuchungen in dieser Arbeit verwende ich 720 Muster zum Training, das sind 10%. Den Generalisierungsfehler auf über 6000 Testdaten kann man in diesem Fall als sehr sichere Größe bewerten, bei der Zufallseinflüsse durch Rauschen vernachlässigbar sind.

Die Abbildungen 5.13a-c zeigen die Korrelationen der Merkmale mit den drei Ausgaben. Insbesondere die Merkmale 17,18,19 und 21 weisen Korrelationen mit den Ausgaben 1 und 3 auf. Die linearen Abhängigkeiten zur 2. Ausgabe sind gering. Der Informationsgehalt der einzelnen Komponenten über die gesamte Ausgabe ist in Abbildung 5.13d zu sehen. Hier zeigt sich deutlich, daß die binären Merkmale Information tragen. Die Werte streuen sehr eng um den Wert 0.45. Tabelle 5.1 legt auch für diesen Datensatz nahe, nicht mehr als 6

bis 7 Merkmale zu verwenden, wenn mit allen Trainingsdaten gearbeitet wird. Bei diesem Problem ist es also sinnvoll, bereits vorher Merkmale zu entfernen. In einem Experiment werden wir später sehen, daß das auch zu einer Leistungssteigerung führt.

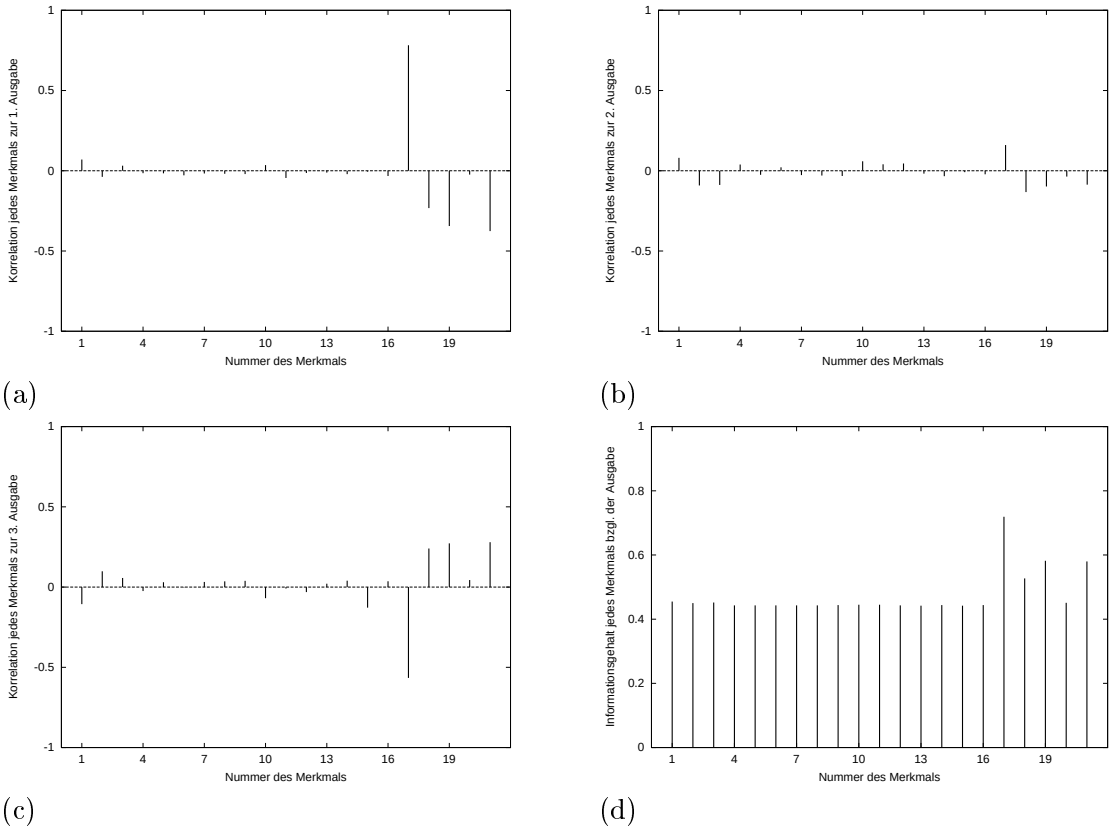
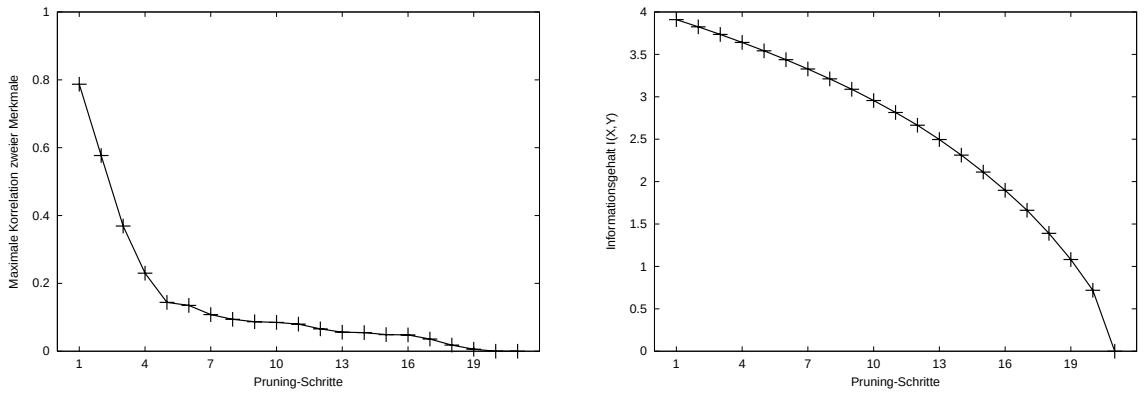


Abbildung 5.13: Die Abbildung zeigt unter a)-c) die Korrelation $\rho(X_i, Y)$ jeder Eingabekomponente X_i mit jeder Ausgabe Y_i für das Schilddrüsen-Klassifikationsproblem. d) Informationsgehalt $I(X_i, Y)$ jeder Eingabekomponente X_i bezüglich des Ausgabevektors Y. Der Informationsgehalt der binären Merkmale ist deutlich von 0 verschieden und hat für alle eine ähnliche Größe.

Um den Eingabevektor zu verringern, kann man wieder die Rückwärtssuche anwenden. Mittels des Korrelationskoeffizienten lassen sich höchstens ein bis zwei Merkmale entfernen, bevor die maximale Korrelation zu weit absinkt (Abbildung 5.14a). Verkleinert man den Eingabevektor basierend auf dem Mutual Information Kriterium, dann läßt sich aus der Grafik (Abbildung 5.14b) auch keine optimale Größe bestimmen. Nach Tabelle 1.4 ist aber eine Obergrenze von 10 Merkmalen durchaus gerechtfertigt. Es werden die 11 Merkmale weggelassen, die durch den Algorithmus zuerst entfernt wurden. Dadurch erreicht der Eingaberaum eine Größe, die mit einem evolutionären Verfahren noch durchsuchbar ist. Daß dieses Vorgehen vernünftig ist, wird im folgenden belegt.

5.2.3 Merkmalselektion basierend auf Mutual Information

In der Einleitung wurde am Beispiel der Schilddrüsen-Klassifikation verdeutlicht, daß die Zahl der Merkmale eine wichtige Rolle für die Leistung des Modells spielt (Abbildung 1.4).



(a) (b)

Abbildung 5.14: Anwendung der Rückwärtssuche mit Korrelationskoeffizienten und Mutual Information auf das Schilddrüsen-Klassifikationsproblem. (a) Maximale Korrelation zweier Merkmale in jedem Pruningschritt: Die maximale Korrelation fällt schnell ab, was insbesondere auf die binären Merkmale zurückzuführen ist. (b) Informationsgehalt $I(X, Y)$. Wie im Falle der Diabetes-Daten sinkt der Informationsgehalt des Eingabevektors relativ gleichmäßig. Die Reihenfolge der Merkmale beim Pruning ist: 15, 13, 7, 5, 16, 8, 14, 9, 6, 4, 12, 19, 11, 20, 2, 10, 3, 1, 21, 18, 17.

In realen Anwendungen ist das Verhältnis dieser Zahl zur Größe der Mustermenge im allgemeinen ungünstig. Bei dem gegebenen Problem existieren 7200 Muster, von denen nur 720 zum Training verwendet werden. Bei einer zunehmenden Zahl an Merkmalen kann die Leistung des Modells beständig abnehmen, da es bei einer zu geringen Datenmenge die bedingte Wahrscheinlichkeit zunehmend schlechter schätzt. Wenn die Merkmale voneinander abhängig oder binärer Natur sind, dann kommt man eventuell mit weniger Mustern für die Dichteschätzung aus. Im allgemeinen lohnt es sich aber, diese Schranken zu beachten.

Tabelle 5.7: Trefferquote für Modelle mit 21 Merkmalen und Modelle mit 8 Merkmalen. Der t-Test zeigt, daß der Unterschied signifikant ist. Eine lineares Modell hat sowohl bei 21 Merkmalen als auch bei 10 Merkmalen eine Trefferquote von 96.6%, d.h. einen Fehler von 3.4%. Der Fehler bei einer konstanten Ausgabe beträgt 7.7%.

Dimensionen	Mittelwert	σ	Maximum	Minimum
21	97.15%	0.35	97.7%	96.14%
10	97.38%	0.30	98.2%	95.8

Abbildung 5.14 zeigt die Ergebnisse der Rückwärtssuche basierend auf Mutual Information für die Schilddrüsenklassifikation. Wählt man als Eingabemerkmale die letzten 10 verbliebenen Merkmale aus, dann läßt sich die durchschnittliche Leistung der Modelle bereits leicht verbessern. Tabelle 5.7 zeigt die mittlere, minimale und maximale Trefferquote, die beim Training von 200 Netzen erreicht wurden. Auf Basis der Generalisierungsleistung ist bereits gerechtfertigt, die Zahl der Merkmale durch die Vorverarbeitung deutlich zu reduzieren. Weiterhin können in der gleichen Zeit bei der kleineren Eingabe etwa viermal mehr

Netze trainiert werden, da die Zahl der Gewichte nur halb so groß ist. Dies hat vor allem den Vorteil, daß der Parameterraum besser durchsucht werden kann, da ungünstige Bereiche aufgrund der geringeren kombinatorischen Vielfalt seltener auftreten. Dadurch sind im Durchschnitt weitere Zugewinne an Leistung gegenüber einer größeren Eingabe zu erwarten.

5.2.4 Initialisierung der Population

Zu Beginn der Evolution ist die Population zu initialisieren. Geht man nicht von der Maximaltopologie aus, dann stellt sich die Frage, wie man die Suchpunkte über dem Parameterraum streuen soll. Die Optimierungsschritte der Vorverarbeitung basierend auf dem Mutual Information Kriterium werden hier in zweierlei Hinsicht von Nutzen sein. Erstens läßt sich den Abbildungen zur Rückwärtssuche entnehmen, wieviel Merkmale ein Netz mindestens haben sollte (Abbildungen 5.8,5.10,5.12,5.14). Dazu wählt man den Wert, an dem die Kurve anfängt, deutlich steiler abzufallen. Zweitens wird, wie bereits erwähnt, durch die Reihenfolge des Prunings eine Ordnung auf den Merkmalen definiert. Sortiert man die Merkmale gemäß dieser Reihenfolge in eine Liste, dann kann man mittels einer Selektionsfunktion die unwichtigeren Merkmale häufiger entfernen (Abbildung 4.3). Den Bevorzugungsfaktor y wählt man aus dem Intervall $[1.5, 2]$.

Tabelle 5.8: Die Tabelle faßt die Ergebnisse für die beiden Strategien zur Initialisierung der Population für alle untersuchten Anwendungen zusammen. Für die Klassifikationsprobleme ist jeweils die mittlere Trefferquot (mit Standardabweichung) auf der Testmenge angegeben, für das Regressionsproblem der mittlere Testfehler. Die angegebenen Werte beziehen sich auf die Leistung der Netze am Ende der ersten Generation der evolutionären Suche. Die Verbesserungen sind gemäß dem t-Test alle signifikant bei der Schwelle $t_{0.95;10} = 1.82$.

Anwendung	Auswahl zufällig	Selektion mit Mutual Information	Mindestzahl an Merkmalen
Diabetes	$73.8\% \pm 0.56$	$76.0\% \pm 0.40$	3
Krebs	$96.6\% \pm 0.24$	$97.1\% \pm 0.18$	3
Add10	0.0298 ± 0.0013	0.0185 ± 0.0018	5
Schilddrüsen (10)	$95.3\% \pm 0.19$	$96.4\% \pm 0.20$	3
Schilddrüsen (21)	$95.3\% \pm 0.17$	$95.8\% \pm 0.14$	3

Für die fünf Datensätze wurden für jede Strategie 10 Versuche durchgeführt und jeweils 100 Netze trainiert. Die Initialisierungen der Netze waren für beide Strategien in jedem Versuch identisch, d.h. der Zufallsgenerator wurde mit der gleichen Zahl gestartet. Die Netze unterschieden sich nur in der Kombination der Merkmale. Die Hyperparameter wurden zweimal angepaßt. Das heißt, die Leistung der Netze wird nach der ersten Generation der evolutionären Suche verglichen. Die Größe der versteckten Schicht variierte zufällig zwischen 3 und 20 Neuronen. Die Zahl der Merkmale wurde zufällig zwischen der Mindestzahl und der Gesamtzahl gewählt.

Für alle Beispiele zeigt sich, daß die geeignete Initialisierung die Qualität der Suchpunkte

deutlich verbessert (Tabelle 5.8). Schlechte Merkmalskombinationen treten in diesem Falle seltener auf. In den folgenden Experimenten wird immer die Initialisierung der Population basierend auf dem Mutual Information Kriterium durchgeführt. Der Bevorzugungsfaktor für die Auswahl der Eingabeneuronen wird immer auf 2 eingestellt.

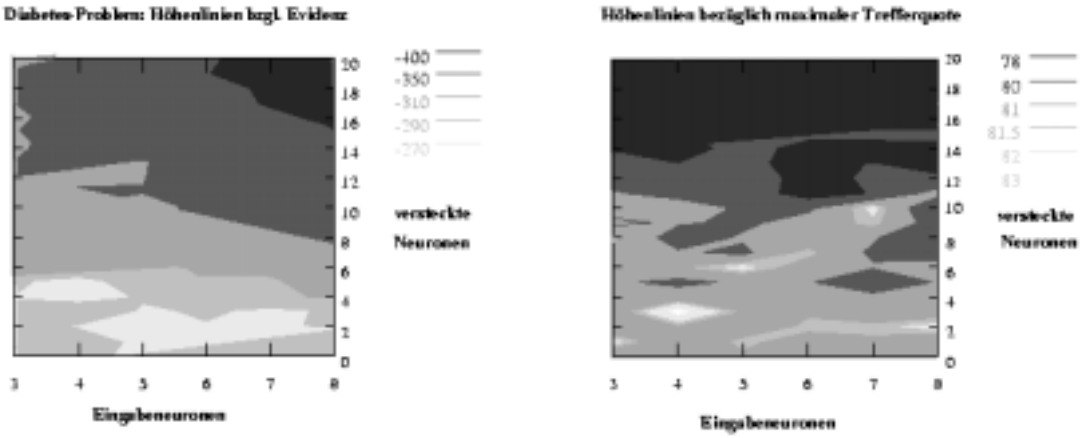
5.2.5 Modelloptimierung

In diesem Abschnitt stelle ich für alle vier Anwendungen einen empirischen Zusammenhang zwischen der Evidenz bzw. dem Testfehler und der Zahl der Merkmale und versteckten Neuronen her. Dabei werden bis zu zwanzig versteckte Neuronen betrachtet, ausgehend von einem linearen Modell mit nur der Eingabe- und Ausgabeschicht. Insgesamt wurden jeweils 4000 Netze trainiert, ca. 30 für jede Kombination der Zahl der Eingabemerkmale und der Zahl der versteckten Neuronen. Dazu werden dann die Ergebnisse der evolutiven Modelloptimierung ins Verhältnis gesetzt. Eingabeneuronen werden im folgenden nur als Merkmale bezeichnet. Für die Neuronen der versteckten Schicht lasse ich die nähere Kennzeichnung als 'versteckt' der besseren Lesbarkeit wegen manchmal weg, da aus dem Zusammenhang ersichtlich ist, was gemeint ist.

Empirischer Zusammenhang

Die Ergebnisse werden in einer zweidimensionalen Grafik mit Höhenlinien präsentiert. Auf der x-Achse ist die Zahl der Eingabemerkmale eingetragen, auf der y-Achse entsprechend die Zahl der versteckten Neuronen. Die Höhenlinien vermitteln ein anschauliches Bild davon, in welchem Bereich Maxima der Evidenz bzw. des Testfehlers liegen. Idealerweise sind beide miteinander negativ korreliert. Bei Klassifikationsproblemen sollte entsprechend die Korrelation mit der Trefferquote positiv sein. Für alle vier Anwendungen ergaben sich deutliche Werte. Die geringste Korrelation war 0.43 für das Krebs-Problem, dann 0.60 (Diabetes) und nahezu perfekt mit -0.89 für das Add10-Problem bzw. 0.90 für die Schilddrüsenklassifikation.

Für das Diabetes-Klassifikationsproblem ergibt sich, daß die Evidenz mit zunehmender Zahl an versteckten Neuronen abnimmt (Abbildung 5.15a). Verwendet man mehr als vier Merkmale, dann liegt die optimale Zahl an Neuronen im Bereich von 0 bis 3. Im Falle von drei oder vier Merkmalen liegt das Maximum bei 4 bis 5 Neuronen. Abbildung 5.15b zeigt in schöner Übereinstimmung, daß die Trefferquote bei zunehmender Komplexität ebenso sinkt. Bereiche mit guter Generalisierung finden sich insbesondere dort, wo auch die Evidenz hoch ist, mit einem Maximum bei 4 Merkmalen und 3 bis 4 Neuronen. Vereinzelt gibt es auch Modelle größerer Komplexität mit hoher Trefferquote. Die Testmenge hat insgesamt 192 Muster. Aufgrund der geringen Größe ist diese Leistungsmessung mit Rauschen behaftet, d.h. der Wert kann auch einmal zufällig gut sein, z.B. der Ausreißer bei 7 Merkmalen und 10 Neuronen. Bei zusammenhängenden größeren Gebieten ist das dagegen nicht anzunehmen. Die optimale Zahl an Merkmalen für diesen Datensatz liegt nach Tabelle 5.1 bei vier. Die empirischen Ergebnisse bestätigen diesen Wert. Der Fluch der Dimensionen zeigt sich hier also darin, daß bei zu großen Merkmalsvektoren die linearen oder fast linearen Modelle am besten abschneiden.



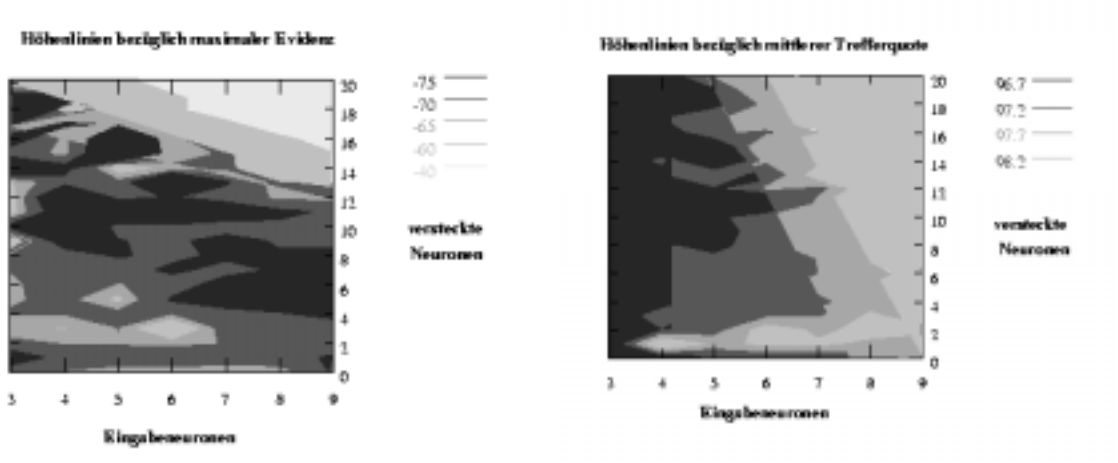
(a) (b)

Abbildung 5.15: Die Abbildung zeigt für das Diabetes-Problem empirische Abhängigkeit der Evidenz bzw. Trefferquote auf der Testmenge von der Zahl der Merkmale und versteckten Neuronen. Insgesamt wurden ca. 4000 Netze trainiert und damit eine Kartierung mit Höhenlininen erstellt. a) Netze mit vielen Eingabemerkmalen haben nur eine große Evidenz, wenn die Zahl der Neuronen klein ist, d.h. das Modell ist nahezu linear (Curse of dimensionality!). Dagegen können Netze mit wenigen Merkmalen auch eine größere Netzkomplexität verwenden. Maximale Evidenz erreicht man mit 3 bis 4 Merkmalen und 4 bis 5 Neuronen. (b) Die Bereiche hoher Evidenz decken sich sehr schön mit den Bereichen einer guten Generalisierungsleistung. Die beste Trefferquote liegt bei einem Netz mit 4 Merkmalen und 5 Neuronen. Die Korrelation zwischen Evidenz und Trefferquote auf der Testmenge (berechnet für alle 4000 Netze) beträgt 0.60.

Ein ganz eigenartiges Verhalten ergibt sich für die Krebs-Klassifikation. Abbildung 5.16a kann man entnehmen, daß in einem trichterförmigen Tal von rechts unten nach links oben ein Bereich niedrigerer Evidenz liegt. Erstaunlicherweise haben hier Modelle mit vielen versteckten Neuronen eine hohe Evidenz. Insgesamt zerfällt die Grafik im Gegensatz zum Diabetes-Problem in viele Inseln. Der Testfehler verhält sich ganz ähnlich. Betrachtet man die maximale Trefferquote, dann ergeben sich sehr viele abgegrenzte Bereiche. Da die Leistung der Modelle insgesamt sehr ähnlich ist, ist die Funktion sehr flach mit vielen kleinen Erhebungen. Das heißt, es gibt fast überall Modelle mit hoher Evidenz und Trefferquote. Von daher zeigt Abbildung 5.16b ausnahmsweise den mittleren Testfehler. Hier kann man zumindest das trichterförmige Tal erkennen, das sich auch in der linken Abbildung abzeichnete.

Die Tatsache, daß Modelle mit vielen versteckten Neuronen keine Anzeichen von Overfitting zeigen, beruht darauf, daß die Merkmale ordinalen Charakter haben. Das heißt, die Muster liegen nicht dicht im Eingaberaum, sondern sind auf einem Gitter angeordnet. Die zunehmende Leistung zeigt, daß die Punkte der verschiendenen Klassen auf der Gitterstruktur an manchen Stellen mit einer linearen Funktion nicht richtig zu trennen sind, was auch intuitiv verständlich ist. Für das Schätzen der bedingten Wahrscheinlichkeit reduziert sich die Aufgabe dann auf den Fall einer diskreten Verteilung.

Für das Add10-Problem ist uns die optimale Eingabe bekannt und ebenso, daß ein nichtlinearer Zusammenhang existiert. Die empirischen Ergebnisse genügen diesem Sachverhalt

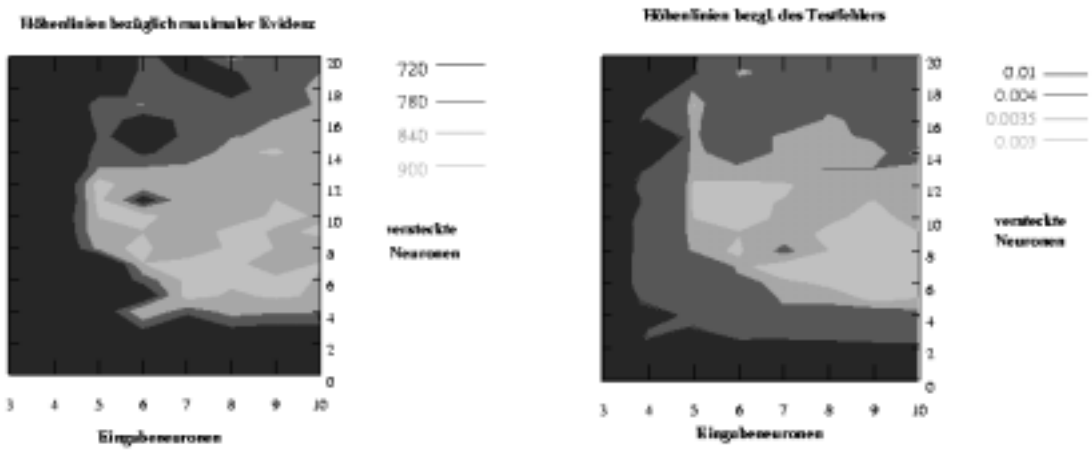


(a) (b)

Abbildung 5.16: Krebs-Klassifikation. Mit Höhenlinien verdeutlichte empirische Abhängigkeit der Evidenz bzw. Trefferquote auf der Testmenge von der Zahl der Merkmale und versteckten Neuronen. Siehe auch Unterschrift zu Abbildung 5.15. a)Es ergibt sich von rechts unten nach links oben ein breiter Trichter mit niedriger Evidenz mit vereinzelten Inseln von einigen Modellen mit höherer Evidenz. Die Evidenz steigt zu den linearen Modellen hin an, ebenso zu Netzen mit großer Zahl an Neuronen. (b) Für die mittlere Trefferquote ist der Trichter von rechts unten nach links gut zu erkennen. In schöner Übereinstimmung mit der linken Abbildung zeigen Modelle mit wenigen Neuronen (≤ 2) oder solche mit vielen (≥ 10) die beste Leistung. Die Korrelation zwischen Evidenz und Trefferquote auf der Testmenge (berechnet für alle 4000 Netze) beträgt 0.43.

in schöner Weise (Abbildung 5.17a). Bereiche maximaler Evidenz liegen, wenn man mehr als 6 Merkmale benutzt, zwischen 6 und 10 Neuronen (einschließlich). Bei 5 oder 6 Merkmalen ist die Evidenz maximal im Falle von 10 bis 12 Neuronen. Verwendet man weniger Merkmale oder weniger versteckte Neuronen, dann fällt die Evidenz stark ab. Im anderen Fall, wenn die versteckte Schicht zu groß gewählt ist, sinkt die Evidenz ebenfalls deutlich. Abbildung 5.17b zeigt für den Testfehler ein analoges Verhalten. Im Gegensatz zu den ersten beiden Problemen besteht die Testmenge hier aus 4500 Mustern. Dadurch kann man den Wert als gute Approximation der tatsächlichen Generalisierungsleistung ansehen. Der Zusammenhang des Testfehlers mit der Zahl der Merkmale und Neuronen stimmt nahezu perfekt mit dem Zusammenhang für die Evidenz überein. Daß die Leistung bei der Verwendung von mehr als den 5 optimalen Merkmalen nicht so stark abnimmt, liegt darin begründet, daß die zusatzlichen Merkmale wenig Information tragen.

Als letztes Beispiel wird das Schilddrüsen-Problem betrachtet. Der Eingabevektor wurde bereits durch die Vorverarbeitung auf 10 Merkmale reduziert. Da die Testmenge hier ebenfalls groß ist – sie umfaßt 6480 Muster – sollten die empirischen Ergebnisse für die Evidenz und den Testfehler ebenso eindeutig ausfallen wie für das Add10-Problem. Abbildung 5.18 zeigt auch hier eine schöne Übereinstimung. Verwendet man weniger als 7 Merkmale, dann kann die versteckte Schicht wie im Fall der Diabetes-Klassifikation etwas größer ausfallen. Nimmt die Zahl der Merkmale zu, dann liegt das Optimum bei zwei bis drei Neuronen. Trefferquoten von ca. 98% erziehlt man immer, wenn man zwischen einschließlich 4 und 7 Merkmalen verwendet. Optimal ist die Kombination von vier bis fünf Merkmalen und zwei



(a) (b)

Abbildung 5.17: Add10-Regressionsproblem. Mit Höhenlinien verdeutlichte empirische Abhängigkeit der Evidenz bzw. des Testfehlers von der Zahl der Merkmale und versteckten Neuronen. Siehe auch Unterschrift zu Abbildung 5.15. a) Bereiche maximaler Evidenz setzen voraus, daß mindestens die fünf notwendigen Merkmale im Eingabevektor vorhanden sind. Die optimale Zahl an Neuronen in der versteckten Schicht liegt zwischen 6 und 10 bzw. 12, falls die Zahl der Merkmale fünf oder sechs beträgt. (b) Die Kartierung für den Testfehler stimmt sehr schön mit derjenigen für die Evidenz überein. Die Korrelation zwischen Evidenz und dem Fehler auf der Testmenge (berechnet für alle 4000 Netze) beträgt -0.89, d.h. sie ist nahezu perfekt.

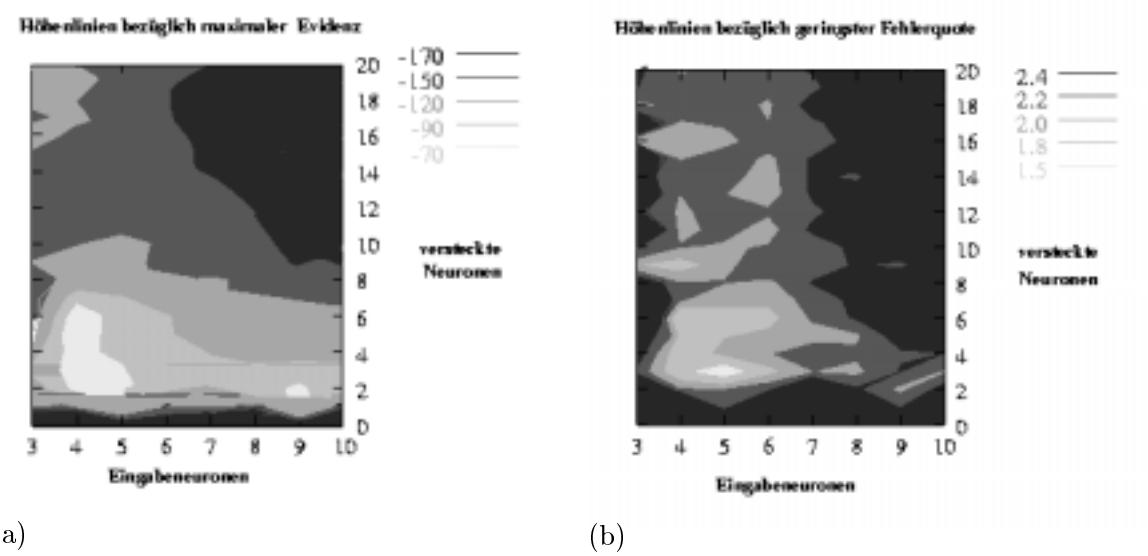
bis sechs versteckten Neuronen. Auch hier werden die angegebenen Grenzen aus Tabelle 5.1 bestätigt.

Voraussetzung für die gute Leistung ist das Finden der richtigen Merkmalskombination. Die stastischen Kriterien erfüllen hier eine wichtige Funktion im evolutiven Suchprozeß. Die Netze mit hoher Leistung wurden in den Experimenten durch eine breite Zufallssuche gefunden, indem jeweils ca. 4000 Netze trainiert wurden. Im nächsten Abschnitt werden wir sehen, daß die evolutive Optimierung bei wesentlich geringerem Aufwand die Maxima der Evidenz ebenso findet.

Ergebnisse der evolutiven Optimierung

Nachdem jetzt der empirische Zusammenhang zwischen Evidenz bzw. Generalisierungsleistung mit der Zahl der Merkmale und der Größe der versteckten Schicht bekannt ist, stellt sich die Frage, ob die evolutive Optimierung die Maxima der Evidenz findet. Dazu sind im folgenden die Ergebnisse der Modelloptimierung in den Kartierungen markiert. Für alle Probleme wurden 10 Versuche durchgeführt. Verglichen wird die Leistung mit dem linearen Modell und der durchschnittlichen Leistung (über 100 Netze), die man mittels des Bayes'schen Lernens erhält. Weiterhin wird der letztgenannte Versuch noch zehnmal wiederholt und jeweils das Netz mit der höchsten Evidenz ausgewählt. Für das Bayes'sche Lernen wurden der gesamte Merkmalsvektor und immer zehn Neuronen in der versteckten Schicht verwendet.

Für das evolutive Verfahren wurden die Merkmalsvektoren der Netze unter Verwendung des Mutual Information Kriteriums initialisiert, wie das bereits ausgeführt wurde. Um möglichst



(a) (b)

Abbildung 5.18: Schilddrüsenklassifikation. Mit Höhenlinien verdeutlichte empirische Abhängigkeit der Evidenz bzw. Trefferquote auf der Testmenge von der Zahl der Merkmale und versteckten Neuronen. Siehe auch Unterschrift zu Abbildung 5.15. a) Es zeigt sich ein ähnliches Verhalten wie bei der Diabetes-Klassifikation. Der Bereich maximaler Evidenz liegt zwischen 4 und 6 Merkmalen (einschließlich) und weniger als 6 Neuronen. (b) Auch hier stimmt die Kartierung für die Trefferquote sehr schön mit der für die Evidenz überein. Die Korrelation zwischen Evidenz und der Trefferquote auf der Testmenge (berechnet für alle 4000 Netze) beträgt 0.90, d.h. sie ist ebenfalls nahezu perfekt.

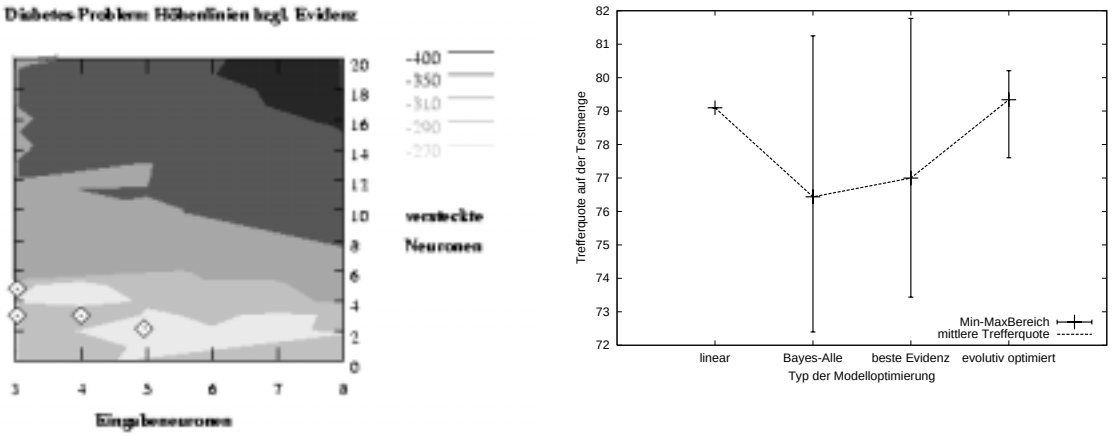
viele Kombinationen an Eingabemerkmalen zu generieren, wurden die Randbereiche bei der Initialisierung nicht berücksichtigt. Das heißt, die minimale Zahl an Merkmalen war immer um eins größer als in Tabelle 5.8 angegeben. Die maximale Zahl war um eins kleiner als die Zahl aller Merkmale. Die Größe der versteckten Schicht wurde für alle Netze (in allen Versuchen) zufällig aus dem Intervall $[3, 10]$ gezogen.

Im Falle des Diabetes-Problems hatten die Netze am Anfang der Evolution demnach zwischen 4 und 7 Merkmale und zwischen 3 und 10 versteckte Neuronen. Die Größe der versteckten Schicht wurde so bestimmt, daß die Zahl der Gewichte für das größte Netz etwa einem Zehntel der Zahl der Muster entspricht (vgl. dazu (White, 1989)).

Für die Krebsklassifikation ist die gewählte Initialisierung suboptimal. Würde man eine größere Zahl an versteckten Neuronen wählen, dann fände die Evolution immer das Maximum der Evidenz. Diese Eigenschaft, daß die Evidenz aufgrund des ordinalen Charakters der Daten mit zunehmender Zahl an versteckten Neuronen ansteigt, ist allerdings erst aufgefallen, nachdem ich den empirischen Zusammenhang hergestellt hatte. Auf eine weitere Versuchsreihe wurde verzichtet, da das Ergebnis offensichtlich ist. Andererseits sollte das Verfahren möglichst unabhängig von der Initialisierung sein, d.h. es sollten keine Experimente nötig sein, um die Parameter der Evolution einzustellen. Betrachtet man ein Problem mit ordinalen Daten, dann wird man die hier gewonnenen Erkenntnisse in Zukunft berücksichtigen. Für einen reellwertigen Eingaberaum ist die gewählte Initialisierung in jedem Fall gerechtfertigt.

Die schwarz-weißen Rauten in Abbildung 5.19a markieren die Topologien, die durch zehn

Versuche der evolutiven Optimierung gefunden wurden. Sie konzentrieren sich alle im Bereich von 3 bis 5 Merkmalen und 2 bis 4 versteckten Neuronen. Der Bereich des Maximums wird also beständig gefunden. Aus diesem Grund ist die Leistung der Modelle im Mittel auch etwas besser als das lineare Modell. Für diesen Datensatz ist die Leistung des linearen Modelles aufgrund des hohen Rauschanteils und der spärlichen Datenmenge, insbesondere des geringen Anteils an Testdaten, schwer zu schlagen.



(a) (b)

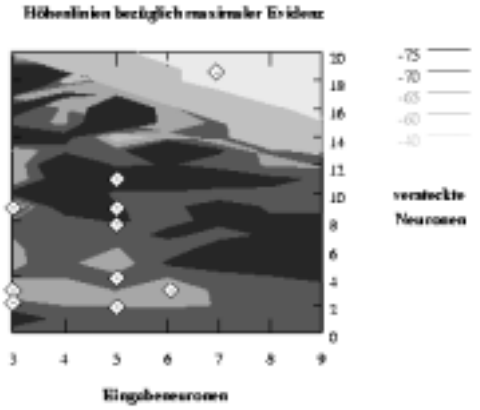
Abbildung 5.19: Diabetesklassifikation. Vergleich der Ergebnisse der evolutiven Optimierung mit der durch Höhenlinien verdeutlichten empirischen Abhängigkeit der Evidenz von der Zahl der Merkmale und versteckten Neuronen aus Abbildung 5.15. (a) Die schwarz-weißen Rauten markieren die Topologien der 10 Netze, mit denen das evolutive Verfahren in 10 Versuchen terminierte. Das Verfahren konvergierte immer gegen eine von 4 Topologien. Die durchschnittliche Evidenz der 10 Netze betrug -274. (b) Vergleich der Trefferquoten für verschiedene Methoden der Modelloptimierung. Außer der evolutiven Optimierung sind alle Methoden im Mittel schlechter als das lineare Modell. Die starke Streuung der nicht-linearen Modelle ist auch durch die kleine Testmenge von 192 Mustern bedingt.

Abbildung 5.19b zeigt die Leistung der Modelle auf der Testmenge. Ein lineares Modell erreicht 79.1%. 100 mit Bayes'schem Lernen trainierte Netze erreichen im Mittel 76.4%, streuen aber insgesamt zwischen 72.4% und 81.3%. Führt man zehn Versuche des Bayes'schen Lernens durch (mit je 100 Netzen) und wählt das Netz mit der höchsten Evidenz aus, dann erreicht man im Mittel bereits 77% und maximal 81.8%. Die evolutive Optimierung liegt als einzige Methode im Mittel mit 79.4% über dem linearen Modell. Der einzige Versuch, der unter 79.1% lag, terminierte bei 77.6%. Das Maximum lag bei 80.2%.

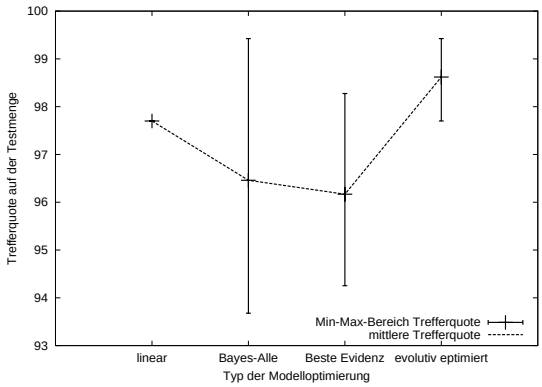
Es sei angemerkt, daß bei Vertauschen von Trainings- und Testmenge die nicht-linearen Modelle im Vergleich besser abschneiden. Das liegt daran, daß die Testmenge mit 576 Mustern dann wesentlich größer ist und die Ergebnisse damit aussagefähiger sind. Die Vergleichbarkeit mit aus der Literatur bekannten Studien würde in dem Fall aber verlorengehen.

Die Krebs-Klassifikation zeigte für die Bereiche mit hoher Evidenz ein uneinheitliches Bild. Es gibt offenbar aufgrund der ordinalen Struktur der Eingabe sehr viele lokale Maxima. Die Varianz der Evidenz im gesamten Raum ist im Vergleich zu den anderen Problemem sehr viel niedriger. Die evolutive Optimierung findet in diesem Fall auch in 10 Versuchen 10 verschiedene Topologien (Abbildung 5.20a). Dies ist ein deutlicher Hinweis darauf, daß man hier mit Komitees einen weiteren Qualitätssprung erreichen kann. Wir werden darauf

zurückkommen. Die Evidenz der 10 Modelle beträgt im Mittel 40, d.h. sie liegt in etwa in der Größenordnung des Randmaximums. Das Gefälle der Evidenz in diese Richtung ist also eher schwach ausgeprägt. Das trägt auch zur Erklärung bei, daß verschiedene Topologien gefunden werden.



(a)



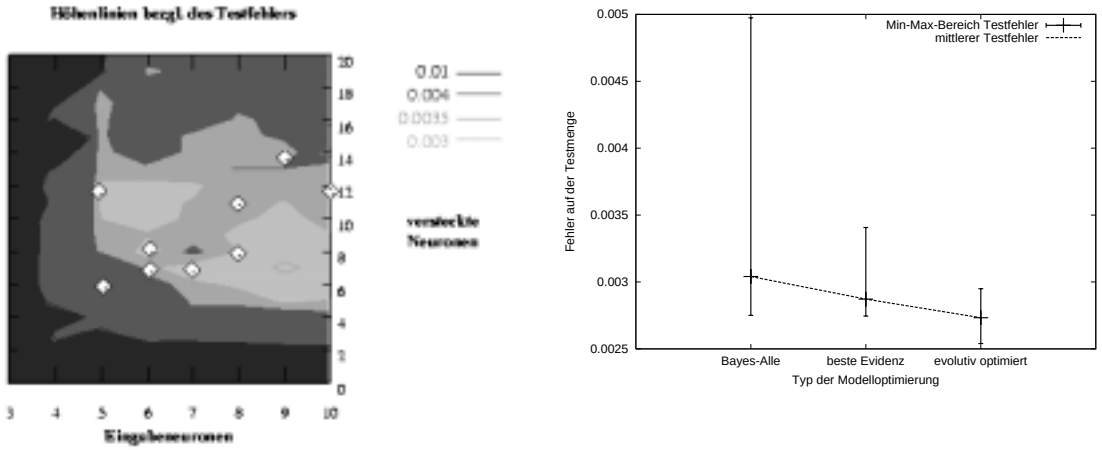
(b)

Abbildung 5.20: Krebsklassifikation. Vergleich der Ergebnisse der evolutiven Optimierung mit der durch Höhenlinien verdeutlichten empirischen Abhängigkeit der Evidenz von der Zahl der Merkmale und versteckten Neuronen aus Abbildung 5.16. (a) Die schwarz-weißen Rauten markieren die Topologien der 10 Netze, mit denen das evolutive Verfahren in 10 Versuchen terminierte. Das Verfahren terminierte nicht so einheitlich wie im Falle des Diabetes-Problems. Obwohl mit weniger als zehn versteckten Neuronen initialisiert wurde, wird einmal sogar das Randmaximum mit einer breiten versteckten Schicht gefunden. Die mittlere Evidenz der 10 Netze beträgt 40. (b) Vergleich der Trefferquoten für verschiedene Methoden der Modelloptimierung. Die beiden anderen Methoden sind im Mittel wieder schlechter als das lineare Modell. Die evolutive Optimierung schneidet in keinem der 10 Versuche schlechter ab. Der Anteil an Fehlklassifikationen wurde im Mittel von 2.3% auf 1.4% reduziert. Das ist eine Verbesserung um 40%.

Ein lineares Modell erreicht für diesen Datensatz 97.7% (Abbildung 5.20b). Mit Bayes'schem Lernen erreicht man im Mittel 96.5%, maximal 99.4%. Wählt man aus zehn Versuchen mit je 100 Netzen jeweils das Netz mit der höchsten Evidenz aus, dann erreicht man im Mittel 96.2% und maximal 98.3%. Die evolutive Optimierung ist nie schlechter als das lineare Modell, im Mittel sogar um 1.4% besser. Dies entspricht einer Reduktion des Anteils an Fehlklassifikationen um ca. 40%. Das beste Ergebnis war auch hier 99.4%. Wie bereits erwähnt, gibt es sehr viele Inseln mit hoher Evidenz bzw. hoher Trefferquote. Man kann unter den 10 gefundenen Topologien auch keinen Bereich erkennen, der eine bessere Leistung verspricht.

Für die anderen beiden Probleme war die Korrelation zwischen Evidenz und Testfehler, wie bereits gesehen, wesentlich stärker. Ebenso hatte die Testmenge eine aussagefähige Größe. Man würde in diesem Fall erwarten, daß die Ergebnisse deutlicher ausfallen.

Abbildung 5.21a zeigt, daß bis auf eine Topologie alle im Bereich (empirisch) hoher Evidenz liegen. Der Ausreißer erreicht immerhin noch einen Wert von 880, liegt also nicht viel darunter. Im Mittel erreichen die 10 Netze eine Evidenz von 940. Im Vergleich zu den beiden anderen Methoden ist der mittlere Testfehler signifikant kleiner (Abbildung 5.21b). Die Reduktion vom 0.00304 (Durchschnitt Bayes'sches Lernen) auf 0.00273 entspricht 11% Ver-



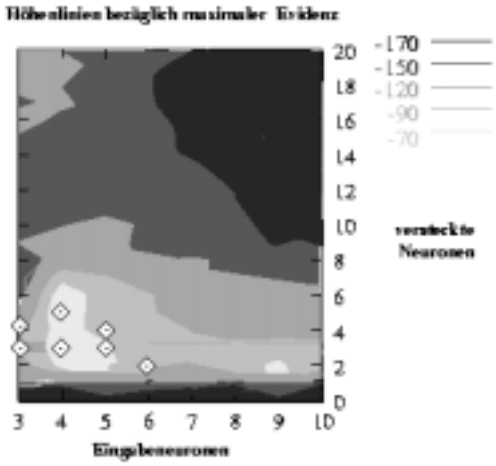
(a) (b)

Abbildung 5.21: Add10-Regressionsproblem. Vergleich der Ergebnisse der evolutiven Optimierung mit der durch Höhenlinien verdeutlichten empirischen Abhängigkeit der Evidenz von der Zahl der Merkmale und versteckten Neuronen aus Abbildung 5.17 (a) Die schwarz-weißen Rauten markieren wieder die Topologien, die durch die evolutive Optimierung gefunden wurden. Eine davon tritt doppelt auf. Die Evidenz der 10 Netze beträgt im Mittel 940. (b) Vergleich der Testfehler für verschiedene Methoden der Modelloptimierung. Das lineare Modell hat einen Fehler von 0.016 und bleibt hier unberücksichtigt. Die evolutive Optimierung schneidet im Mittel signifikant besser ab. Gegenüber dem Durchschnitt aller Netze ist der Testfehler um 11% kleiner, gegenüber der Auswahl des Netzes mit der höchsten Evidenz um 5%. Insbesondere ist wieder die Streuung bezüglich des Fehlers kleiner.

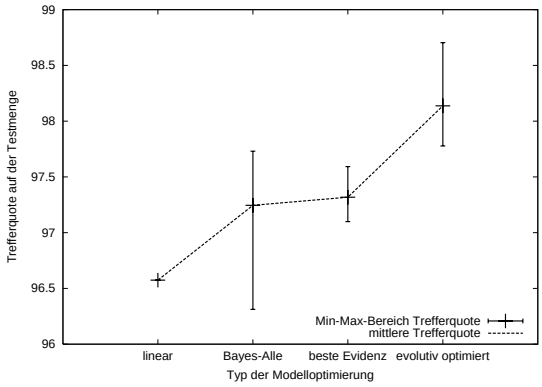
besserung. Gegenüber dem mittleren Fehler von 0.00287 der Netze mit der besten Evidenz sind es noch 5% Verbesserung. Ein lineares Modell erreicht hier einen Fehler von 0.016, das ist fünfmal so groß. Das ist bemerkenswert, da die Korrelation zur Ausgabe etwa so stark war wie für das Diabetes-Problem. Im letzteren Fall war das lineare Modell aber deutlich besser.

Ein schönes Ergebnis ergibt sich für die Schilddrüsen-Klassifikation. Hier hatten wir einen deutlichen Bereich maximaler Evidenz. Alle Topologien, die durch die evolutive Optimierung gefunden werden, gruppieren sich in diesem Bereich (Abbildung 5.22a). Die Zahl der versteckten Neuronen streut von 3 bis 6 und die der Merkmale von 2 bis 5. Damit liegt man im 'grünen' Bereich gemäß Tabelle 5.1. Abbildung 5.18b zeigt, daß in diesem Bereich die Trefferquote auch signifikant höher liegt, ca. bei 98.2%. Die evolutiv optimierten Modelle erreichen mindestens 97.8%, im Mittel 98.2% und maximal 98.7%. Damit reduziert sich der erwartete Fehler gegenüber einem linearen Modell von 3.7% auf etwa die Hälfte, d.h. 1.8%. Auch gegenüber den beiden anderen Methoden ist der mittlere Fehler deutlich geringer (Abbildung 5.22b).

Die Experimente in diesem Abschnitt haben gezeigt, daß es sinnvoll ist, die Evidenz als Optimierungskriterium zu wählen und darauf aufbauend den Parameterraum systematisch, beispielsweise mit einem evolutiven Verfahren, zu durchsuchen. Die Evidenz der optimierten Netze lag höher als die der emprisch gefundenen. Das liegt daran, daß die empirische Suche einen Punkt des Parameterraumes zufällig auswählt und dann nur den Gewichtsvektor optimiert. Bereiche höherer Evidenz können so nicht systematisch gefunden werden. Die evolutionäre Strategie verschiebt die Suchpunkte dagegegen immer in die Richtungen



(a)



(b)

Abbildung 5.22: Schilddrüsenklassifikation. Vergleich der Ergebnisse der evolutiven Optimierung mit der durch Höhenlinien verdeutlichten empirischen Abhängigkeit der Evidenz von der Zahl der Merkmale und versteckten Neuronen aus Abbildung 5.18 (a) Die schwarz-weißen Rauten markieren wieder die Topologien, die durch die evolutive Optimierung gefunden wurden. Alle Modelle gruppieren sich im Bereich maximaler Evidenz in der linken unteren Ecke. Die mittlere Evidenz der 10 Netze beträgt -64. (b) Vergleich der Trefferquoten für verschiedene Methoden der Modelloptimierung. Die beiden anderen Methoden sind im Mittel wieder schlechter als das lineare Modell. In diesem Fall schneiden alle Methoden besser ab als das lineare Modell. Der erwartete Anteil an Fehlklassifikationen reduziert sich von 3.7% (lineares Modell) über 2.8% (Bayes im Mittel) und 2.6% (höchste Evidenz) auf 1.8% für die evolutive Optimierung.

höherer Evidenz. Übertragen auf das Beispiel des Bergsteigers aus Kapitel 4 bedeutet das, daß er an einer kleinen Erhebung stehenbleibt, auch wenn es daneben noch weiter den Berg hinauf geht. Durch die evolutionäre Strategie findet er den Weg.

Weiterhin hat sich gezeigt, daß die Evidenz stark mit dem Testfehler korreliert war, insbesondere dann, wenn die Testmenge groß war. Das bedeutet zum einen, wenn die Korrelation stark ist, daß man durch die evolutive Optimierung systematisch die besten Modelle finden kann. Im Falle der beiden Klassifikationsprobleme, Diabetes und Krebs, kann man umgekehrt schließen, daß die Testmenge zu klein ist, um die Güte der Modelle eindeutig zu beurteilen. In der Einleitung wurde bereits dargestellt, daß man auf kleinen Testmengen immer Modelle finden kann, die einen kleinen Fehler haben, aber trotzdem kein angemessenes Modell der Daten sind. Das zeigt umso mehr, daß Verfahren, die auf Kreuzvalidierung beruhen, im allgemeinen nur suboptimale Parametereinstellungen haben werden. Verwendet man die Kreuzvalidierungsmenge iterativ, dann neigen die Verfahren zum Overfitting. Der Evidenz-Ansatz ermöglicht es hier, auf allen Daten zu trainieren und weiterhin systematisch die Qualität der Modelle durch Suche zu optimieren. Dies ist insbesondere auch wichtig, wenn man ein Komitee von Netzen bilden möchte. Der Gesamtfehler des Komitees hängt auch von der durchschnittlichen Leistung der einzelnen Mitglieder ab (vgl. Gleichung (3.3)). In welcher Weise die evolutionäre Suche von Generation zu Generation fortschreitet, wird im nächsten Abschnitt beleuchtet.

5.2.6 Verlauf der Evolution

Wie im Falle der Optimierung der Zahl der versteckten Neuronen ist es auch hier interessant, den Verlauf der Evolution an einem Beispiel zu visualisieren. Dafür wähle ich für das Diabetes-Problem eine spezielle Initialisierung, die weit genug von einem Maximum der Evidenz entfernt liegt und zeige, wie sich die Suche durch den Parameterraum bewegt. Dabei ist das Netz mit der höchsten Evidenz jeweils extra gekennzeichnet (Abbildung 5.23). Man kann im Verlauf der Evolution sehr schön sehen, wie sich die Suchpunkte im Parameterraum ausbreiten, bis die Suche dann in einem Bereich kollabiert. Ein Netztypus hat sich durchgesetzt. An der Bewegung des Netzes mit der höchsten Evidenz ist zu erkennen, daß sich die Richtung der Suche mehrmals verändert.

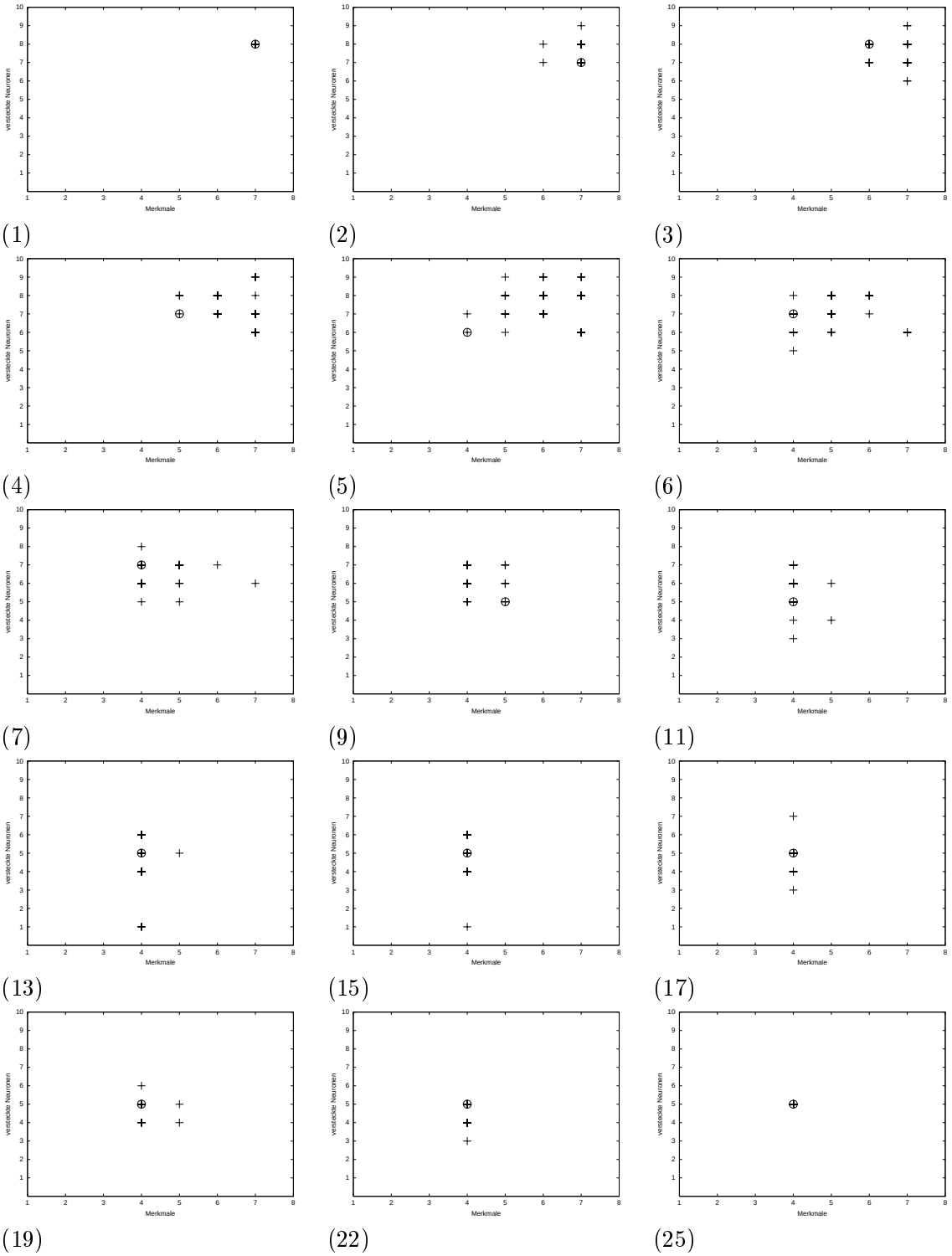
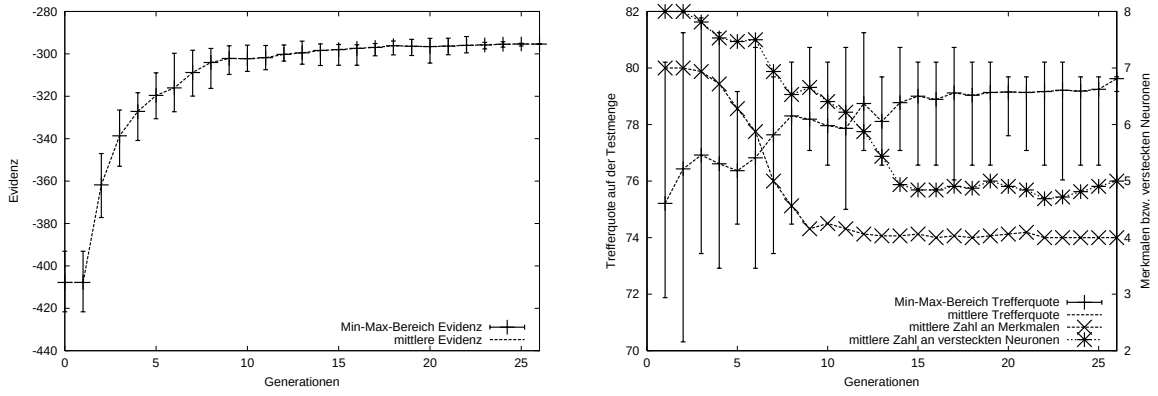


Abbildung 5.23: Verlauf der Evolution in Abhängigkeit der Zahl der Merkmale und versteckten Neuronen. Die Zahl in Klammern gibt jeweils die Generation an. Insgesamt wurden 25 Generationen gerechnet. Die Suche wird mit Netzen initialisiert, die 7 Merkmale und 8 versteckte Neuronen haben. Wie man es gemäß Abbildung 5.15 erwarten würde, breiten sich die Netze in Richtung zunehmnder Evidenz aus, bis die Suche in einem Maximum kollabiert. Das Netz mit maximaler Evidenz springt mehrmals im Raum. Das heißt, es gibt mehrere Bereiche mit ähnlich hoher Evidenz, von denen sich einer durchsetzt.

Abbildung 5.24a entnimmt man den Verlauf der mittleren, minimalen und maximalen Evidenz. Bis etwa zur zwanzigsten Generation nimmt die Evidenz mehr oder weniger kontinuierlich zu und pendelt dann um ein Maximum. In der rechten Abbildung ist neben der Trefferquote auch die mittlere Zahl an Merkmalen und versteckten Neuronen dargestellt. Die Trefferquote bewegt sich in schöner Übereinstimmung mit der Evidenz nach oben (Abbildung 5.24b). Gleichzeitig nimmt die Zahl an Merkmalen und versteckten Neuronen beständig ab, bis ein Maximum der Evidenz erreicht ist. Dann bleiben die Werte konstant. In manchen Experimenten beobachtet man, daß die mittlere Zahl an Merkmalen manchmal für mehrere Generationen stabil bleibt, bevor sie weiter absinkt. In diesem Fall fanden mehr ungünstige Mutationen statt, so daß die entstandenen kleineren Netze nicht überlebten. Je kleiner der Eingabevektor ist, desto größer ist die Chance, daß ein wichtiges Merkmal entfernt wird.



(a) (b)

Abbildung 5.24: Die Abbildung zeigt die Entwicklung der Evidenz und der Trefferquote in Abhängigkeit der Zahl der Generationen. (a) Mittlere, minimale und maximale Evidenz: Bis zur zehnten Generation nimmt die Evidenz deutlich zu und pendelt dann um das Maximum. (b) Entwicklung der Trefferquote auf der Testmenge und der mittleren Netzkomplexität, d.h. der Zahl an Merkmalen und versteckten Neuronen. Die mittlere Leistung der Netze wird zunehmend besser. Auch hier pendelt die Trefferquote ab der zehnten Generation um eine feste Größe. Die Zahl der versteckten Neuronen und der Merkmale nimmt kontinuierlich ab. Die mittlere Zahl der Merkmale bleibt streckenweise über einige Generationen konstant und nimmt dann wieder ab. In diesen Fällen sind die Mutationen ungünstig ausgefallen, d.h. wichtige Merkmale wurden entfernt, so daß die kleineren Netze nicht überlebt haben.

5.3 Grenzen der Modelloptimierung

Eine wesentliche Voraussetzung für die evolutive Optimierung ist die Korrelation von Evidenz und Generalisierungsfehler. Sind die Merkmale günstig gewählt und ausreichend Muster verfügbar, dann wird die Korrelation im allgemeinen gut sein. Bei allen bisher betrachteten Experimenten war das der Fall.

Je größer der Rauschanteil in den Daten, desto mehr Muster sind nötig, damit die Verteilung des Rauschens richtig geschätzt wird. Das heißt, mit zunehmendem Rauschen in den Daten werden die Auswahlkriterien (Evidenz, Kreuzvalidierung) immer unsicherer. Dies liegt daran, daß es dann mehrere alternative Erklärungen für die Daten gibt.

Vergleicht man die beiden Datensätze aus Abbildung 1.1, dann ist erkennbar, daß für die

rechte Abbildung statt der Sinusfunktion genausogut ein linearer Prozeß zugrunde liegen könnte. Um diese Grenzfälle zu untersuchen, wurden zu der Sinusfunktion zwei verschiedene Verteilungen von Rauschen addiert. Das linke Histogramm in Abbildung 5.25 zeigt das Rauschen für die bereits bekannten Daten aus Abbildung 1.1. Dieser Datensatz wurde nach der Vorschrift $sin(x) + 0.6 \cdot rand(x)$ gebildet. Die Funktion $rand(x)$ steht für die Zufallszahlen, wie sie in dem Histogramm gezeigt sind. Der zweite Datensatz, der im folgenden betrachtet wird, ist mit gleichverteiltem Rauschen behaftet (Abbildung 5.25b). Die Berechnungsvorschrift war $sin(x) + 0.2 \cdot rand(x)$. Damit wurde der selbe Faktor gewählt wie für das früher verwendete Problem aus Abbildung 1.1a, nur die Verteilung des Rauschens ist eine andere.

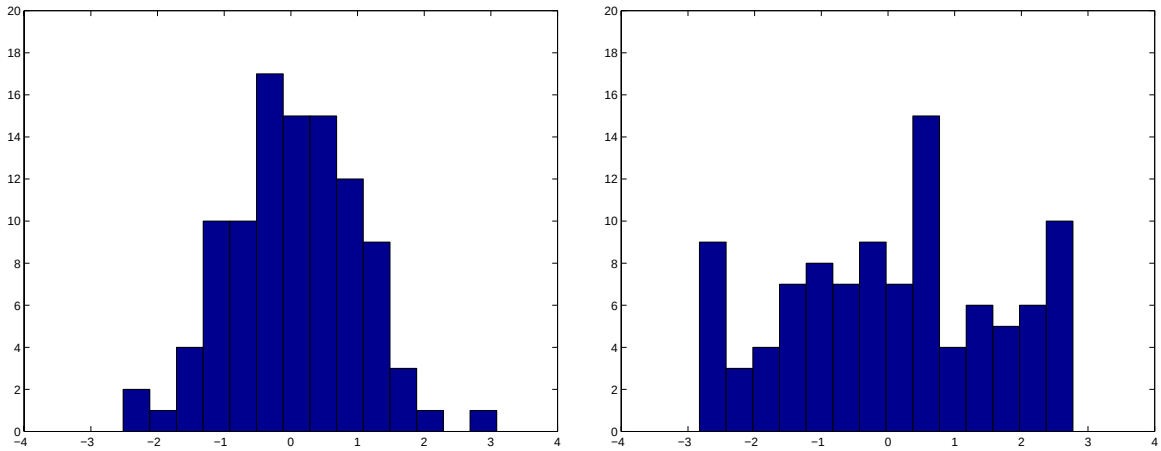
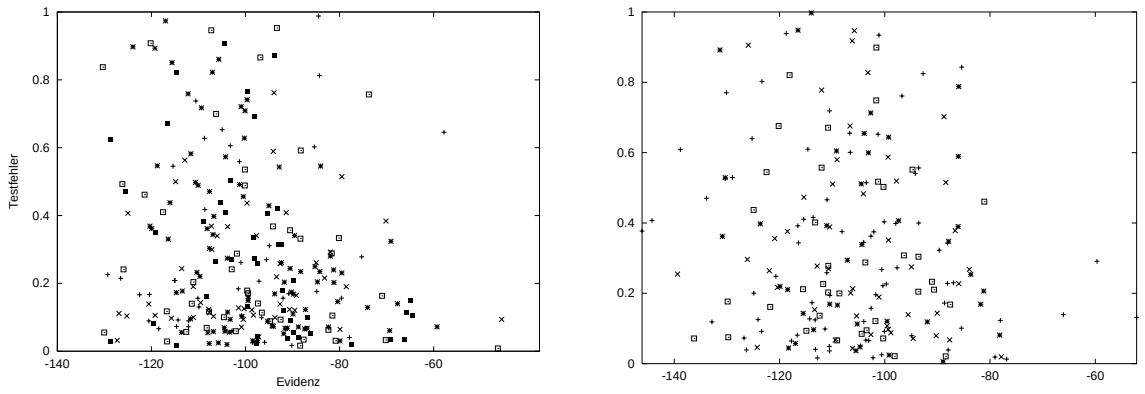


Abbildung 5.25: Die Abbildung zeigt die Histogramme von zwei Datensätzen mit je 100 Punkten. Die Histogramme weichen mehr oder weniger stark von der Idealform einer Standardnormalverteilung ab. (a) Daten gezogen aus einer Standardnormalverteilung. Der Kolmogorov-Smirnov Test ergibt beim Vergleich dieses Datensatzes mit 2500 Datenpunkten aus einer Standardnormalverteilung einen p-Wert von 0.85. Das Rauschen wurde mit einem Faktor von 0.6 zu den Originaldaten addiert. Das entspricht einer Varianz von ca. 0.4, d.h. sie ist zehnmal größer gegenüber dem Rauschen in Abbildung 1.1a. (b) Datensatz mit extremer Abweichung von der Standardnormalverteilung. Der KS-Test ergibt im Vergleich zu dem Datensatz mit 2500 Mustern einen p-Wert von 0.01. Die Verteilung der 100 Punkte entspricht eher einer Gleichverteilung. Das Rauschen wurde mit einem Faktor von 0.2 zu den Originaldaten addiert. Die Varianz beträgt dann ebenfalls ca. 0.4

Für die beiden Datensätze wurden in 10 Versuchen jeweils 50 Netze trainiert. Die Topologie war genauso gewählt wie schon zu Anfang für das Sinus-Regressionsproblem ($1 - 20 - 10 - 1$). Abbildung 5.26 zeigt den Zusammenhang zwischen Evidenz und Generalisierungsfehler. Dieser wurde bezüglich der Orginal-Sinusfunktion berechnet. Gegenüber den Daten, wie sie bisher verwendet wurden, nimmt die Korrelation von Evidenz und Testfehler deutlich ab. Für den stärkeren Rauschanteil sinkt sie (betragsmäßig) auf -0.21 (Abbildung 5.26a), für die andere Verteilung des Rauschens auf -0.13. In beiden Fällen gibt es Netze, die sowohl eine hohe Evidenz und trotzdem einen hohen Fehler haben.

Wählt man aus den zehn Versuchen jeweils die zehn Netze mit der höchsten Evidenz aus, dann beträgt der mittlere Fehler für den ersten Fall 0.185 und streut zwischen 0.08 und 0.65. Für den zweiten Datensatz liegt der mittlere Fehler der zehn Netze bei 0.159 und streut zwischen 0.019 und 0.46.



(a) (b)

Abbildung 5.26: Die Abbildung zeigt den Zusammenhang zwischen Evidenz und Testfehler für je 250 Netze, die mit den stark verrauschten Sinus-Regressionsdaten trainiert wurden. Der Übersichtlichkeit wegen sind nur 5 Versuche à 50 Netze abgebildet. Der erste der beiden Datensätze wurde in Abbildung 1.1b gezeigt. (a) Das Sinus-Regressionsproblem mit stärkerem Rauschen. Die Korrelation zwischen Testfehler und Evidenz beträgt jetzt nur noch -0.21. Insbesondere gibt es auch Modelle mit hoher Evidenz und hohem Fehler. (b) Das Sinus-Regressionsproblem mit anders verteiltem Rauschen. Die Korrelation zwischen Testfehler und Evidenz beträgt noch -0.13. Es gibt ebenso auch Modelle mit hoher Evidenz und hohem Fehler.

Die Ergebnisse zeigen, daß die Modellauswahl desto unsicherer wird, je stärker das Rauschen ist. Das Netz mit der höchsten Evidenz kann auch einen sehr hohen Testfehler haben. Wie wir an den bisherigen Beispielen gesehen haben, reduziert die evolutive Optimierung der Netze die Unsicherheiten bei der Modellauswahl. Man würde erwarten, daß der mittlere Fehler deutlich niedriger liegt. Die Resultate der Experimente, die das belegen, werden im nächsten Kapitel zusammen mit der Evolution unabhängiger Netze vorgestellt. Wenn allerdings verschiedene Maxima existieren, dann findet die evolutive Modelloptimierung im allgemeinen nur eines davon. Durch Integration über die verschiedenen Lösungen kann zum einen aber der Generalisierungsfehler weiter verringert und zum anderen die Unsicherheit bei der Modellauswahl eliminiert werden.

Probleme, die einen starken Rauschanteil besitzen, sind genau die Fälle, für die eine Komiteelösung besonders interessant ist. Gibt es ein eindeutiges Optimum, wie z.B. für das Add10-Regressionsproblem, dann wird ein Komiteeansatz die beste

und der geringen Datenmenge mehrere konkurrierende Erklärungen, dann kann die Komiteebildung zu einer signifikanten Leistungssteigerung führen.

5.4 Zusammenfassung

Die Experimente in diesem Kapitel haben gezeigt, daß die geeignete Verzahnung des iterativen Bayes'schen Verfahrens und des evolutionären Algorithmus es ermöglicht, den Parameterraum effizient nach Modellen mit hoher Evidenz zu durchsuchen. Der Anwender ist davon entlastet, eine optimale Topologie selbst zu finden, sondern kann dies vollständig dem Verfahren überlassen. Voraussetzung dafür ist, daß die Annahmen des Bayes'schen Verfahrens nicht verletzt sind.

Zuerst wurde nachgewiesen, daß die Strategie, nach zwei Anpassungsschritten Nachkommen zu generieren, am günstigsten ist, wie das vorher mit einer Plausibilitätsüberlegung begründet wurde. Das Sinus-Regressionsproblem und das Sinus-Klassifikationsproblem dienten als Beispiele dafür, daß die Suche nach der optimalen Anzahl an versteckten Neuronen zum gewünschten Ergebnis führt, wie das in Abbildung 4.2 ganz allgemein illustriert wurde.

Im zweiten Abschnitt dieses Kapitels wurde insbesondere die Merkmalselektion betrachtet. Ausgehend von der Vorüberlegung, daß in realen Anwendungen oft viele Merkmale verfügbar, gleichzeitig aber die vorhandenen Datenmengen klein sind, wurden zwei Algorithmen vorgeschlagen, um eine Vorauswahl zu treffen. Anhand des Schilddrüsen-Problems wurde gezeigt, daß dadurch die Ergebnisse verbessert werden können. Weiterhin wurde für alle Anwendungen gezeigt, daß die Information, die man durch die Voroptimierung gewonnen hat, auch für die Initialisierung der Population verwendet werden kann und die Suche wesentlich verbessert.

Für vier reale Anwendungen habe ich durch Training vieler Netze den Zusammenhang zwischen Evidenz bzw. Testfehler und Zahl der Merkmale sowie Zahl der versteckten Neuronen hergestellt. Die Ergebnisse wurden mittels einer Höhenliniengrafik präsentiert, die es erlaubt, visuell zu erfassen, in welchen Bereichen des Parameterraumes eine hohe Evidenz bzw. ein geringer Testfehler zu erwarten sind und daß diese, wenn man die Abbildungen übereinander legt, sich nahezu decken. Es hat sich gezeigt, daß die evolutionäre Suche auch in diesem komplexeren Parameterraum die Bereiche mit hoher Evidenz bzw. mit geringem Testfehler findet. Im Vergleich zur Optimierung der Zahl der versteckten Neuronen ist die Suche nach einer geeigneten Kombination von Merkmalen ungleich schwieriger.

Weiterhin weisen die evolutiven optimierten Netze eine höhere Evidenz auf als normal trainierte Netze. Das liegt daran, daß die evolutive Suche systematisch den ganzen Parameterraum durchsucht, während beim normalen Training zufällig Topologien herausgegriffen und nur die Gewichte und Hyperparameter optimiert werden. Die Bestimmung einer geeigneten Merkmalskombination und der optimalen Zahl an versteckten Neuronen entspricht im letzten Fall also gerade einer Zufallssuche.

Die Korrelation der Evidenz mit dem Testfehler war besonders hoch, wenn die Testmenge groß war. Für das Add10-Problem und die Schilddrüsen-Klassifikation waren zehnmal mehr Testdaten vorhanden als Trainingsdaten. Damit zeigte sich, daß in diesen Fällen die Evidenz nahezu ein perfekter Indikator für die Güte des Modells ist. Waren weniger Daten vorhanden, dann war auch die Korrelation deutlich niedriger. Daran ist zu erkennen, daß die Leistungsmessung auf kleinen Testmengen mit einer starken Unsicherheit behaftet ist. Das Evidenz-Kriterium ist deswegen der Parameterbestimmung oder auch Modellauswahl mittels Kreuzvalidierung vorzuziehen, da alle Daten zum Training verwendet werden können.

Zusammenfassend läßt sich sagen, daß durch die vorgestellten Methoden das Problem, ein Modell mit möglichst hoher a posteriori Wahrscheinlichkeit zu finden, hinreichend gelöst ist, solange ausreichend Daten vorhanden sind. Im nächsten Kapitel wird darauf aufbauend die Frage behandelt, wie man ein gutes Komitee aus neuronalen Netzen systematisch finden kann. Die Komiteebildung ist insbesondere dann von Wichtigkeit, wenn die Datenmengen klein und mit Rauschen behaftet sind.

Kapitel 6

Evolution unabhängiger Modelle

Das Konzept der evolutiven Modelloptimierung ermöglicht es, den Parameterraum nach Netzen maximaler Evidenz zu durchsuchen. In den Experimenten des vorigen Kapitels konnte empirisch nachgewiesen werden, daß Bereiche hoher Evidenz einen niedrigen Generalisierungsfehler aufweisen. Wiederholt man diese Suche mehrfach, dann erhält man im allgemeinen jedesmal ein anderes Ergebnis. Das heißt, es werden verschiedene Maxima der Evidenz gefunden. Der durchschnittliche Fehler dieser Netze wird klein sein in dem Sinne, wie die Dekomposition des Komiteefehlers (3.3) das fordert. Wendet man auf die erhaltenen Netze das Clusterverfahren aus Kapitel 3.5 an und bestimmt die Kombination an Netzen, die Gleichung (3.12) maximiert, dann wird das so definierte Komitee maximale Generalisierungsleistung haben.

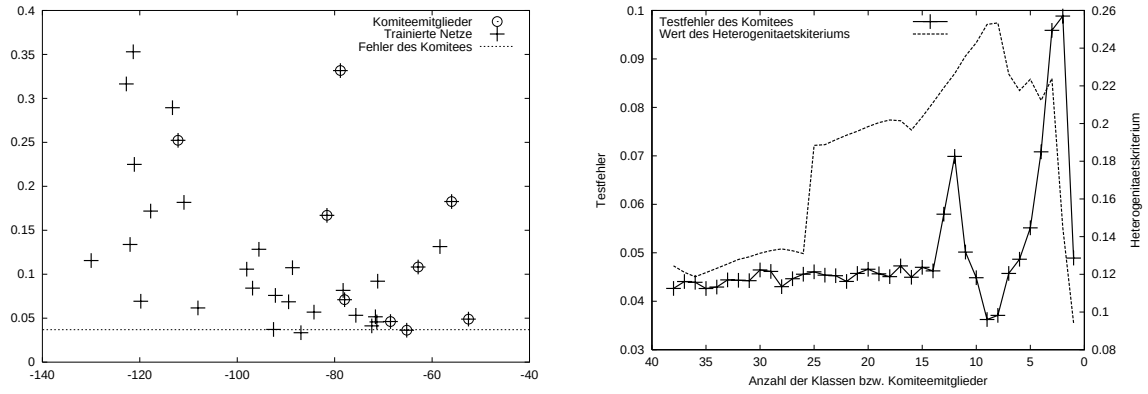
Im folgenden möchte ich die Frage behandeln, ob man das Komitee nicht direkt, innerhalb eines einzigen Evolutionsprozesses, gewinnen kann. Diese verzahnte Lösung wäre wesentlich effizienter, als beide Schritte sequentiell durchzuführen. Dazu ist es notwendig, das Wissen um die Abhängigkeit zwischen den Populationsmitgliedern in den Selektionsprozeß einzubringen. Die Idee kann man mit einem einfachen Beispiel veranschaulichen.

In einer Landschaft werden mehrere Sucher ausgesetzt, deren Aufgabe es ist, Nahrung zu suchen und sich durch Teilung fortzupflanzen. Die Aufgabe der Sucher ist nur, sich in möglichst vielen neuen Suchern der nächsten Generation weiter zu entwickeln. Die Fortpflanzungswahrscheinlichkeit eines Individuums ist direkt proportional zur Nahrung, die es findet. Es gibt in der Landschaft mehrere Gebiete. Einige davon sind sehr fruchtbar, andere erlauben den Individuen nur ein kärgliches Dasein. Die Sucher ziehen beständig umher. Verlassen sie ein fruchtbares Gebiet, dann laufen sie Gefahr zu verhungern, da das Nahrungsangebot zu klein sein kann. Befinden sie sich in einem Gebiet zusammen mit vielen anderen, dann müssen sie ihre Nahrung teilen. Allein ein weniger fruchtbares Gebiet zu besiedeln, kann also für ein Individuum vorteilhafter sein. Um als Sucher erfolgreich zu sein in dem Sinne, eine große Nachkommenschaft zu haben, spielt sowohl das Nahrungsangebot als auch sein Standort eine Rolle. Diesen Prozeß müssen wir nachbilden, wenn neben der Evidenz (Nahrungsangebot) auch die Unabhängigkeit der Netze (Standort) bei der Selektion beachtet werden soll. Die technische Umsetzung dieser Idee wird im folgenden beschrieben.

6.1 Evolution unabhängiger Modelle

Die Untersuchungen im vorigen Kapitel (Abschnitt 5.3) haben verdeutlicht, daß bei zunehmendem Rauschen in den Daten und/oder spärlicher Datenlage verschiedene mögliche Modelle des Datensatzes existieren können. Es gibt dann mehrere Maxima der Evidenz, die aber einen grundsätztlich anderen Generalisierungsfehler haben können. Abbildung 6.1a zeigt diese Situation für das Sinus-Regressionsproblem mit starkem Rauschen aus Abbildung 1.1b. Das Netz mit der höchsten Evidenz (ca. -50) hat einen Fehler von etwa 0.05. Danach folgt ein Netz mit großem und zwei mit einem mittelgroßen Fehler, gefolgt von einigen Modellen, die wieder einen Fehler zwischen 0.05 und 0.1 haben. Dann kommen aber gleich wieder zwei Modelle mit einem relativ großen Fehler. Die Auswahl des Modells mit der höchsten Evidenz ist jetzt ein sehr unsicheres Verfahren.

Clustert man die Netze bezüglich ihrer Ähnlichkeit, wie das in Kapitel 3.5 beschrieben wurde, dann kann man die Unsicherheit bei der Modellauswahl stark abmildern, indem man ein Komitee aus den 'unabhängigsten' Netzen bildet. In Abbildung 6.1b ist der Wert des Kriteriums (3.11) in Abhängigkeit der Klassenzahl abgebildet. Der Wert ganz links entspricht dem Fall, daß alle Netze eine eigene Klasse bilden. In jedem Schritt in Richtung absteigender Klassenzahl wurden zwei Klassen verschmolzen und der zugehörige Komiteefehler berechnet. Durch die Komiteebildung ist jetzt keine Modellauswahl mehr nötig. Das Komitee erreicht aber denselben niedrigen Testfehler wie nur drei weitere Netze in der Population (linke Abbildung). Aufgrund ihrer schlechteren Evidenz wären diese aber nicht selektiert worden.



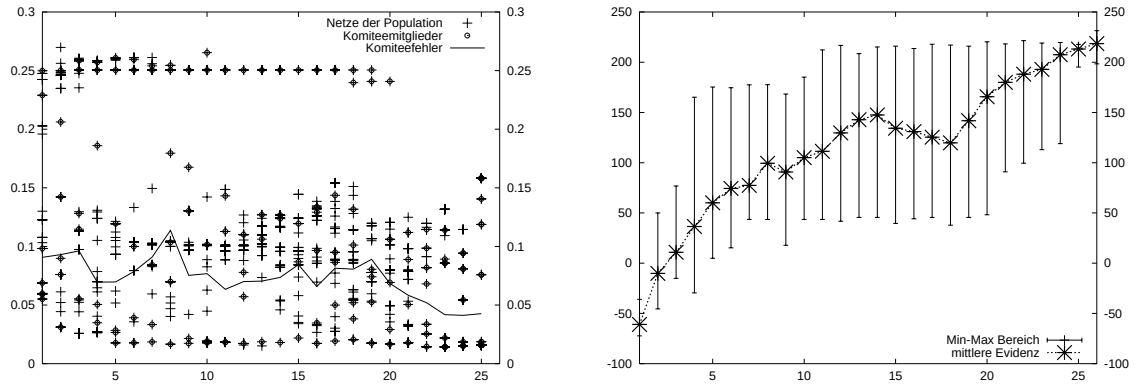
(a) (b)

Abbildung 6.1: Die Abbildung zeigt für das Sinus-Regressionsproblem mit starkem Rauschen die Leistung einzelner Netze und die des Komitees, das aus den unabhängigsten Netzen gebildet wurde. (a) Zusammenhang zwischen Evidenz und Testfehler. Es gibt etliche Netze mit einer hohen Evidenz und einem hohen Testfehler. Die Auswahl eines Modells ist mit großer Unsicherheit behaftet. Die Netze, die durch das Clusterverfahren selektiert wurden, sind mit einem Kreis markiert. Die Linie unten im Bild gibt den Fehler des Komitees wieder, der nur von einem anderen Netz leicht unterschritten wird. (b) Zusammenhang zwischen dem Wert des Heterogenitätskriteriums und dem Fehler des Komitees, wenn zunehmend Klassen verschmolzen werden. Das Maximum des Kriteriums deckt sich mit dem Minimum des Komiteefehlers. Das Komitee hat einen Fehler von ca. 0.037.

Für die evolutive Modelloptimierung wurden die Netze nur nach ihrer Evidenz sortiert. Auf diese Liste wurde dann eine Auswahlfunktion wie in Abbildung 4.3 angewendet, die fittere Netze bevorzugt. Möchte man jetzt sowohl bezüglich der Evidenz als auch der Un-

abhängigkeit selektieren, dann kann man das erreichen, indem man die Liste umordnet. Mögliche Strategien dazu wurden in Kapitel 4.5 vorgeschlagen. Im folgenden Abschnitt werde ich die verschiedenen Möglichkeiten, den Selektionsoperator zu definieren, anhand des Sinus-Regressionsproblems vergleichen, das am Ende des vorigen Kapitels betrachtet wurde. Weiterhin wird dann für die zusätzlichen Mutationsoperatoren untersucht, inwieweit sie den Verlauf der Evolution günstig beeinflussen.

Für das zugrundegelegte Beispiel ist in Abbildung 6.2 die Entwicklung der Population dargestellt. Für jede Generation sind alle Netze mit ihrem Testfehler abgebildet. Die Kreise markieren die Komiteemitglieder, die durch die Clusterung gewonnen wurden.



(a) (b)

Abbildung 6.2: Die Abbildung zeigt für das Sinus-Regressionsproblem mit starkem Rauschen die Entwicklung der Population und des Komitees während der Evolution. (a) Testfehler der einzelnen Netze und Fehler des Komitees in jeder Generation. Die Kreise markieren die Netze, die durch das Clusterverfahren als Komiteemitglieder ausgewählt wurden . Es existieren mehrere konkurrierende Entwicklungslinien. (b) Verlauf der mittleren, minimalen und maximalen Evidenz. Die Evidenz nimmt beständig zu. In Bereichen hoher Evidenz liegen Netze mit völlig unterschiedlichem Testfehler.

Es ist deutlich zu erkennen, daß es mehrere konkurrierende Bereiche hoher Evidenz gibt. Die rechte Abbildung zeigt, daß die mittlere Evidenz beständig ansteigt. Es bilden sich aber verschiedenartige Modelle heraus, erkennbar an dem gänzlich unterschiedlichen Testfehler (linke Abbildung). Der Komiteefehler, den man erhält, wenn man in jeder Generation aus jeder Klasse die Netze mit der höchsten Evidenz auswählt, ist durch die Linie markiert. Er liegt immer im unteren Bereich. Durch die Mittelung erhält man ein gut generalisierendes Modell. Als Parametereinstellung für diesen Versuch wurde diejenige gewählt, die sich im folgenden Abschnitt als die günstigste erweisen wird.

6.1.1 Vergleich der Selektionsstrategien

Am Ende jeder Generation werden die μ potentiellen Eltern ausgewählt. Auf diese Netze wird dann das Clusterverfahren angewendet, so daß eine Menge von Klassen vorliegt. Wieviele Klassen pro Generation entstehen, hängt auch von den zufälligen Mutationen ab. Um zu vermeiden, daß von Generation zu Generation starke Sprünge in der Zahl der Klassen auftreten, kann man die Anzahl abweichend von der optimalen Einstellung gemäß dem Kriterium gedämpft verändern (Strategie $\mathcal{K}$). Die zu verwendende Anzahl ergibt sich dann aus

einer Interpolation zwischen dem Wert der letzten Generation und dem aktuellen Wert. Eine langsame Veränderung während der Suche soll eine gleichmäßige Konvergenz ermöglichen.

Innerhalb einer Klasse werden die Netze nach ihrer Evidenz sortiert. Die Klassen werden nach einer der folgenden Strategien $\mathcal{E}$ oder $\mathcal{U}$ angeordnet.

Strategie $\mathcal{E}$ Die Klassen werden nach der maximalen Evidenz, die ein Mitglied der Klasse erreicht, sortiert.

Strategie $\mathcal{U}$ Die Klassen werden nach ihrer Unabhängigkeit sortiert. Dazu wird die Reihenfolge ihrer Verschmelzung verwendet, wie in Abbildung 4.11 illustriert wurde.

Strategie $\mathcal{B}$ Die besten Netze jeder Klasse, d.h. die Komiteemitglieder, erzeugen bevorzugt Nachkommen. Sortierung der Klassen nach einer der obigen Strategien. Im Extremfall werden nur die Vetreter jeder Klasse berücksichtigt.

Strategie $\mathcal{K}$ Klein vererben, groß variieren. Die Zahl der Klassen wird zu Anfang der Evolution optimal gemäß dem Kriterium eingestellt. Verändert sie sich sprunghaft während der Evolution, dann wird diese Änderung nur gedämpft an die nächste Generation weitergegeben.

Die sortierte Liste wird nun so generiert, daß am Anfang das Netz mit der höchsten Evidenz aus der an erster Stelle stehenden Klasse kommt, gefolgt von dem Netz mit der höchsten Evidenz aus der an zweiter Stelle stehenden Klasse, usw. Sind die besten Netze jeder Klasse einsortiert, dann wird auf die nächstbesten das gleiche Verfahren angewendet. Etwas formeller ausgedrückt:

$$NN_i \prec NN_j \quad :\Longleftrightarrow \quad \begin{array}{l}(1)\ \text{Ranking von } NN_i \text{ in seiner Klasse } K_{NN_i} \text{ ist kleiner als} \\ \qquad \text{Ranking von } NN_j \text{ in dessen Klasse } K_{NN_j} \\ (2)\ \text{bei Gleichheit: } K_{NN_i} \prec K_{NN_j}\end{array}$$

wobei $NN_i, i = 1, \ldots, \mu$, alle Netze der Population darstellen. Im folgenden werden die Ergebnisse der Experimente wiedergegeben. Es wurden für jede Strategie jeweils 10 Durchläufe gemacht. Die allgemeinen Parameter waren wie folgt eingestellt: $\mu = 45$; $\lambda = 50$; Generationen $= 15$; $p_{Mut-Hidden} = 0.2$; Strategie $\mathcal{K}$ wurde verwendet; Bevorzugungsfaktor für höhere Evidenz und beste Netze jeder Klasse $= 1.2$. Alle Parameter des Bayes'schen Verfahrens waren eingestellt, wie in Kapitel 5 bereits angegeben. Wird zusätzlich die Eingabestruktur optimiert, sollte man λ weiter erhöhen, gegebenenfalls auch die Zahl der Generationen verlängern, um die Konvergenz des Verfahrens sicherzustellen. Maximal verwende ich $\lambda = 100$.

In den Experimenten wurden zuerst die beiden Strategien $\mathcal{E}$ und $\mathcal{U}$ verglichen. Prinzipiell wird nach jedem Experiment nur mit der besseren Parametereinstellung weitergearbeitet. Abbildung 6.3 zeigt, daß es günstiger ist, die Klassen nach maximaler Evidenz zu sortieren. Das schlechtere Abschneiden von $\mathcal{U}$ liegt darin begründet, daß sich der Zeitpunkt der Verschmelzung zweier Klassen und damit die Position in der Liste in jeder Generation verändert.

Der momentane Selektionsvorteil einer Klase kann also in der nächsten Generation schon wieder verloren sein. Es ist aber für die evolutive Optimierung wichtig, daß der Vorteil beständig wirkt. Weiterhin verändert sich die Unabhängigkeit einer Klasse von den anderen, wenn mehr Nachkommen aus der ersteren entsprungen sind. Diese haben dann einen größeren Anteil an der Population und teilen sich eventuell sogar in mehrere Klassen auf.

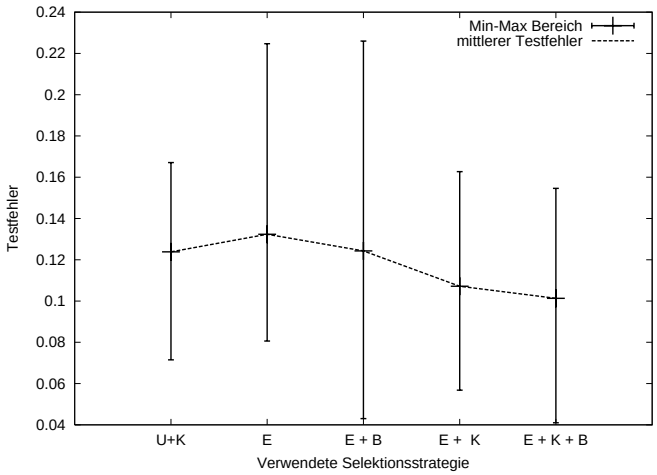


Abbildung 6.3: Vergleich der vorgeschlagenen Selektionsstrategien. Die Abbildung zeigt den mittleren Komiteefehler sowie den Min-Max Bereich. Es wurden jeweils 10 Versuche mit identischer Initialisierung für alle Verfahren durchgeführt. Dis Sortierung der Klassen nach maximaler Evidenz ($\mathcal{E}+\mathcal{K}$) schneidet signifikant besser ab als die Sortierung nach Unabhängigkeit ($\mathcal{U}+\mathcal{K}$) (t-Test bei Schwelle $t_{0.95;10} = 1.82$). Kombiniert man die Strategie $\mathcal{E}$ mit $\mathcal{B}$ und/oder $\mathcal{K}$, dann erreicht die Variante $\mathcal{E}+\mathcal{K}+\mathcal{B}$ wiederum einen signifikant geringeren Testfehler.

Anschließend wurde die Strategie $\mathcal{E}$ mit bzw. ohne die zusätzlichen Optionen $\mathcal{B}$ und $\mathcal{K}$ getestet. Die Kombination aller drei Strategien führt nochmals zu einer signifikanten Verbesserung. Für die im folgenden zu vergleichenden Mutationsoperatoren wird die Strategie $\mathcal{E}+\mathcal{K}+\mathcal{B}$ zugrunde gelegt.

6.1.2 Vergleich der zusätzlichen Mutationsoperatoren

Neben der Mutation von Merkmalen und versteckten Neuronen lassen sich noch weitere sinnvolle Mutationsoperatoren definieren, die die Muster und die Gewichte eines Netzes betreffen. Im Detail leisten die Operatoren folgendes:

$\mathcal{PSEL}$ Pattern-Selection. Um den ganz speziellen Einfluß der Wahl der Trainingsmenge zu dämpfen, mittelt man über mehrere Teilmengen. Standardmäßig werden für jedes Netz zufällig 90% der Daten ausgewählt.

$\mathcal{NEW}$ Bei festgehaltener Topologie werden die Gewichte neu initialisiert und die Muster neu gewählt. Dadurch wird ein neuer Suchpunkt im Gewichtsraum definiert.

$\mathcal{MUTP}$ Mit Wahrscheinlichkeit P_{Flip} werden Muster aus der individuellen Trainingsmenge entfernt bzw. neue hinzugefügt. Die Größe der Menge wird dabei konstant gehalten. P_{Flip} beträgt im allgemeinen 0.01.

$\mathcal{JOG}$ Der Gewichtsvektor wird mit Zufallszahlen verrauscht. Dadurch werden die Nachkommen im Gewichtsraum um die Eltern gestreut.

Abbildung 6.4 zeigt den Testfehler des Komitees bei Verwendung der zusätzlichen Mutationsoperatoren. Es wurden jeweils 10 Versuche ausgehend von derselben Initialisierung durchgeführt. Die besten Werte erreicht man bei Verwendung von $\mathcal{PSEL}$ und $\mathcal{NEW}$. Die Integration über mehrere Teilmengen war eine Voraussetzung für die Dekomposition des Komiteefehlers. Sie wird deswegen in jedem Fall verwendet. Die durch den Operator $\mathcal{NEW}$ definierten neuen Suchpunkte können potentiell neue Klassen hervorbringen, wenn die Suchpunkte nur unabhängig genug sind und keine zu niedrige Evidenz aufweisen.

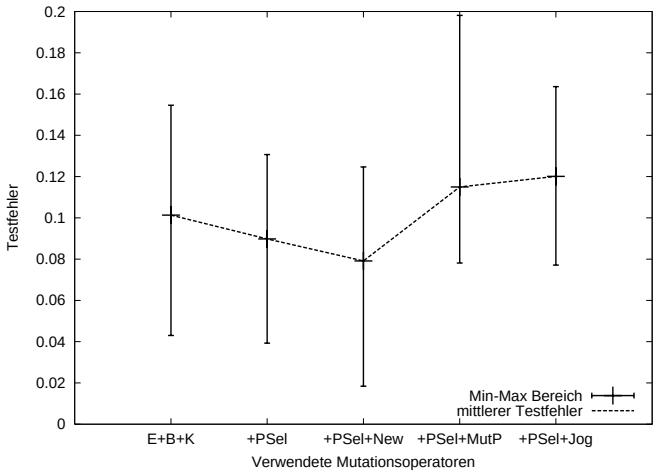


Abbildung 6.4: Vergleich der zusätzlichen Mutationsoperatoren. Die Abbildung zeigt den mittleren Komiteefehler sowie den Min-Max Bereich. Es wurden jeweils 10 Versuche mit identischer Initialisierung für alle Verfahren durchgeführt. Durch Verwendung von $\mathcal{PSEL}$ und $\mathcal{NEW}$ lassen sich signifikante Verbesserungen erreichen. Dagegen bewirken die anderen beiden Operatoren einen ungünstigen Verlauf der Evolution.

Die direkte Mutation der Mustermengen und das Verrauschen des Gewichtsvektors haben dagegen einen negativen Einfluß auf das Ergebnis. Bei der Mustermutation liegt ein Grund für die Verschlechterung in einem möglichen Konflikt mit dem Bayes'schen Lernen. Kommen neue Muster zur Trainingsmenge hinzu und fallen andere weg, dann ändert sich der Fehler ganz erheblich. Abbildung 2.9 illustrierte, wie das Bayes'sche Verfahren eine a priori Wahrscheinlichkeit in eine a posteriori Wahrscheinlichkeit überführt. Betrachtet man zwei identische Netze und ändert bei einem die Mustermenge, dann wird das Bayes'sche Verfahren bei diesem Netz aufgrund des nun größeren Fehlers die Regularisierung verstärken. Man beachte, daß sich der Prior nicht geändert hat. Das Netz wird jetzt überregularisiert. Im ungünstigsten Fall bildet dieser Prozeß der Überregularisierung eine Rückkopplung, die sich selbst verstärkt. Eine Mustermutation sollte deshalb mit einer Neuinitialisierung der Hyperparameter einhergehen. Aus diesem Grund war der $\mathcal{NEW}$-Operator erfolgreicher als die beiden anderen.

6.2 Problemlösung durch Komitees

In diesem Abschnitt soll das integrierte Konzept auf die Problemstellungen, für die im vorigen Kapitel bereits die evolutive Modelloptimierung durchgeführt wurde, angewendet und mit anderen Methoden zur Komiteebildung verglichen werden.

Es wird sich zeigen, daß Boosting und Bagging besonders für Regressionsprobleme deutlich schlechter abschneiden als die evolutive Optimierung und die darauf aufbauende Komiteebildung aus unabhängigen Netzen. Ein wesentlicher Grund dafür ist, daß die beiden Methoden das zugrundeliegende Modell nicht optimieren, sondern nur auf der Musterebene operieren.

Die Ergebnisse in Kapitel 5 haben aber deutlich belegt, daß diese Optimierung wichtig ist, selbst wenn das zugrundeliegende Modell bereits regularisiert wird. Für Boosting geht man davon aus, daß die Komplexität des Modells durch Kreuzvalidierung optimiert ist (Drucker, 1999). Es ist an dieser Stelle nicht klar, ob man nur die Parameter des singulären Modells möglichst gut einstellen soll oder ob der Komiteebildungsprozeß durch Kreuzvalidierung zu optimieren ist. Unabhängig davon sind beide Vorgehensweisen in jedem Fall suboptimal.

Für das Bagging-Verfahren wurden jeweils über 30 Netze gemittelt. Boosting ist nach (Freund & Schapire, 1996, Drucker, 1999) implementiert. Die Netze wurden dabei linear zu einem Komitee kombiniert.

Im folgenden werden zuerst das Sinus-Regressionsproblem und Klassifikationsproblem aus Kapitel 5.2 betrachtet, dann die vier bekannten Benchmarkprobleme (Diabetes, Krebs, Add10 und Schilddrüsen). Am Ende gehe ich auf Probleme mit starkem Rauschanteil ein und werde an diesen auch demonstrieren, in welchen Fällen die adaptive Gewichtung der Komiteemitglieder weitere Leistungssteigerungen erwarten läßt.

6.2.1 Sinus-Regression und Klassifikation

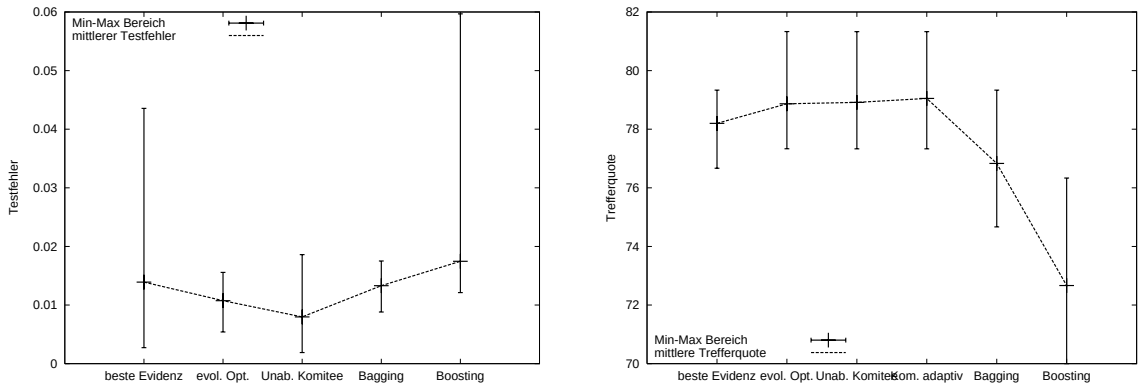
Das Sinus-Regressionsproblem zeichnet sich neben den verrauschten Zielwerten noch durch die fehlenden Daten im Intervall $[0;2]$ aus. In diesem Bereich ist es für die Netze schwierig, die Funktion richtig zu schätzen. Hier hilft die Komiteebildung weiter. Sie mittelt über mehrere mögliche Erklärungen. Dadurch kann der Fehler gegenüber der reinen Modelloptimierung noch signifikant verringert werden (Abbildung 6.5a). Bagging erreicht etwa die gleiche Leistung wie die Auswahl des Netzes mit der höchsten Evidenz, die Varianz ist aber deutlich geringer. Die Resultate für Boosting sind signifikant schlechter, obwohl in diesem Fall das zugrundeliegende Modell fast optimale Komplexität hatte. Das Rauschen in den Daten wird von Boosting nicht richtig erkannt.

Für das Sinus-Klassifikationsproblem waren die Trainingsmuster über den ganzen Eingaberaum verteilt. Der erzeugende Prozeß kann deswegen gut erkannt werden. Die Komiteebildung bringt in diesem Fall nur geringfügige Vorteile (Abbildung 6.5b). Bagging und Boosting bewirken sogar eine Verschlechterung. Neben der fehlenden Optimierung der versteckten Schicht kann in diesen Fällen auch die mehrfache Auswahl eines Musters durch den Bootstrap-Algorithmus zu Ausreißern im Lernverhalten führen, da die Mustermengen klein sind. Für das Regressionsproblem schätzt das Bayes'sche Verfahren in diesem Fall zumindest die Varianz des Rauschens falsch ein.

6.2.2 Anwendung auf vier bekannte Probleme

Die Berechnung des empirischen Zusammenhangs zwischen Evidenz und Netzkomplexität, gemessen an der Zahl der Merkmale und versteckten Neuronen, zeigte, daß es für die vier Datensätze klar abgegrenzte Bereiche gibt, in denen eine hohe Evidenz und damit eine gute Generalisierungsleistung zu erwarten ist.

Verwendet man die Evidenz als Fitneßwert in einem evolutionären Suchverfahren, dann findet man systematisch die Netze, deren Topologie in die Bereiche maximaler Evidenz



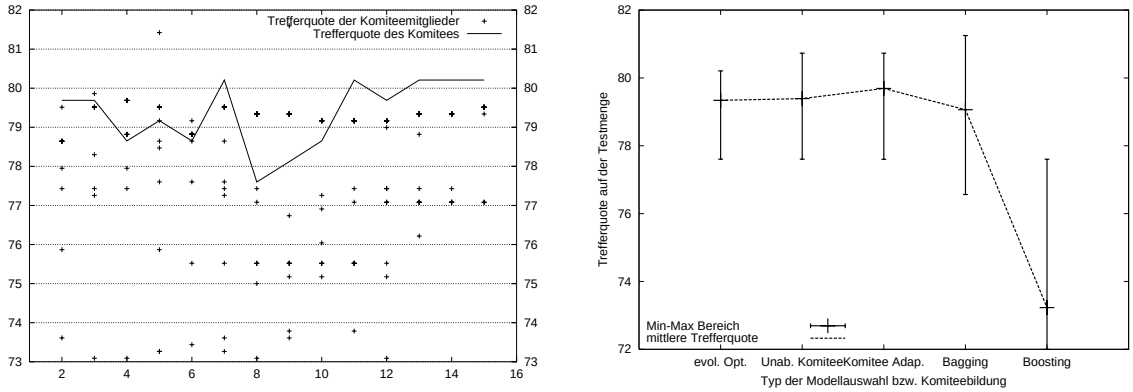
(a) (b)

Abbildung 6.5: Die Abbildung vergleicht die Ergebnisse der verschiedenen Methoden der Komiteebildung bzw. der Modelloptimierung für die beiden Probleme aus Kapitel 5.1 (a) Sinus-Regressionsproblem. Das Komitee aus unabhängigen Netzen weist einen signifikant geringeren Fehler auf als alle anderen Verfahren. Bagging schneidet etwa so gut ab wie die Auswahl des Netzes mit der höchsten Evidenz aus einer Versuchsreihe. Boosting ist signifikant schlechter als alle anderen Verfahren. (b) Sinus-Klassifikationsproblem. Hier wird der zugrundeliegende Prozeß bereits durch die evolutiv optimierten Modelle sehr gut approximiert. Das Komitee ist nur geringfügig besser. Die Gewichtung der Komiteemitglieder mit ihren Verlustwahrscheinlichkeiten bringt noch eine leichte Steigerung. Bagging und Boosting schneiden signifikant schlechter ab.

fallen. Es zeigte sich aber auch, daß man im allgemeinen bei mehrfachen Versuchen nicht zur selben Lösung gelangt. Das Beispiel in der Einführung dieses Kapitels (Abbildung 6.2) belegte am Beispiel des Sinus-Regressionsproblems, daß verschiedenartige Modelle eine ähnliche Evidenz haben können, wenn die Daten verrauscht sind. Für die folgenden Beispiele kann man von der Komiteebildung erwarten, daß sie durch Integration über verschiedene Erklärungsmöglichkeiten die Generalisierung steigert.

Die Unterscheidung der Netze auf Basis der Anzahl an Merkmalen und versteckten Neuronen ist in diesem Fall weniger aussagekräftig als für die evolutive Optimierung, da jedes Netz mit einer Teilmenge aller Trainingsdaten trainiert wurde. Das heißt, Netze die in einem Punkt lokalisiert sind, können sich außer in ihrer Merkmalskombination auch noch in ihren Trainingsdaten unterscheiden. Die Abbildungen 6.6 bis 6.9 zeigen für die vier Probleme auf der linken Seite jeweils den Verlauf der Evolution an einem Beispiel. Es ist der Fehler der Netze, der Fehler der Komiteemitglieder und der Komiteefehler über die Generationen aufgetragen. Am Anfang der Suche kommt es durchaus auch vor, daß die Leistung des Komitees abnimmt, bis sich die Netze in Bereichen höherer Evidenz in der Population durchsetzen. Die Gründe für diesen Verlauf liegen in der Verzahnung der Suche mit dem Bayes'schen Lernen. Es dauert einige Generationen, bis die Hyperparameter der Netze gut eingestellt und Netze mit geringer Evidenz aus der Population verdrängt sind. Danach kann es durch Mutationen immer wieder dazu kommen, daß sich Klassen mit niedriger Evidenz ausbilden, z.B. dann, wenn zufällig viele ungünstige Mutationen gleichzeitig auftreten. Durch zunehmendes Training nimmt die Evidenz dieser Netze entweder zu oder sie werden durch die Reduktion der Population auf die Zahl der Eltern am Ende jeder Generation nach und nach wieder eliminiert. Während der Suche ist dieser Verlauf durchaus wünschenswert, da so auch Täler der Evidenz durchwandert werden können.

Umgekehrt ist es wichtig, wenn eine feste Zahl an Generationen gesucht wird, rechtzeitig keine Mutationen mehr durchzuführen. In den Beispielen wurde in der 13. Generation zum letzten Mal mutiert. Damit soll sichergestellt werden, daß am Ende nur Netze in Bereichen hoher Evidenz vorliegen.



(a) (b)

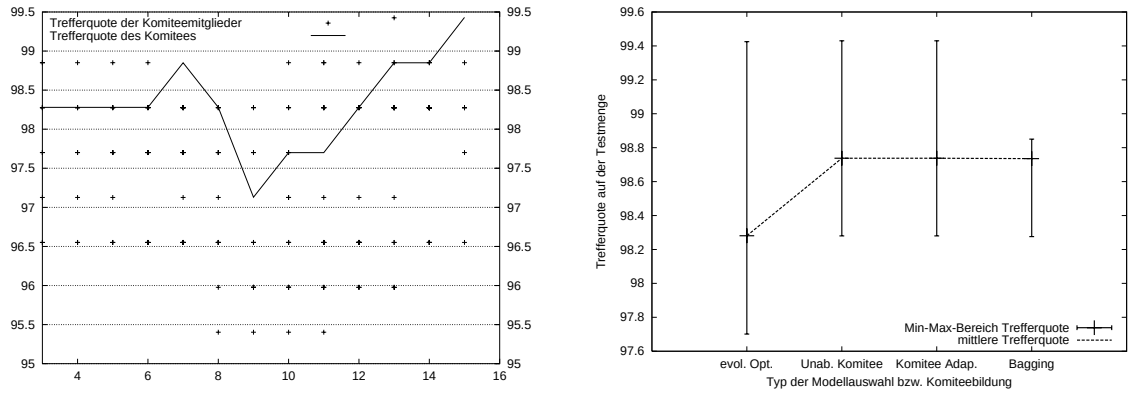
Abbildung 6.6: Diabetes-Klassifikation. Verlauf der Suche nach unabhängigen Netzen und Vergleich der Ergebnisse der verschiedenen Methoden. (a) Aufgetragen ist die Trefferquote der Komiteemitglider sowie des Komitees. Letztere liegt immer im oberen Bereich bzw. über der Leistung der einzelnen Netze. (b) Vergleich der verschiedenen Methoden. Evolutive Optimierung, Komitees aus unabhängigen Netzen und Bagging schneiden etwa gleich ab. Die mittlere Trefferquote liegt zwischen 79.1% und 79.4%, Boosting ist signifikant schlechter.

Für das Diabetes-Problem liegt die Trefferquote des Komitees immer im oberen Bereich der Leistung der individuellen Mitglieder, gegen Ende sogar deutlich darüber (Abbildung 6.6a). Der Vergleich der verschiedenen Methoden in Abbildung 6.6b zeigt, daß alle (bis auf Boosting) zu ähnlichen Ergebnissen führen. Für Boosting kann man ein deutliches Overfitting erkennen. Die Leistung weicht signifikant von allen anderen ab (gemäß t-Test bei einer Schwelle von $t_{0.95;10} = 1.82$). Die hier erreichte Leistung entspricht den Werten, die auch aus der Literatur bekannt sind (Freund & Schapire, 1996, Breiman, 1999).

Gewichtet man die Komiteemitglieder mit ihrer Verlustwahrscheinlichkeit, dann läßt sich die Leistung etwas steigern. Insgesamt kann man festhalten, daß die Diabetes-Klassifikation ein Problem mit hohem Rauschanteil ist, dessen zugrundeliegende Funktion nur einen geringen nicht-linearen Anteil hat. Außer von Boosting wird das von allen Verfahren korrekt erkannt.

Bei der Klassifikation von Brustkrebs läßt sich durch die Komiteebildung eine signifikante Reduktion der Fehlklassifikationen erreichen. In Kapitel 5 wurde angemerkt, daß die Eingabedaten ordinale Struktur haben, d.h. sie liegen auf einem Gitter angeordnet. Damit läßt sich plausibel begründen, warum die Mittelung über mehrere Netze besser abschneidet. Für ein einzelnes Netz ist es schwer, eine Trennlinie zwischen den Gitterpunkten zu finden, ohne daß eine hochgradig nicht-lineare Funktion gelernt wird. In diesem Fall besteht dann aber die Gefahr, daß in Bereichen mit wenig Daten starke Schwankungen der Funktion auftreten, die zu einer schlechten Leistung führen. In Abschnitt 6.3 wird dies an einem Beispiel deutlich werden.

Bemerkenswert ist weiterhin, daß die adaptive Gewichtung mit Verlustwahrscheinlichkeiten keine Verbesserung bringt. Auch hier ist die ordinale Struktur der Grund dafür, daß der



(a) (b)

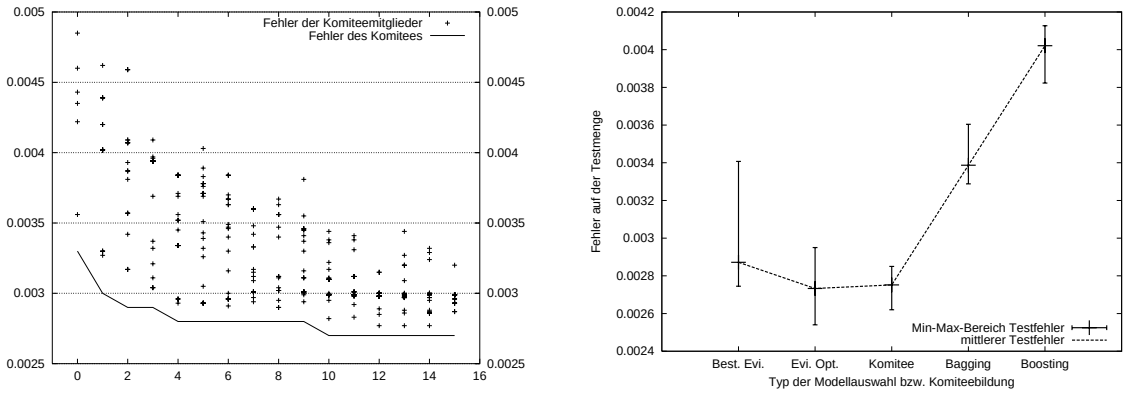
Abbildung 6.7: Krebs-Klassifikation. Verlauf der Suche nach unabhängigen Netzen und Vergleich der Ergebnisse der verschiedenen Methoden. (a) Die Trefferquote des Komitees verläuft wieder im oberen Bereich der Leistung der einzelnen Netze und liegt am Ende wieder darüber. (b) Die mittlere Leistung der Methoden zur Komiteebildung ist signifikant besser als die Auswahl eines einzelnen Netzes durch die evolutive Optimierung. Der Grund dafür liegt in der Gitterstruktur der Eingabedaten. Die Werte für Boosting liegen weit unter 97% und sind nicht eingetragen.

Eingaberaum gut mit Mustern abgedeckt ist und so keine Bereiche hoher Varianz auftreten. Der wesentliche Leistungsgewinn der Komitees rührt daher, daß durch Übereinanderlegen mehrerer Funktionen die zackigen Verläufe der Klassifikationsgrenze in der Gitterstruktur besser approximiert werden können als durch einzelne Netze.

Die Werte für Boosting sind nicht in der Grafik angegeben. In keinem Experiment konnten bei diesem Problem mehr als 94% erreicht werden.

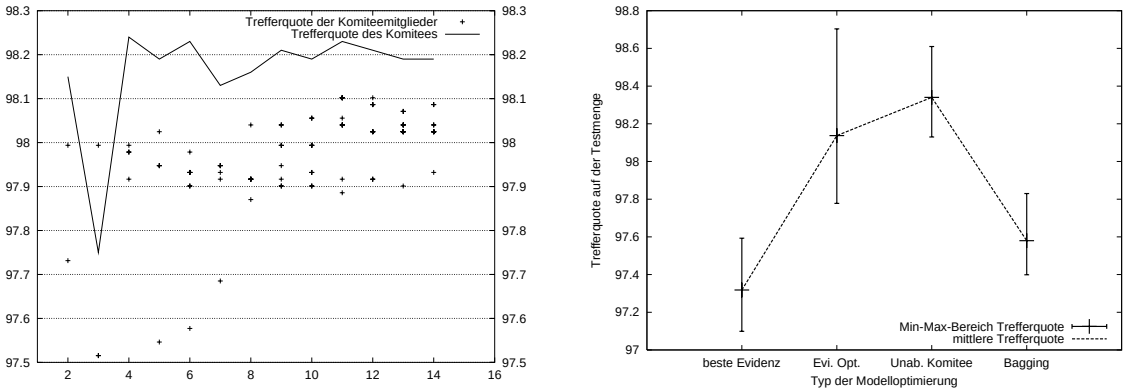
Für das Add10-Problem läßt sich schön erkennen, wie sich die Evolution zunehmend in Bereiche hoher Evidenz bewegt, aber auch immer wieder durch Mutation neue Netze mit zunächst geringer Leistung auftreten. Der Komiteefehler stabilisiert sich schnell auf niedrigem Niveau immer unterhalb des besten Netzes. Abbildung 6.8b zeigt aber auch, daß die mittlere Leistung gegenüber der evolutiven Optimierung nicht zunimmt, nur die Varianz wird kleiner. Die evolutive Optimierung fokussiert sich wesentlich stärker in einem einzigen Bereich hoher Evidenz und findet so teilweise noch bessere Netze, teilweise aber auch schlechtere. Für dieses Problem zeigt sich ebenfalls, daß die fehlende Merkmalsselektion und Topologieoptimierung für Bagging und Boosting zu signifikant schlechteren Ergebnissen führt (gemäß t-Test bei einer Schwelle von $t_{0.95;10} = 1.82$). Der Generalisierungsfehler ist um ein Sechstel bzw. ein Drittel größer.

Ein sehr schönes Ergebnis ergibt sich für das Schilddrüsen-Problem. Auch hier zeigt sich der Vorteil des integrierten Konzepts. Die Trefferquote der Komiteemitglieder liegt ab der 8. Generation über der maximalen Leistung, die mit allen 21 Merkmalen erreichbar ist (vgl. Abbildung 1.4). Die Trefferquote des Komitees liegt deutlich über der Leistung von Bagging oder den Netzen mit der höchsten Evidenz eines Versuchs. Durch die Suche nach unabhängigen Netzen läßt sich die Zahl der Fehlklassifikationen auf 1.65% reduzieren gegenüber 1.8% für die evolutive Modelloptimierung. Dies ist umso bemerkenswerter, da auch hier die Evidenz ein nahezu perfekter Indikator der Generalisierungsleistung war.



(a) (b)

Abbildung 6.8: Add10-Regression. Verlauf der Suche nach unabhängigen Netzen und Vergleich der Ergebnisse der verschiedenen Methoden. (a) Der Testfehler des Komitees stabilisiert sich schnell auf niedrigem Niveau. Er liegt immer unterhalb des kleinsten Fehlers eines Komiteemitglieds. (b) Vergleich der Methoden zur Komiteebildung. Die Fehler für Bagging und Boosting sind signifikant größer. Die unabhängigen Komitees und die evolutive Optimierung erreichen das gleiche Ergebnis. Durch die starke Korrelation der Evidenz mit dem Testfehler ist die evolutive Optimierung eines einzelnen Modells nicht zu schlagen.



(a) (b)

Abbildung 6.9: Schilddrüsen-Klassifikation. Verlauf der Suche nach unabhängigen Netzen und Vergleich der Ergebnisse der verschiedenen Methoden. (a) Die Trefferquote des Komitees verläuft bis auf den kurzen Einbruch am Anfang immer oberhalb der Leistung der einzelnen Netze. Die Suche stabilisiert sich schnell in Bereichen hoher Evidenz. Die Leistung des Komitees pendelt um 98.2%. (b) Vergleich der Methoden zur Komiteebildung. Die Leistung für Bagging ist signifikant schlechter als das Komitee aus unabhängigen Netzen oder die evolutive Modelloptimierung. Der Wert liegt aber noch über dem Maximalwert, den ein Netz mit 21 Merkmalen erreicht hat. Der Unterschied zwischen der evolutiven Optimierung und dem Komitee aus unabhängigen Netzen ist nicht signifikant.

Insgesamt bleibt festzuhalten, daß mit dem integrierten Ansatz immer gleich gute oder bessere Modelle im Vergleich zu allen anderen Methoden gefunden werden. In manchen Fällen läßt sich durch adaptive Gewichtung der Komiteemitglieder die Leistung zusätzlich verbessern. Im folgenden werden noch die beiden Probleme mit starkem Rauschanteil untersucht und im nächsten Abschnitt wird auf die Frage eingegangen, wann die Gewichtung der Komiteemitglieder auf Basis der Konfidenzwerte weitere Leistungssteigerungen erwarten läßt.

6.2.3 Probleme mit hohem Rauschanteil

Die Abbildungen 5.26 und 6.2 haben gezeigt, daß bei zunehmendem Rauschen in den Daten verschiedenartige, aber durchaus berechtigte Erklärungen des zugrundeliegenden Prozesses gefunden werden. In diesem Fall ist es sowohl für die evolutive Optimierung als auch für Bagging bzw. Boosting schwieriger, eine sehr gute Generalisierung zu erreichen. Im ersten Fall konvergiert das Verfahren immer gegen eine Lösung, was sich an der hohen Varianz zeigt (Abbildung 6.10). Bagging integriert über mehrere Netze, die mehr oder weniger gut sind. Für das Boosting-Verfahren wirkt sich hier der bekannte Nachteil gravierend aus, daß Ausreißer umso häufiger ausgewählt werden und die Komiteebildung damit schlechter wird (Rätsch *et al.*, 1998). Insgesamt zeigt sich für beide Probleme, daß die integrierte Optimierung eine gute Gesamtlösung aus den verschiedenen Erklärungsmöglichkeiten synthetisiert. Der Fehler liegt für beide Probleme signifikant unter dem aller anderen Methoden (Abbildung 6.10).

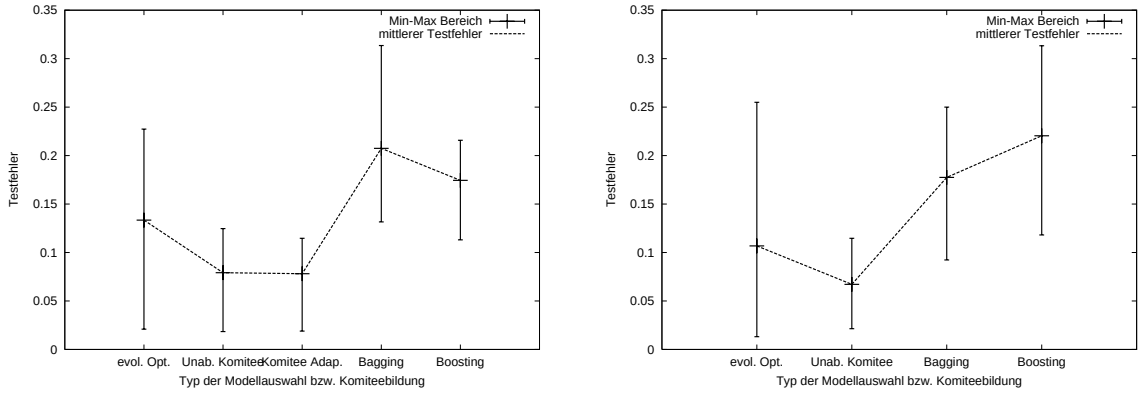
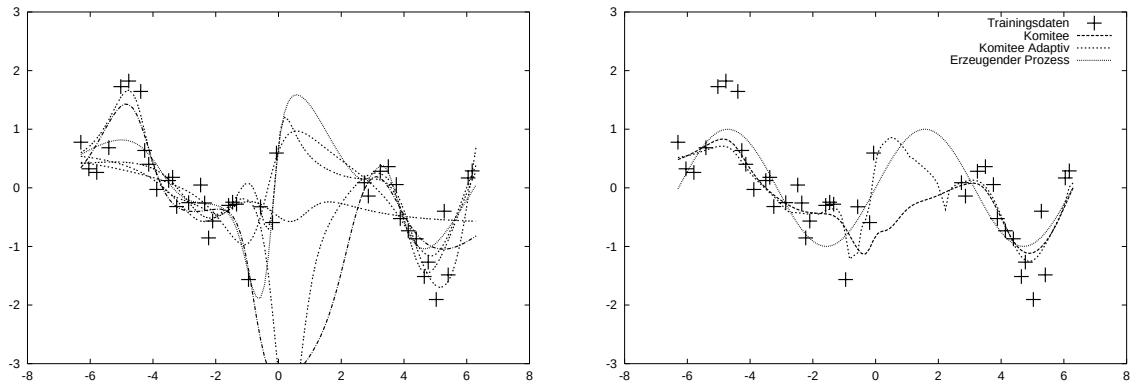


Abbildung 6.10: Vergleich der Ergebnisse der verschiedenen Methoden für das Sinus-Regressionsproblem mit starkem Rauschanteil und gleichverteiltem Rauschen. Das Komitee aus unabhängigen Netzen erreicht signifikante Verbesserungen für beide Datensätze. (a) Testfehler für Daten aus Abbildung 1.1b. Die Komiteebildung aus unabhängigen Netzen erreicht einen um die Hälfte niedrigeren Testfehler im Vergleich zur evolutiven Optimierung der Evidenz. Der Fehler für Bagging und Boosting ist mehr als doppelt so groß und auch signifikant schlechter als die evolutive Modelloptimierung. (b) Testfehler für Sinus-Regression mit Rauschen aus Abbildung 5.25b. Die Ergebnisse sind ähnlich wie für die linke Abbildung. Offenbar ist vor allem die Varianz des Rauschens von Bedeutung und weniger die genaue Form der Verteilung.

Die adaptive Gewichtung bringt kaum oder nur geringfügige Verbesserungen. Im nächsten Abschnitt wird die Methode der adaptiven Gewichtung genauer beleuchtet und durch grafische Aufbereitung auch die Wirkungsweise der integrierten Optimierung besser verständlich.

6.3 Adaptive Gewichtung der Komiteemitglieder

Die Abbildung 6.11 zeigt die Ausgabefunktion von sechs trainierten Netzen und die des resultierenden Komitees. Dieser Fall entspricht der Wirkungsweise von Bagging. Weiterhin ist auch das Ergebnis der adaptiven Gewichtung dieser Komiteemitglieder mittels ihrer Konfindenzwerte angegeben. Jede einzelne Ausgabefunktion ist mit ihrem zugehörigen Konfindenzbereich in Abbildung 6.12 dargestellt. Für mehrere Netze kann man erkennen, daß



(a) (b)

Abbildung 6.11: Die Abbildung zeigt ein Komitee aus sechs Netzen, das mit den Sinus-Regressionsdaten trainiert wurde. (a) Die sechs Ausgabefunktionen der Komiteemitglieder. Die Funktionen unterscheiden sich stark, insbesondere im Intervall $[0;2]$. Die Kreuze markieren die Trainingsdaten. (b) Ausgabefunktion des Komitees aus den sechs Netzen und des adaptiv gewichteten Komitees. Zum Vergleich ist der erzeugende Prozeß eingezeichnet. Die Gewichtung mit den Konfidenzwerten verbessert die Approximation in den Bereichen hoher Varianz.

in dem Bereich, in dem wenig Daten liegen, die Varianz deutlich zunimmt. In diesem Fall bringt die adaptive Gewichtung eine deutliche Reduktion des Fehlers. Die Ausgabe von Netzen mit hoher Varianz findet bei diesen Mustern keine Berücksichtigung, was zu einer verbesserten Approximation des Prozesses führt. Insbesondere im Bereich zwischen 0 und 2 fällt der Fehler geringer aus.

Das Ergebnis der Suche nach unabhängigen Netzen ist in Abbildung 6.13 gezeigt. In der rechten Abbildung ist die Ausgabefunktion des Komitees und des adaptiv gewichteten Komitees dargestellt. Im Vergleich zu Abbildung 6.11 ist der Verlauf wesentlich glatter. Betrachtet man die Daten genauer, dann ist diese Lösung sogar bei dieser Datenlage mehr gerechtfertigt als der erzeugende Prozeß. Die Daten liegen in der rechten Hälfte eher im unteren Bereich, was erklärt, warum der Bogen hier weiter unten verläuft. Die beiden mittleren Bögen der Sinusfunktion sind durch die Daten nicht ausreichend gestützt. Deswegen wäre es unvernünftig, diese bei der gegebenen Datenlage zu approximieren. Der Grund dafür, daß die adaptive Gewichtung keine Verbesserung bringt, ist anhand der Ausgabefunktion der einzelnen Netze und ihrer Varianz zu erkennen. Abbildung 6.14 zeigt, daß die durch die Evolution gefundenen Netze mit hoher Evidenz sehr glatte Lösungen repräsentieren. Eine hohe Evidenz wird offenbar dann erreicht, wenn der Fehler klein ist und die Funktion keine überflüssigen Schwankungen enthält. Weiterhin bemerkenswert ist, daß fünf der Netze im Intervall $[0;2]$ die Funktion symmetrisch zur linken Hälfte approximieren. Eine symmetrische Funktion hat sicherlich eine niedrigere Komplexität und damit eine höhere Evidenz als eine nicht-symmetrische. Ebenso haben alle Netze die Varianz, die dem Rauschen in den Daten zugrundeliegt, genau erkannt. Sie verläuft über den ganzen Bereich gleichmäßig. Da die Varianz für alle Netze nahezu gleich ist, läßt sich das evolutiv gefundene Komitee kaum noch verbessern.

Es bleibt hier festzuhalten, daß Netze mit hoher Evidenz bei diesem Beispiel sehr plausiblen Lösungen entsprechen. Durch die Integration über alle diese Lösungen wird der erzeugende

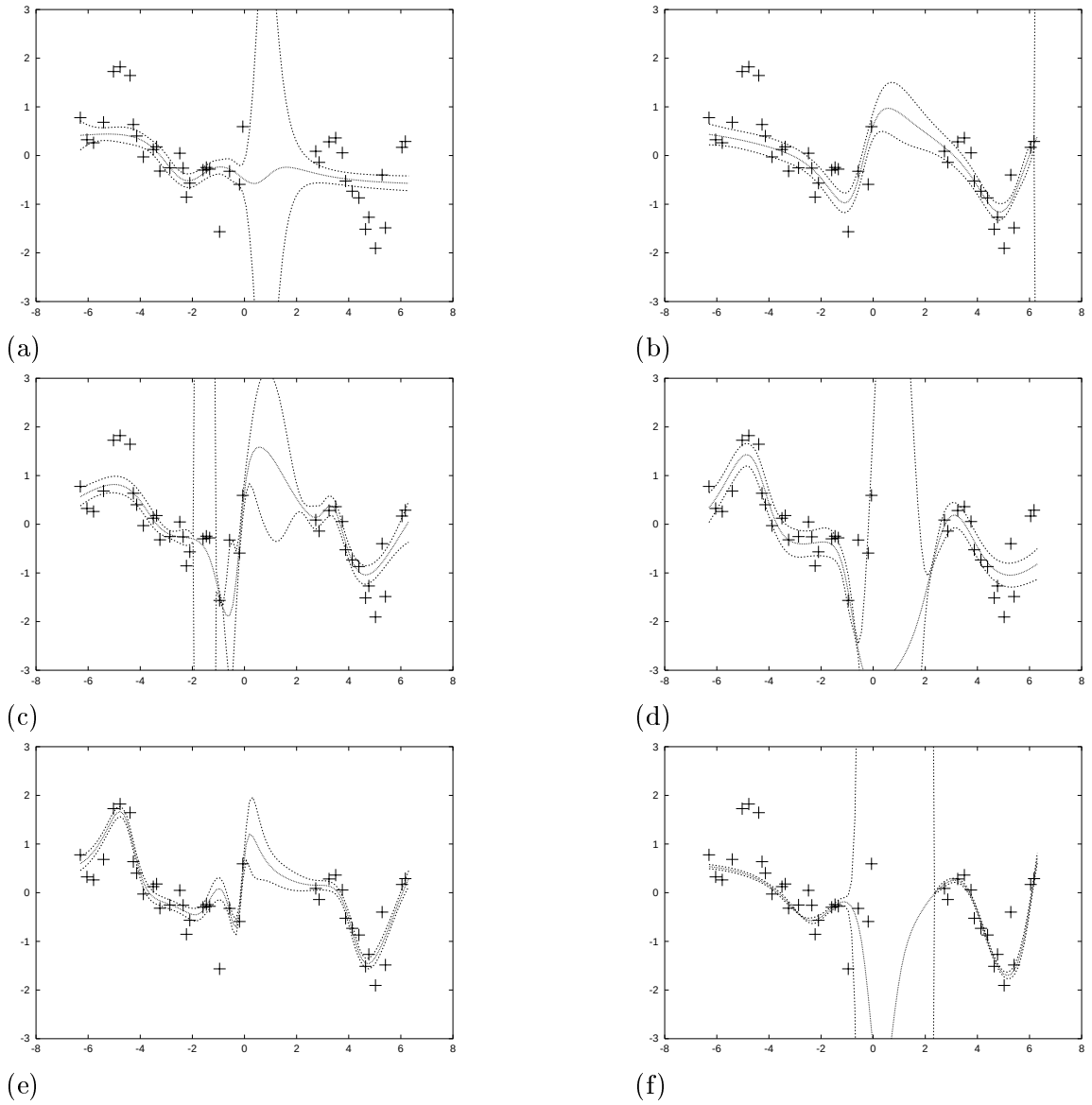
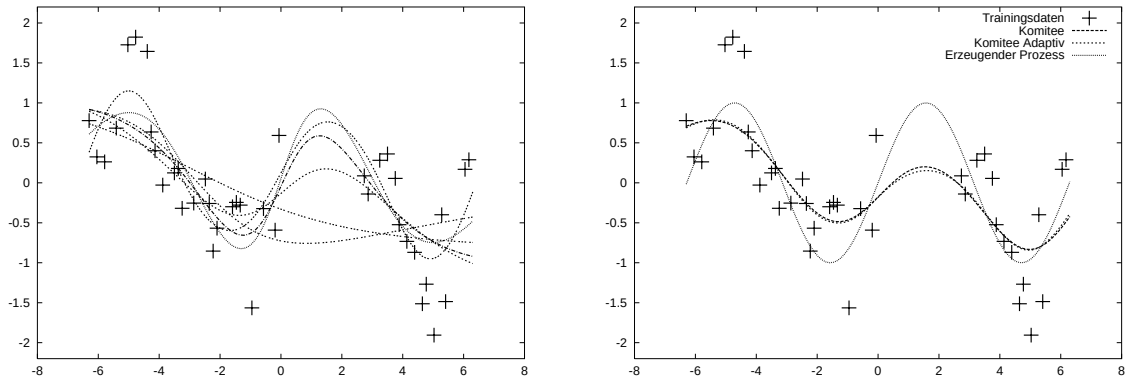


Abbildung 6.12: Die Abbildung zeigt unter (a)-(f) die Ausgabefunktionen der sechs Netze aus Abbildung 6.11a. Zusätzlich ist für jedes Netz die Varianz gemäß Gleichung (2.43) eingezeichnet. Es ist schön zu erkennen, daß sich manche Netze im Bereich zwischen 0 und 2 sehr unsicher sind und das anhand ihrer Varianz erkennen lassen. Ansonsten verläuft die Varianz eng um die Funktion. Tatsächlich wird damit das Rauschen unterschätzt.

Prozeß, soweit es die Daten zulassen, sehr schön approximiert.

6.4 Zusammenfassung und Bewertung der Ergebnisse

Das Ziel dieses Kapitels war es, die Evolution unabhängiger Netze zur Komiteebildung an Beispielen zu untersuchen und die Ergebnisse mit anderen Methoden zu vergleichen. Zu



(a) (b)

Abbildung 6.13: Die Abbildung zeigt ebenfalls für das Sinus-Regressionsproblem ein Komitee aus sechs Netzen, das durch die integrierte Optimierung gefunden wurde. (a) Die Ausgabefunktionen der sechs Komiteemitglieder zeigen ein glattes, symmetrisches Verhalten. (b) Ausgabefunktion des Komitees und des adaptiv gewichteten Komitees. Die Daten werden sehr plausibel approximiert. Die beiden Varianten unterscheiden sich nur geringfügig. Daß die mittleren Bögen der Sinusfunktion nicht genau approximiert werden, liegt daran, daß dieser Verlauf durch die Trainingsdaten nicht gestützt ist. Es wäre nicht vernünftig, wenn das Modell das lernen würde.

Beginn wurde gezeigt, daß die Bildung eines Komitees auf Basis des in Kapitel 3 entwickelten Verfahrens ein Maximum an Generalisierungsfähigkeit erreicht. Der Inhalt des ersten Abschnitts drehte sich dann um die Frage, wie die Clusterung der Netze und die Optimierung der Evidenz in einem evolutionären Prozeß verzahnt werden können. Dazu wurden mehrere Selektionsstrategien untersucht. Die besten Ergebnisse erreichte man durch Sortierung der Klassen nach ihrer Evidenz (Strategie $\mathcal{E}$) in Kombination mit der Bevorzugung der in jeder Generation ausgewählten Komiteemiglieder (Strategie $\mathcal{B}$) und einer gedämpften Veränderung der Klassenzahl von Generation zu Generation (Strategie $\mathcal{K}$).

Aufbauend auf diesem Ergebnis wurde untersucht, welche der zusätzlichen Mutationsoperatoren die Evolution günstig beeinflussen. Hier zeigte sich, daß neben dem Training auf Teilmengen (Operator $\mathcal{PSEL}$) die Definition neuer Suchpunkte (Operator $\mathcal{NEW}$) zu signifikant geringeren Fehlern für das Sinus-Regressionsproblem mit hohem Rauschanteil führte. Das Problem wurde deshalb zur Einstellung der Parameter gewählt, da sich aufgrund des Rauschanteils am ehesten signifikante Unterschiede zeigen. Mit dem so eingestellten Evolutionsverfahren wurden dann alle andere Probleme untersucht.

Das integrierte Konzept führt immer zu gleichwertigen oder besseren Lösungen als alle anderen Verfahren. Optimiert man nur die Evidenz (evolutive Modelloptimierung), dann erreicht man für die Krebs-Klassifikation und für die beiden Probleme mit hohem Rauschanteil eine signifikant geringere Leistung. Für das Diabetes- und Add10-Problem sind die Ergebnisse etwa gleich. Bei der Schilddrüsen-Klassifikation schneidet die Komiteebildung wieder etwas besser ab. Der Unterschied ist allerdings nicht signifikant.

Bagging schnitt dann vergleichbar gut ab, wenn die zu approximierende Funktion keinen starken nicht-linearen Anteil hatte, z.B. bei der Diabetes- und Krebs-Klassifikation. Für die anderen Beispiele (Add10, Schilddrüse, Sinus-Regression und -Klassifikation) war die Generalisierungsleistung signifikant geringer.

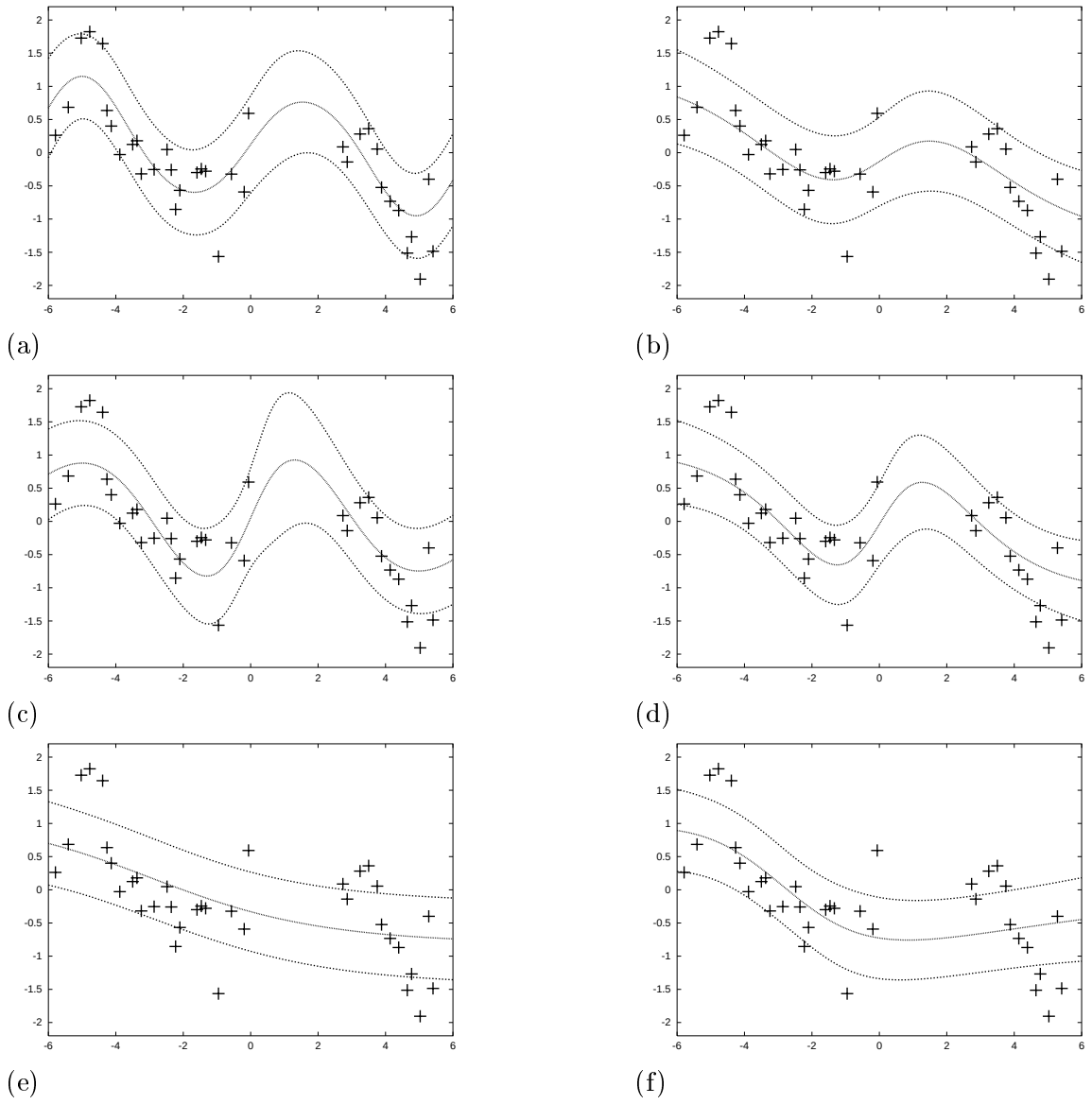


Abbildung 6.14: Die Abbildung zeigt (a)-(f) die Ausgabefunktionen der sechs Netze aus Abbildung 6.13a. Zusätzlich ist für jedes Netz die Varianz eingezeichnet. Die Netze zeigen alle einen symmetrischen und glatten Verlauf. Insbesondere schätzen sie die Varianz des Rauschens gut ein. Aus diesem Grund bringt die adaptive Gewichtung hier kaum eine Verbesserung.

Der Boosting-Algorithmus schnitt in Kombination mit dem Bayes'schen Lernverfahren deutlich schlechter ab als alle anderen Methoden. Ursprünglich wurde das Verfahren entwickelt, um für *Weak Learner* durch Komiteebildung zu besseren Lösungen zu gelangen. Das Bayes'sche Verfahren ist dagegen eher eine starke Lernmethode, die durch Boosting nicht verbessert werden kann. Vielmehr stören sich die Methoden gegenseitig. Wenn durch Boosting die Auswahl der Trainingsmenge verändert wird, dann paßt das Bayes'sche Verfahren die Hyperparameter anders an. Bei den Regressionsproblemen führt die stärkere Gewichtung von Ausreißern dazu, daß die Varianz des Rauschens überschätzt und das Modell stärker regularisiert

wird, Drucker bemerkt zu Recht, daß Boosting voraussetzt, daß die Generalisierungsleistung von dem zugrundeliegenden Lernverfahren und der Regularisierung durch Kreuzvalidierung abhängt (Drucker, 1999). Offenbar setzt das Verfahren voraus, daß man die Regularisierung durch Kreuzvalidierung bestimmt und diese dann nicht mehr ändert. Dies ist in jedem Fall suboptimal. Auch hängt das Ergebnis von Boosting dann stark von der (heuristischen) Gewichtung des Regularisierungsterms ab. Die Kombination mit Bayes'schen Lernen erfordert aufgrund der wechselseitigen Beeinflussung der beiden Verfahren noch weitere Forschungsarbeit.

Durch die adaptive Gewichtung der Komiteemitglieder auf Basis ihrer Konfidenzwerte ließen sich für das Diabetes-Problem und das Sinus-Klassifikationsproblem leichte Verbesserungen erreichen, die allerdings nicht signifikant waren. Zum Abschluß dieses Kapitels wurde die Gewichtung mittels Verlustwahrscheinlichkeiten genauer untersucht. Es zeigt sich, daß diese inbesondere dann eine starke Verbesserung bewirken können, wenn die Komiteemitglieder, wie z.B. bei Bagging, beliebig zusammengestellt sind. Bildet man ein Komitee aus unabhängigen Netzen, von denen jedes eine hohe Evidenz hat, dann approximiert das Komitee den zugrundeliegenden Prozeß schon sehr gut. Netze hoher Evidenz entsprechen Lösungen mit niedriger Komplexität und niedrigem Fehler. Diese hatten zugleich eine gleichmäßige Varianz im ganzen Eingaberaum, die auch noch für alle Netze ähnlich war. Das heißt, die Netze haben das Rauschen in den Daten richtig modelliert, so daß durch die adaptive Gewichtung keine weiteren Synergieeffekte entstanden. Die Konfidenzwerte verschlechtern das Komitee aber auch nicht. In jedem Fall sind sie ein wichtiges Kontrollmittel, um zu prüfen, ob neue Eingaben im Arbeitsbereich der Netze liegen. Gegebenenfalls kann der Entwickler bzw. Anwender hier eine Fehlentwicklung erkennen und in den Prozeß eingreifen.

Im nächsten Kapitel werden die hier entwickelten Methoden auf ein reales Anwendungsproblem - die Prognose von Absatzzahlen für viele Verkaufsstellen - angewendet.

Kapitel 7

Prognose von Absatzzahlen

Jeder Zeitungsverlag muß für Titel, die eine hohe Auflage haben, das Problem lösen, diese möglichst gut an die Verkaufsstellen zu verteilen, so daß die Remissionsquote, d.h. der Anteil an Zeitungen, der an den Verlag zurückgeht, minimal wird. Einerseits möchte der Verlag seinen Verkaufserlös maximieren. Dazu muß verhindert werden, daß Verkaufsstellen eine Nachfrage nicht mehr bedienen können, d.h. ausverkauft sind. Andererseits müssen die Kosten minimiert werden, die für Druck und Logistik aufgewendet werden. Eine Verkaufsstelle sollte also im Idealfall keine Zeitungen zurücksenden müssen. Nur wenn beide Ziele optimal gelöst sind, wird der Profit maximal groß.

In der Praxis wird man dies nicht erreichen können, da die genauen Verkäufe jedes Einzelhändlers Zufallseinflüssen unterliegen. Um aber die Remissonsquote möglichst niedrig zu halten, benötigt man eine gute Schätzung der Verkäufe für alle Verkaufsstellen, so daß die Zeitungen optimal verteilt werden können. Der Axel Springer Verlag hat hierzu die Dispositionssoftware RAMBOS[1] entwickelt und stellt sie seinen Grossisten seit 1998 zur Verfügung. Dieses System beinhaltet zwei Komponenten, um die Gesamtliefermenge zu berechnen (Abbildung 7.1). Die erste Komponente basiert auf neuronalen Netzen und schätzt den genauen Absatz an Zeitungen, während die zweite eine Sicherheitsreserve berechnet. Dieser Zuschlag soll verhindern, daß ein Händler mit zu wenigen Zeitungen beliefert wird und vorzeitig ausverkauft ist.

In Kooperation mit dem Axel Springer Verlag in Hamburg beschäftigt sich unser Institut seit 1998 mit der Optimierung der neuronalen Netze zur Verkaufsprognose. Die wesentlichen Ziele, die für die Zusammenarbeit formuliert wurden, waren:

- Verbesserung der mittleren Systemleistung für viele Einzelhändler.
- Optimierung der Eingabestruktur unter der Nebenbedingung, daß für verschiedene Händler möglichst die gleichen Merkmale verwendet werden können.
- Bestimmung von Ursachen für schlechte Prognoseleistungen.

[1] **R**egelbasiertes **A**daptives **M**arktorientiertes **B**ezugs **O**ptimierungs **S**ystem

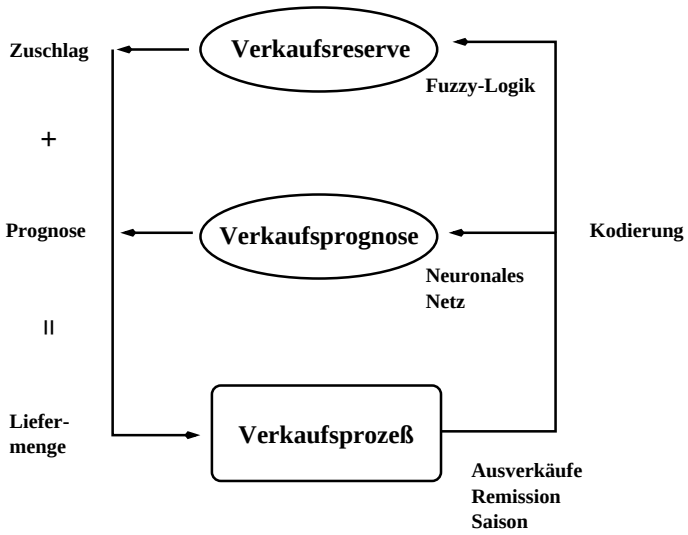


Abbildung 7.1: Die Abbildung zeigt die zwei Komponenten des Systems RAMBOS zur Berechnung der Liefermenge eines Händlers. Ausgehend von der Zeitreihe (Verkaufsprozeß) wird eine Beschreibung in Form einer Kodierung generiert, die als Eingabe für das Prognosesystem benötigt wird. Das neuronale Netz hat die Aufgabe, die Zeitreihe möglichst genau vorherzusagen, während die Fuzzy-Komponente einen Zuschlag berechnet. Dieser soll verhindern, daß es zu einem frühzeitigen Ausverkauf kommt. Der Zuschlag hängt vor allem von der Varianz der Differenzenzeitreihe ab. Aus Zuschlag und Verkaufsprognose ergibt sich dann die tatsächliche Liefermenge. Die Abbildung ist einer Grafik der Bild-Vertriebsabteilung nachempfunden.

Aufbauend auf die bisherigen Ergebnisse wurden inzwischen weitergehende Ziele formuliert, die in laufenden Studien untersucht werden. In Abschnitt 7.5 wird dazu ein Ausblick gegeben. Im folgenden möchte ich zuerst genauer in die Problemstellung einführen und anschließend die Daten, die uns zur Verfügung standen, beschreiben. Danach wird zuerst die Optimierung der Eingabestruktur betrachtet und gezeigt, wie man basierend auf den Methoden aus Kapitel 5.2 eine einheitliche, kompakte Kodierung für viele Datensätze gewinnen kann. Anschließend werden die hier entwickelten Methoden auf die Problemstellung angewendet. Zuerst wird für zwei ausgewählte Verkaufsstellen der empirische Zusammenhang zwischen Evidenz, Testfehler und Größe des Netzes betrachtet und nachgewiesen, daß die evolutive Optimierung der Evidenz zu einer Leistungssteigerung führen kann. Es wird sich zeigen, daß für manche Verkaufsstellen mehrere verschiedene Maxima der Evidenz existieren, die aber einen ganz unterschiedlichen Testfehler haben. Für die Beispiele in Kapitel 5.3 zeigte sich ein ganz ähnliches Bild. Dadurch trat manchmal der Fall ein, daß auch ein deutlich schlechteres Netz ausgewählt wurde. Durch die Komiteebildung auf Basis unabhängiger Netze konnte diese Unsicherheit stark reduziert und der mittlere Fehler signifikant reduziert werden.

Die Vielzahl an Datensätzen ermöglicht es bei dieser Anwendung, eine Leistungssteigerung nachzuweisen und zu belegen, daß diese auch signifikant ist und nicht nur auf eine Zeitreihe beschränkt bleibt.

7.1 Problembeschreibung

Die meistverkaufte Tageszeitung in Deutschland ist die BILD-Zeitung. Jeden Tag werden durchschnittlich 5,2 Millionen Exemplare gedruckt, von denen 4,3 Millionen verkauft werden. Verkauft wird von Montag bis Samstag über etwa 110.000 Angebotsstellen des Einzelhandels. Etwa 100 Großhändler haben die Aufgabe, die Zeitungen so zu verteilen, daß die richtige Menge zum richtigen Zeitpunkt am richtigen Ort ist (Abbildung 7.2). Dabei stützen sie

sich auf unterschiedliche maschinelle Dispositionsverfahren und seit Einführung des Systems RAMBOS auch zunehmend auf dieses. Die Großhändler handeln autonom und sind nicht dazu verpflichtet, das neue System einzusetzen. Die Mitarbeiter der BILD-Vertriebsabteilung, die für RAMBOS verantwortlich sind, müssen also vor allem durch die Leistungsfähigkeit ihres Systems die Grossisten überzeugen.

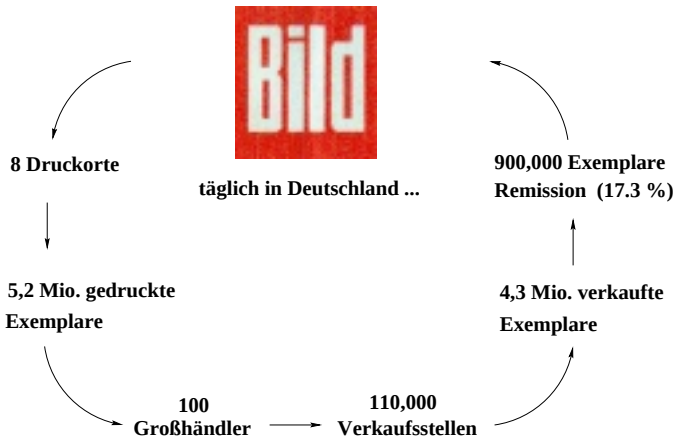


Abbildung 7.2: Die Abbildung illustriert das Problem der Verteilung von Zeitungen. Jeden Tag werden 5,2 Millionen Exemplare der Bildzeitung gedruckt, die an ca. 100 Grossisten zur weiteren Verteilung an die Verkaufsstellen geliefert werden. Jeder Grossist benötigt für jeden Einzelhändler eine möglichst genaue Prognose des erwarteten Verkaufs. Vor dem Einsatz neuronaler Netze zur Verkaufsprognose wurden im Mittel 900 000 Exemplare wieder an den Verlag zurückgegeben, was einer Remissionsquote von 17,3% entspricht. Die Abbildung ist einer Grafik der Bild-Vertriebsabteilung nachempfunden.

Neben der reinen Verkaufsprognose beinhaltet das System auch eine Komponente, die basierend auf Fuzzy-Logik eine Reservemenge berechnet. Dieser Zuschlag hat den Zweck, einen vorzeitigen Ausverkauf zu verhindern und wird im wesentlichen von der Varianz der Differenzenzeitreihe bestimmt. Durch diese notwendige Überbelieferung entsteht die Remission. Man beachte. daß Ausverkäufe auch eine Rückwirkung auf das Datenmaterial haben, da nicht ermitelt werden kann, wieviel Zeitungen noch hätten verkauft werden können, nachdem ein Ausverkauf eintrat. Das heißt, aus Sicht der Datenerfassung wäre es am günstigsten, wenn mindestens ein Exemplar pro Verkaufsstelle übrigbliebe.

Für den Verlag sind die Kosten für zurückgegebene Zeitungen pro Exemplar niedriger als der entgangene Gewinn im Falle eines vorzeitigen Ausverkaufs. Es genügt also nicht, die Remission dadurch zu senken, daß man weniger Exemplare ausliefert. Dies würde auch den Verkaufsgewinn erheblich schmälern. Vielmehr muß man einerseits versuchen, die Ausverkäufe zu vermeiden, indem man die Zeitungen auf Basis einer guten Verkaufsprognose verteilt. Andererseits führt eine Überschätzung des Verkaufs zu einer hohen Remission. Bei gleicher Auflage kann durch eine geschicktere Verteilung der Zeitungen ein deutlicher Mehrverkauf erreicht werden. Diese wesentliche Aufgabe, den Verkauf für jeden Einzelhändler möglichst genau vorherzusagen, wurde von der Vertriebsabteilung des Axel Springer Verlags mittels neuronaler Netze gelöst.

Die Prognoseaufgabe für das Netz ist, die Abweichung von einem gleitenden Durchschnitt[2], der über die letzten 11 Verkäufe berechnet wird, vorherzusagen. In diesem Sinne bildet das neuronale Netz eine Art Fehlerkorrekturmodell. Jeder Wochentag bildet eine eigene Zeitreihe, die getrennt von den anderen Wochentagen aufbereitet wird. Pro Händler existieren also sechs Zeitreihen. Die Zielwerte sind die Differenz zum Verkauf am selben Wochentag der

[2] Dabei handelt es sich um eine ganz spezielle Berechnungsvorschrift, die hier nicht weiter ausgeführt wird.

nächsten und der übernächsten Woche. Das heißt, es wird ein Multi-Tasking Ansatz eingesetzt mit zwei Ausgaben. Durch die Verwendung des übernächsten Wertes in der zweiten Ausgabe wird die Steigung der Zeitreihe miteinbezogen.

Als Eingabe für das Netz dienen vorberechnete Merkmale. Dazu gehören beispielsweise die Zielwerte an den fünf vorangegangenen Zeitpunkten sowie deren Mittelwert, die Veränderungen des gleitenden Durchschnitts, Differenzen zweiter Ordnung, Saisonindikatoren, die einen Auf- bzw. Abschwung in der Zeitreihe signalisieren, sowie Merkmale, die das Verkaufsverhalten an den anderen Wochentagen beschreiben. Insgesamt stehen 47 Merkmale zur Verfügung.

In den folgenden Abschnitten werde ich zuerst einige allgemeine Aussagen zur Zeitreihenprognose machen und dann speziell auf Absatzzahlen eingehen. Dann werden die Daten für diese Studie betrachtet und die Ergebnisse einiger statistischer Tests vorgestellt. Bevor dann die eigentlichen Experimente und deren Resultate dargelegt werden, komme ich noch auf einige spezifische Eigenschaften dieser Aufgabenstellung zu sprechen.

7.1.1 Zeitreihenprognose

Die Prognose von Zeitreihen ist eine der schwierigsten und interessantesten Aufgaben im Bereich des maschinellen Lernens, insbesondere wenn die Zeitreihe den Handel mit einem Wirtschaftsgut erfaßt. Das Kauf- bzw. Verkaufsverhalten unterliegt Zufallseinflüssen, die den zugrundeliegenden Prozeß überlagern. Mit Ansätzen des maschinellen Lernens versucht man, diesen Prozeß auf Basis der Daten zu entdecken und das Rauschen herauszufiltern. Die Prognose solcher Zeitreihen hat eine enorme wirtschaftliche Bedeutung, sowohl für Banken oder Versicherungen als auch für Verlage, Großhändler, Stromversorger u.v.m. Aus diesem Grund finden sich in der Literatur zahlreiche Arbeiten, die sich mit der Analyse und Prognose von Zeitreihen befassen. Für eine Einführung in die Zeitreihenanalyse und in klassische Modellierungsansätze siehe beispielsweise (Box *et al.*, 1994), (Ramsay & Silverman, 1997) oder (Schlittgen & Streitberg, 1999).

Seit 1994 führte unser Institut mehrere Projekte zur Zeitreihenprognose mit Partnern aus der Wirtschaft durch. In Zusammenarbeit mit der Landesbank Hessen-Thüringen wurden mehrere Prognosemodelle für Finanzzeitreihen zum realen Einsatz im Handel entwickelt, beispielweise für verschiedene Futures (Bundfuture, T-Bond-Future) aber auch für Wechselkurse (Dollar-DM) oder Zinsen, siehe z.B. (Gutjahr, 1996, Ragg & Gutjahr, 1997b, Ragg & Gutjahr, 1998, Gutjahr, 1999, Menzel, 1999). In Kooperation mit dem Axel Springer Verlag wird seit 1998 auch die Optimierung von neuronalen Modellen zur Prognose von Absatzzahlen betrachtet (Menzel, 1999, Ragg, 2000, Ragg *et al.*, 2000).

Es ist zu betonen, daß Systeme für Finanzmarktprognosen wesentlich schwieriger zu erstellen und zu evaluieren sind als solche für reine Verkaufszeitreihen. Das liegt unter anderem daran, daß die Käufer bzw. Verkäufer aktive Marktteilnehmer sind und den Verlauf der Zeitreihe durch ihr Handeln mitbestimmen. An den Finanzmärkten übersteigt der reine Handel mit Währungen bereits bei weitem den tatsächlichen 'Umtausch', d.h. Kauf und Verkauf aus einem bestimmten Bedarf heraus. Die meisten Marktteilnehmer handeln also aus der Absicht heraus, durch Spekulation einen Gewinn zu erzielen. Trends lassen sich nur schwer vorhersagen, da beispielsweise dem Kaufverhalten bezüglich einer anderen Währung

kein regelmäßiges Verhaltensmuster zugrunde liegt wie dem Kauf einer Zeitung. Die Evaluation eines Modells ist schwieriger, da die Trefferquoten (bezüglich einer Steigt/Fällt-Klassifikation) um etwa 55% liegen. Das heißt, man benötigt einige Jahre an Testdaten, um zu belegen, daß ein Modell signifikant besser ist als eine Zufallsprognose.

Bei Verkäufen von Zeitungen oder anderen Wirtschaftsgütern existieren dagegen oft Trends, die regelmäßig wiederkehren. Man denke beispielweise an die Stromlastanforderung an einen Stromversorger während des Tages und der Jahreszeiten. Morgens steigt der Bedarf an Strom stark an, geht dann während der gewöhnlichen Mittagspause zurück. Nach einer erneuten Zunahme am Nachmittag und frühen Abend fällt die Lastanforderung in der Nacht auf ein Minimum. Diese Grundtendenz ist überlagert von kurzfristigen Trends, beispielsweise einem höheren Strombedarf bei nationalen Ereignissen im Fernsehen. In ähnlicher Weise findet man Trends beim Verkauf von Zeitungen. In Feriengebieten gibt es typische Aufschwung- und Abschwungphasen zu Beginn und am Ende der Saison. In den Großstädten kann die Zeitreihe eher gleichmäßig verlaufen. Etliche Faktoren beeinflussen zusätzlich das Verkaufsverhalten jedes Einzelhändlers, z.B. die Veränderung der Auslage oder die momentane Erreichbarkeit der Verkaufsstelle aufgrund von Baustellen oder einer geänderten Verkehrsführung. Dadurch entstehen kurzfristige Trends in der Zeitreihe.

Einige der hier betrachteten Zeitreihen sind so gleichmäßig, daß die Prognoseaufgabe als lineares Modell realisiert werden kann. Bei anderen dagegen erzielt man mit nicht-linearen Verfahren einen deutlichen Zugewinn an Leistung. Möchte man die optimale Leistung erzielen, dann muß man für jeden Datensatz die Komplexität des Modells automatisch optimieren. Das heißt, das Verfahren muß erkennen, wann ein Problem eher linearer Natur ist und die Zahl der versteckten Neuronen dann entsprechend reduzieren. Die evolutive Modelloptimierung, wie sie in Kapitel 5 eingeführt wurde, soll genau dies leisten.

Die Evaluation der Methoden ist für diese Anwendung wesentlich einfacher als für Finanzmarktprognosen. Die Trefferquoten bezüglich einer Steigt/Fällt-Klassifikation liegen zwischen 60% und 70%. Weiterhin stehen viele Datensätze für viele verschiedene Gebiete zur Verfügung. Diese große Anzahl erlaubt es, signifikante Leistungsunterschiede zu bestimmen.

7.1.2 Daten

Die Daten, die für diese Studie zur Verfügung standen, stammen aus dem Verkaufsgebiet 'Münster', einer mittelgroßen Universitätsstadt mit einer ländlich geprägten Umgebung. Insgesamt wurden Zeitreihen von 133 Verkaufsstellen verwendet und davon 50 zufällig ausgewählt, um die Methoden zur Optimierung der neuronalen Netze zu vergleichen.

Eine wesentliche Aufgabe bei der Zeitreihenprognose besteht darin, die Aufgabe so zu formulieren, daß sie durch ein maschinelles Lernverfahren bewältigt werden kann. Die Originalverkäufe bilden in aller Regel keinen stationären Prozeß. Abbildung 7.3 zeigt für zwei Händler die Verkaufsdaten über sechs Jahre. Dabei sind deutlich Trends zu erkennen. Für den Händler Nr. 0117 (linke Seite) kann man sowohl für die Mittwochsreihe als auch für die Samstagsreihe kurzfristige Trends bestimmen. Das Verkaufsverhalten von Händler Nr. 1659 ist durch einen langfristigen Abwärtstrend geprägt, der am Samstag am Anfang sehr stark ist und dann gleichmäßig abnehmend. Im Anhang sind alle relevanten Zeitreihen dieser beiden Händler abgebildet.

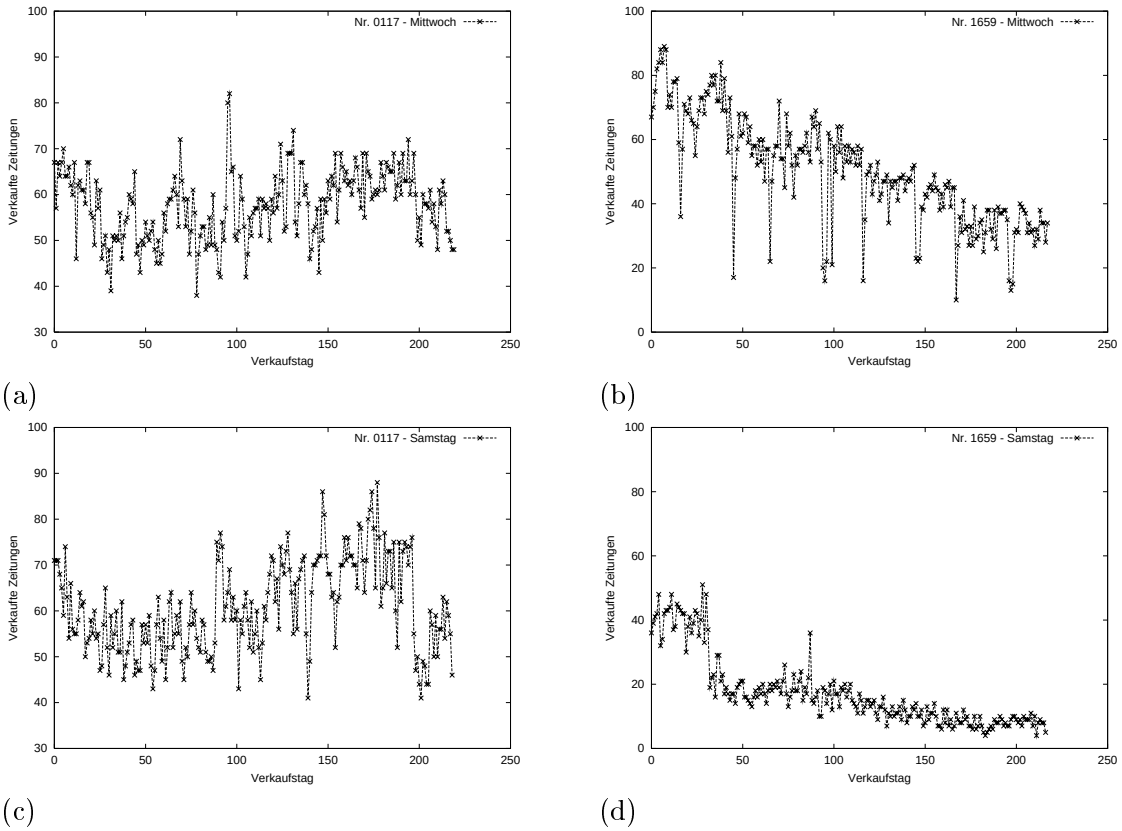


Abbildung 7.3: Die Abbildung zeigt die absoluten Verkaufszahlen von Herbst 1992-1996 für zwei Einzelhändler aus dem Gebiet 'Münster' an zwei verschiedenen Wochentagen. Händler-Nr. 1659 (rechte Spalte) zeigt einen deutlichen langfristigen Trend. Ebenso ist das Verkaufsverhalten am Samstag sehr unterschiedlich zu den Wochentagen. Händler-Nr. 0117 hat einige kurzfristige Trends. a) Mittwoch; Händler-Nr. 0117. b) Mittwoch; Händler-Nr. 1659. c) Samstag; Händler-Nr. 0117. d) Samstag; Händler-Nr. 1659.

Das Schätzen einer bedingten Wahrscheinlichkeit, wie es ein neuronales Netz leistet, setzt aber voraus, daß sich die zugrundeliegende Verteilung nicht ändert. In aller Regel geht man deswegen zur Zeitreihe der Differenzen erster Ordnung über. Für die Aufgabe genügt es, wenn man die Veränderung der Zeitreihe prognostizieren kann. Aus dem momentanen Wert und der erwarteten Veränderung ergibt sich dann der Wert zum nächsten Zeitpunkt. Abbildung 7.4 zeigt die entsprechenden Differenzenzeitreihen.

Die Differenzen erster Ordnung bilden offensichtlich einen mittelwertstationären Prozeß. Für diese Zeitreihe sollte es möglich sein, ein neuronales Modell auf Basis der Daten der Vergangenheit zu trainieren. Im nächsten Abschnitt werden die Fragen der Stationarität mit statistischen Tests noch genauer untersucht.

In der Einführung zu diesem Kapitel wurde bereits erwähnt, daß der Zielwert der Prognose nicht die Veränderung zum nächsten Verkaufstag ist, sondern die Abweichung von einem gleitenden Durchschnitt. Der Ansatz des Axel Springer Verlages beruht weiterhin darauf, daß für jeden Wochentag die zugehörige Zeitreihe aufbereitet und die entsprechende Kodierung erstellt wird, dann aber die sechs Datensätze zusammengenommen und ein einziges neuronales Netz pro Händler trainiert wird. Dieses Vorgehen hat sich in mehrjähriger Ent-

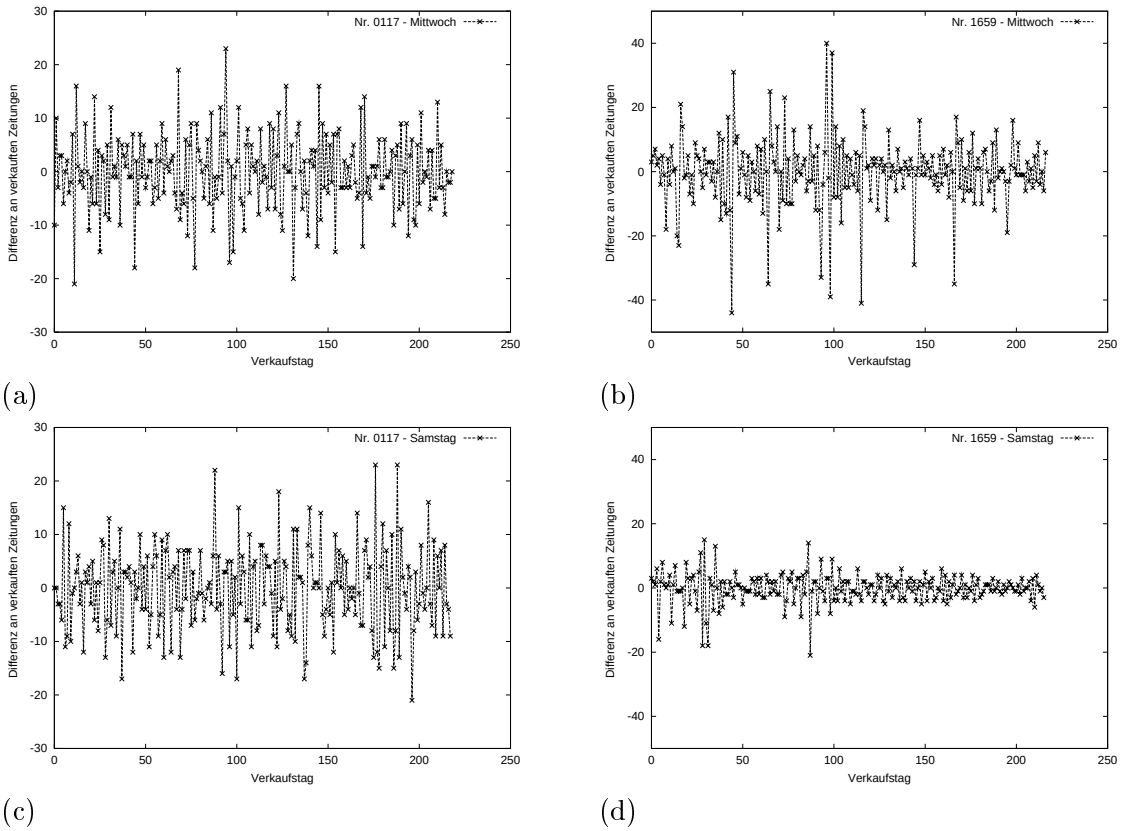


Abbildung 7.4: Die Abbildung zeigt die Differenzen der Verkaufszahlen von Herbst 1992-1996 für zwei Einzelhändler aus dem Gebiet 'Münster' an zwei verschiedenen Wochentagen. Die Differenzen werden vom Verkauf am Tag der aktuellen Woche zum Verkauf am gleichen Tag der nächsten Woche berechnet. a) Mittwoch; Händler-Nr. 0117. b) Mittwoch; Händler-Nr. 1659. c) Samstag; Händler-Nr. 0117. d) Samstag; Händler-Nr. 1659.

wicklungsarbeit empirisch als erfolgreich herausgestellt, auch wenn damit einige Probleme verbunden sind. Am Ende des Kapitels werden diese skizziert und Lösungsmöglichkeiten vorgeschlagen. Andererseits können durch diese Form der Modellierung aber auch Synergieeffekte genutzt werden. Es steht außerhalb der Zielsetzung dieser Arbeit zu beurteilen, welche Form der Modellierung für diese Anwendung die geeignetste ist. Sie wird hier als gegeben vorausgesetzt. Das Ziel ist es dann, für diese Aufgabenstellung mit den hier entwickelten Methoden eine Leistungssteigerung zu erreichen. Abbildung 7.5 zeigt die Zeitreihen der Zielwerte, wie sie den neuronalen Netzen präsentiert werden. Abgebildet sind jeweils die Trainingsdaten für die obigen Händler (Nr. 0117 und Nr. 1659) für die Mittwochszeitreihe und die Samstagszeitreihe.

Die Daten für jede Wochentagszeitreihe werden standardisiert, d.h. der Mittelwert wird abgezogen und durch die Standardabweichung geteilt. Diese Werte werden auf den Trainingsdaten berechnet. Durch die Skalierung erhalten die sechs Zeitreihen ein einheitliches Niveau. Trainiert wird dann pro Händler nur ein Netz. Im Realbetrieb wird auf allen Daten trainiert. Das heißt, Kreuzvalidierung zur Modellauswahl ist nicht möglich. Daraus folgt, daß entweder die mittlere Leistung des Verfahrens sehr gut sein muß oder daß ein Selekti-

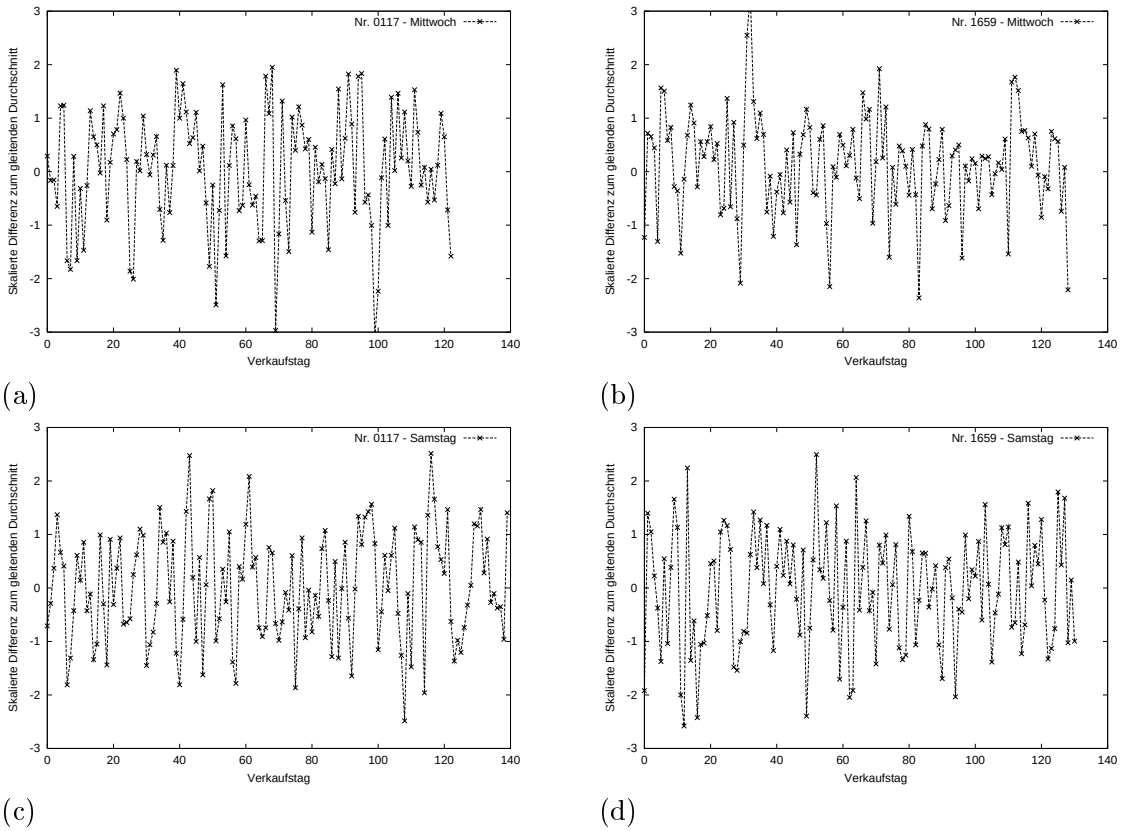


Abbildung 7.5: Die Abbildung zeigt die Zielwerte für die neuronalen Netze auf dem Trainingszeitraum von Herbst 1992-1995 für zwei Einzelhändler aus dem Gebiet 'Münster' an zwei verschiedenen Wochentagen. Die Zielwerte entsprechen der Differenz des nächsten bzw. übernächsten Verkaufs zum gleitenden Durchschnitt über 11 Tage. a) Mittwoch; Händler-Nr. 0117. b) Mittwoch; Händler-Nr. 1659. c) Samstag; Händler-Nr. 0117. d) Samstag; Händler-Nr. 1659.

onskriterium wie die Evidenz im Bayes'schen Ansatz verwendet wird, das es erlaubt, bessere Modelle von schlechteren zu trennen. Das Training der Modelle wird nach einiger Zeit mit den jeweils neuesten Daten wiederholt.

7.1.3 Prognose für viele Verkaufszeitreihen

An den bisherigen Ausführungen wurde der ganz besonderen Charakter der hier behandelten Aufgabenstellung deutlich: Es existieren viele Datensätze für verschiedenste Einzelhändler, wie z.B. Supermärkte, Tankstellen, Bäckereien etc., von denen jeder eine eigene Charakteristik hat. Zudem unterscheiden sich noch ganze Regionen, je nachdem ob es sich um eine Großstadt, ein ländliches Gebiet oder eine Ferienregion handelt. Andererseits ist das Kaufverhalten nicht völlig beliebig. Die Zeitreihen stimmen in gewissen Merkmalen überein. Tatsächlich gibt es zum Teil starke Korrelationen zwischen verschiedenen Verkaufszeitreihen. Im Abschnitt 7.5 werde ich darauf nochmals zurückkommen.

Die Schwierigkeit der Aufgabenstellung liegt unter anderem darin, daß man einerseits eine möglichst einheitliche Modellierung der verschiedenen Zeitreihen erreichen möchte, damit

der Aufwand für das Gesamtsystem niedrig bleibt. Dazu gehört auch, daß das Training und die Auswahl ohne Interaktion durch einen menschlichen Bediener abläuft. Andererseits soll natürlich die spezielle Charakteristik jedes Datensatzes erkannt werden.

Um eine möglichst einheitliche Modellierung zu erreichen, wird im Abschnitt 7.2 die Rückwärtssuche dahingehend erweitert, daß jeweils im zweiten Schritt die Korrelation bzw. der Informationsgehalt auf allen Datensätzen gleichzeitig berechnet und jeweils das betreffende Merkmal markiert wird. Das Merkmal, das am häufigsten markiert wurde, wird dann für alle Datensätze entfernt. Daraufhin wird dann wieder der individuell günstigste Pruning-Schritt berechnet. Der Gedanke dabei ist, daß durch dieses Vorgehen Zufälligkeiten in den Daten, die die Merkmalsauswahl bei einem individuellen Datensatz entscheidend bestimmen, durch die Mittelung eliminiert werden. Nehmen wir zum Beispiel an, Merkmal 1 und 2 sind stark miteinander korreliert. Ist die Korrelation zur Ausgabe für verschiedene Händler jetzt derart, daß manchmal Nummer 1, manchmal aber auch Nummer 2 gelöscht wird, dann erhalten wir mit Sicherheit eine unterschiedliche Merkmalskombination für die beiden Händler. Insbesondere deshalb, da sich das für andere Merkmale fortsetzen kann. Die Unterscheidung ist aber nicht Ausdruck von ganz spezifischen Eigenschaften eines Datensatzes, sondern im wesentlichen durch Zufälligkeiten bestimmt, solange noch reichlich Merkmale mit hoher Korrelation vorhanden sind.

In Kapitel 5 haben wir an einigen Beispielen gesehen, daß die Anzahl der versteckten Neuronen eine wichtige Rolle spielt. Trotz Regularisierung kann die Leistung bei unterschiedlicher Größe der versteckten Schicht deutlich schwanken. Im Abschnitt 7.3 werden wir an zwei Beispielen sehen, daß sich das für die Prognose der Absatzzahlen ebenso verhält.

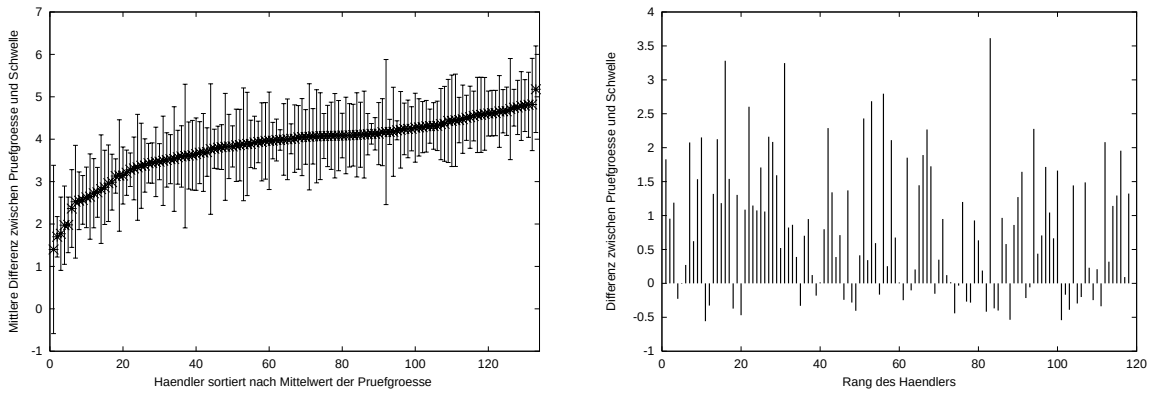
Bevor anschließend die Optimierung des Eingabevektors betrachtet wird, gilt es noch, eine Vorbedingung zu untersuchen. Diese betrifft die Frage, ob die Zeitreihe überhaupt prognostizierbar ist. Dazu muß sichergestellt sein, daß es sich um einen stationären Prozeß handelt. Im nächsten Abschnitt wird das für die vorliegenden Daten behandelt.

7.1.4 Statistische Tests

Im Rahmen des Projektes zur Vorhersage von Absatzzahlen wurden umfangreiche Untersuchungen mit statistischen Tests durchgeführt. Die Ergebnisse, die im Kontext dieser Arbeit von Bedeutung sind, werden im folgenden zusammengefaßt. Dies betraf unter anderem die Frage,

- ob die Zeitreihen zufällig oder vorhersagbar sind,
- in welcher Form die Zeitreihen stationär sind,
- ob das Training der sechs einzelnen Zeitreihen jedes Händlers in einem Netz Probleme verursacht.

Um auf Zufälligkeit der Zeitreihe zu testen, wurde ein Iterationstest durchgeführt (Büning & Trenkler, 1994). Der Test ist im Anhang beschrieben (Kapitel A.1.3). Die Grundidee des Tests ist die folgende: Man teilt die Daten in zwei Klassen ein ('Steigt'/'Fällt') und betrachtet für die Zeitreihe die Anzahl an Übergängen von 'Steigt' auf 'Fällt' und umgekehrt. Zuwenig oder zuviele Sprünge zeigen an, daß die Bewegungen nicht zufällig sind. Die Nullhypothese H_0, daß die Folge der Beobachtungen an 'Steigt'/'Fällt' zufällig ist, wird dann abgelehnt.



(a) (b)

Abbildung 7.6: Ergebnisse des Iterationstests auf Zufälligkeit. Getestet wurde, ob die Reihenfolge der Auf- und Abbewegung der Zeitreihen einer zufälligen Bewegung entsprechen. Die Nullhypothese ist, daß die Bewegungen zufällig sind. Berechnet wurde jeweils die Differenz zwischen Prüfgröße und Schwelle. (a) Iterationstest für jede der sechs Differenzen-Zeitreihen eines Händlers. Abgebildet ist für jeden Händler der Mittelwert und die Standardabweichung der Differenz. Außer für eine einzelne Zeitreihe wird der Test durchgehend abgelehnt, d.h. alle Zeitreihen der absoluten Differenzen enthalten eine Struktur. (b) Iterationstest für die Trainingsdaten, wie sie den neuronalen Netzen zur Verfügung stehen. Für jeden Händler existiert hier nur eine Zeitreihe. Abgebildet ist die Differenz zwischen Prüfgröße und Schwelle. Für etwa ein Viertel der Datensätze wird die Nullhypothese angenommen. Das heißt, sie haben keine Struktur, die mit dem Test erkennbar wäre.

Der Test berechnet eine Prüfgröße, die asymptotisch normalverteilt ist. H_0 wird abgelehnt, wenn der Betrag der Prüfgröße den zugehörigen Wert der Inversen der Normalverteilung überschreitet. Abbildung 7.6 zeigt die Ergebnisse des Iterationstests für die Differenzen-Zeitreihen (linkes Bild) und die Trainingsdaten, wie sie dem Netz präsentiert werden. Die absoluten Differenzen folgen gemäß dem Test keinem Zufallsschema. H_0 wird durchweg abgelehnt. Für die Zielwerte, wie sie dem Netz präsentiert werden, ist das Ergebnis etwas schwächer. Die Differenzen zum gleitenden Durchschnitt sind offenbar schwieriger vorherzusagen. Der Test wird in etwa einem Viertel aller Fälle angenommen. Das heißt, es ist mit dem Test keine Struktur in den Daten nachzuweisen.

Der zweite wichtige Test, der durchgeführt wurde, betraf die Stationarität der Zeitreihen. Hierzu wurde mit dem Kolmogorow-Smirnow-Test (KS-Test) auf Gleichheit der Verteilung geprüft. Die genaue Beschreibung des Tests findet sich ebenfalls im Anhang (Kapitel A.1.4). Sollten sich die Trainings- und Testdaten unterscheiden, kann man die Zeitreihe als nichtstationär betrachten.

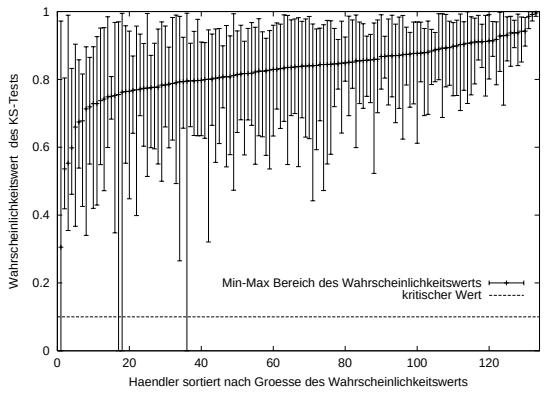
Für den KS-Test nimmt man an, daß die Stichprobenvariablen unabhängig sind und stetige Verteilungsfunktionen F bzw. G haben. Die Nullhypothese ist, daß $F(z) = G(z)$ gilt, für alle $z \in R$. Die Prüfgröße des KS-Tests bestimmt sich aus den Differenzen der empirischen Verteilungsfunktionen, $K_{m,n} = \max_z |F_m(z) - G_n(z)|$, mit

$$F_m(z) = \begin{cases} 0 & : \quad z < x_{(1)} \\ i/m & : \quad x_{(i)} \leq z < x_{(i+1)}, i = 1, .., m-1 \\ 0 & : \quad z \geq x_{(m)} \end{cases}$$

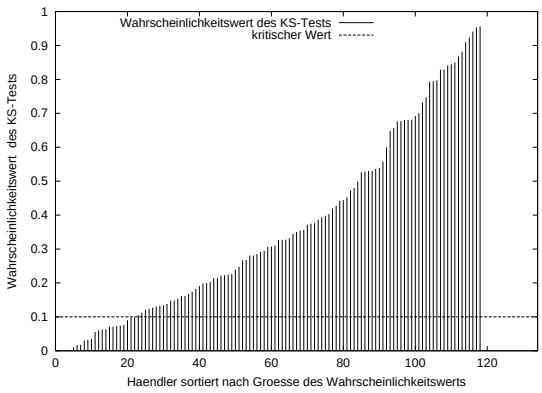
und $G_n(z)$ entsprechend. H_0 wird abgelehnt, wenn für die Prüfgröße $K_{m,n}$ der tabellierte Schwellwert $k_{1-\alpha}$ zum Signifikanzniveau α überschritten wird. Weiterhin berechnet der KS-Test auf Basis der Prüfgröße noch eine Wahrscheinlichkeit - den sogenannten p-Wert - dafür, daß die Verteilungen gleich sind. Bei kleinen Werten für die Wahrscheinlichkeit wird der Test abgelehnt. Der p-Wert ist wesentlich informativer als die bloße Aussage, ob der Test abgelehnt wird oder nicht (Büning & Trenkler, 1994). Abbildung 7.7 bewertet die Zeitreihen mittels dieser Wahrscheinlichkeit.

Für die absoluten Differenzen wird der Test nur selten abgelehnt, d.h. man kann diese als stationär betrachten. Der über die sechs Zeitreihen jedes Händlers gemittelte p-Wert liegt für alle Händler deutlich über dem kritischen Wert, der zu 0.1 gewählt wurde (Abbildung 7.7a).

Für die tatsächliche Prognoseaufgabe des neuronalen Netzes, die Differenz zum gleitenden Durchschnitt zu schätzen, ist die Lage etwas ungünstiger. Berechnet man den p-Wert auf der ersten Ausgabe der Trainings- und Testmenge, dann wird der Test für 22 Datensätze abgelehnt. Die zweite Ausgabe liefert dasselbe Ergebnis, da sie nur zeitverschoben ist.



(a)



(b)

Abbildung 7.7: Ergebnisse des Kolmogorow-Smirnow-Test auf Gleichheit der Verteilungen bezüglich der Trainings- und Testdaten. Aufgetragen ist jeweils der p-Wert, den der KS-Test berechnet. Die Händler sind der besseren Übersicht wegen sortiert nach der Größe des p-Wertes bzw. dessen Mittelwert angeordnet. (a) Zeitreihen der absoluten Differenzen. Der KS-Test wurde auf den sechs Differenzen-Zeitreihen jedes Händlers berechnet. Aufgetragen ist der mittlere p-Wert und der Min-Max Bereich. Der Test wird fast ausnahmslos angenommen. (b) Differenzen zum gleitenden Durchschnitt. Berechnet wurde der p-Wert auf der ersten Ausgabe von Trainings- und Testmenge. Der Test wird für 22 Datensätze abgelehnt.

Für die nicht-stationären Zeitreihen bleibt die Mittelwertstationarität im allgemeinen erhalten, da die Differenzen weiter um den Nullpunkt schwanken. Das gilt allerdings nicht für die Varianzstationarität. Sinkt die Varianz, dann ist das für die Prognose keine Schwierigkeit. Steigt sie dagegen, dann wird der Bereich verlassen, in dem Daten für das Training vorhanden waren. Das heißt, das neuronale Netz müßte extrapolieren. Auf die Prognose ist damit wenig Verlaß. Um zu überprüfen, ob die Varianz steigt, wurde der F-Test auf Gleichheit zweier Varianzen verwendet (siehe Kapitel A.1.2).

Insgesamt bleibt festzuhalten, daß die Veränderungen in den Verkäufen einen prognostizierbaren, stationären Prozeß bilden. Mit einer leichten Einschränkung gilt das auch für die Zeitreihe, die die eigentliche Zielgröße für das neuronale Netz bildet. Damit sollte sich

durch die Fehlerkorrektur, die gelernt werden soll, eine erkennbare Leistungssteigerung gegenüber der Prognose mittels des gleitenden Durchschnitts ergeben. Da die Ausgabedaten standardisiert sind, d.h. Mittelwert 0 und Varianz 1 besitzen, hat der gleitende Durchschnitt einen mittleren Fehler von 1. Eine erfolgreiche Prognose muß also im Durchschnitt über alle Händler signifikant darunter liegen.

Zum Vergleich der verschiedenen Methoden werden später nur die stationären Zeitreihen herangezogen. In Abschnitt 7.5 wird gezeigt, daß die Nicht-Stationarität mit einem extrem großen Generalisierungsfehler, d.h. größer als die einfache Prognose mittels des gleitenden Durchschnitts, verbunden ist. Dies ist unabhängig davon, ob das zugrundeliegende Modell linear oder nicht-linear ist, nur daß der Fehler im nicht-linearen Fall in der Regel größer ausfällt und damit der Vergleich auf den prognostizierbaren Zeitreihen verzerrt wird.

7.2 Optimierung der Eingabestruktur

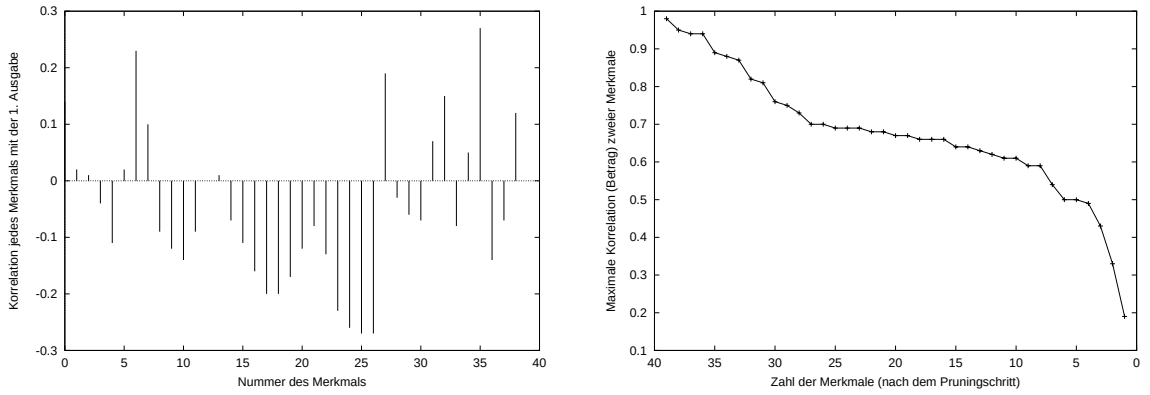
Die vom Axel Springer Verlag entwickelte Kodierung hatte ursprünglich 47 Merkmale, von denen nach einer visuellen Untersuchung 8 Merkmale vorab entfernt wurden. Hierzu wurden die Merkmale über der Zeit sowie gegen die Ausgabe aufgetragen. Bei den Merkmalen 7,8 und 9 handelt es sich um saisonale Indikatoren, die in dem ländlichen Gebiet keine Rolle spielen. Die Merkmale 30 bis 34 besaßen einige visuell auffällige Unregelmäßigkeiten. Daraufhin wurde der Infomationsgehalt dieser Eingaben bezüglich der Ausgabe berechnet. Da dieser sehr gering ausfiel, wurde der Eingabevektor um diese 5 Merkmale reduziert. Dies war notwendig, da die Korrelation dieser Merkmale mit anderen sehr gering war. In der ersten Stufe, der Korrelationsanalyse, wären sie demnach nicht entfernt worden. Die visuelle Kontrolle ist ein wichtiges Hilfsmittel, um Merkmale zu prüfen. Silverman gibt die Empfehlung, daß man sich nie ausschließlich auf die Ergebnisse statistischer Verfahren verlassen sollte (Ramsay & Silverman, 1997). Der weiterhin zu untersuchende Eingabevektor hatte dann noch 39 Komponenten.

Im folgenden werden die beiden Verfahren aus Kapitel 5 – Rückwärtssuche basierend auf Korrelationen bzw. auf dem Informationsgehalt – auf die gegebenen Daten angewendet. Dabei wird die Modifkation, die weiter oben beschrieben wurde, verwendet. Durch die Mittelung über viele Datensätze soll das Ziel erreicht werden, eine einheitliche, kompakte Kodierung zu finden. Wenn im folgenden von Korrelation oder Informationsgehalt gesprochen wird, dann ist damit immer die Mittelung über alle Datensätze gemeint.

Die Korrelation der Merkmale mit der ersten Ausgabe[3] ist in Abbildung 7.8a gezeigt. Man erkennt, daß einige Merkmale eine deutliche lineare Beziehung zur Ausgabe besitzen. Daraus folgt auch, daß die Zeitreihen prinzipiell vorhersagbar sind, ansonsten könnte keine Beziehung mit einem linearen Anteil bestehen. Acht Merkmale haben eine (betragsmäßige) Korrelation zwischen 0.2 und 0.3. Wendet man nun die Rückwärtssuche basierend auf dem Korrelationskoeffizienten an, dann zeigt sich, daß man 24 Merkmale entfernen kann, bis die maximale Korrelation zweier Merkmale unter 0.65 fällt (Abbildung 7.8b). Auch im weiteren Verlauf nimmt die maximale Korrelation nur langsam ab, bis der Eingabevektor nur noch 9 Komponenten hat. Das ist umso erstaunlicher, da es sich hier um einen Mittelwert über

[3] Für die zweite Ausgabe ergibt sich ein ähnliches Bild, auf das hier verzichtet werden kann.

viele Datensätze handelt. Damit läßt sich der Eingabevektor drastisch verkleinern. Für die weiteren Untersuchungen wurden noch 15 Merkmale verwendet.



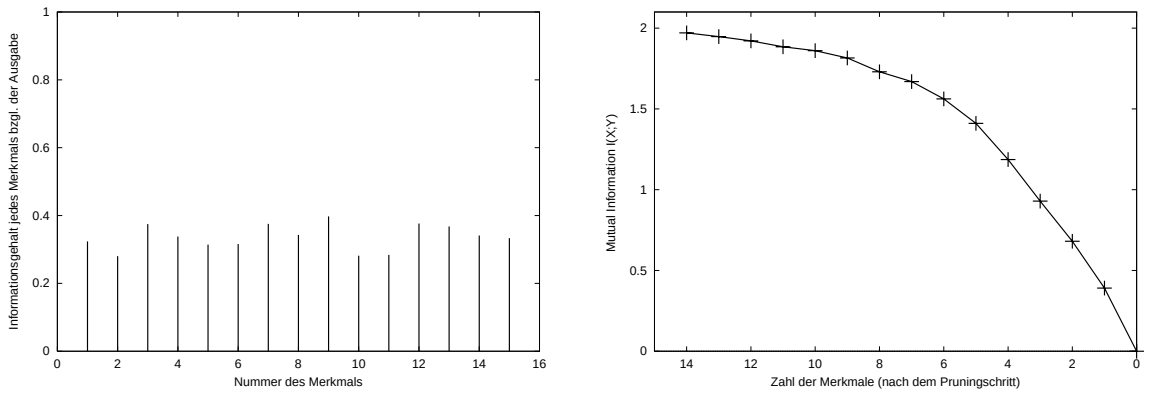
(a) (b)

Abbildung 7.8: Ergebnisse der Korrelationsanalysen. 39 von insgesamt 47 Merkmalen wurden betrachtet. 8 Merkmale wurden vorab entfernt. (a) Gemittelte Korrelation $\rho(X_i, Y)$ zwischen den einzelnen Merkmalen X_i und der Ausgabe Y für die Prognose der Verkaufszahlen. Die Korrelation jedes Merkmals wurde für jeden Datensatz bestimmt und dann über alle gemittelt. (b) Ergebnisse der Rückwärtssuche basierend auf dem Korrelationskoeffizienten. Die Korrelationen wurden über alle Datensätze gemittelt und in jedem Schritt das Paar bestimmt, das bezüglich der gemittelten Korrelationsmatrix den maximalen Wert hat. Wenn noch 15 Merkmale vorhanden sind, liegt die maximale Korrelation zweier Merkmale noch über 0.6.

Abbildung 7.9a zeigt den Informationsgehalt der einzelnen Merkmale über die Ausgabe. Der Wert des Mutual Information beträgt für alle Merkmale zwischen 0.2 und 0.3. Die Unterschiede zwischen den Merkmalen sind weniger stark ausgeprägt als bei Verwendung des Korrelationskoeffizienten, d.h. die Merkmale stehen auch in einer nicht-linearen Beziehung zur Ausgabe. Betrachtet man die Rückwärtssuche mit Mutual Information (Abbildung 7.9b), dann gleicht das langsame Absinken der Kurve dem Verlauf bei der Krebs-Klassifikation (Abbildung 5.9) und dem Add10-Regressionsproblem (Abbildung 5.11). In beiden Fällen konnte man auf Merkmale verzichten. Das langsame Absinken ist für diese Anwendung umso aussagekräftiger, da es sich um einen Mittelwert über viele Datensätze handelt. Auch in diesem Fall wird die Eingabe auf 10 Merkmale reduziert, d.h. fünf weitere Eingabeneuronen werden entfernt.

Für die beiden Merkmalsvektoren mit 47 und mit 10 Komponenten wurden sowohl das lineare Modell, das Lernen mit Weight Decay und das Bayes'sche Lernen miteinander verglichen. Für die letzten beiden wurden jeweils 50 Gewichtsinitialisierungen trainiert. Der Weight Decay wurde auf einigen Datensätzen jeweils optimal eingestellt und dann gemittelt und für alle Datensätze verwendet. Die zugrundeliegende Topologie hatte 4 versteckte Neuronen. Mit dieser Topologie wird im Realbetrieb bisher noch gearbeitet.

Abbildung 7.10 zeigt, daß es für das gegebene Problem sehr schwer ist, für viele Datensätze mit einer festen Topologie für alle Händler gleichzeitig gute Ergebnisse zu erreichen. Die Prognosen können gegenüber dem linearen Modell zwar deutlich besser werden, aber auch schlechter. Das deutet darauf hin, daß man neben der Optimierung der Gewichtung des Regularisierungsterms auch die Zahl der versteckten Neuronen und die Merkmalskombination betrachten muß. Im nächsten Abschnitt wird dieser Aspekt an zwei ausgewählten Zeitreihen verdeutlicht. Betrachtet man den mittleren Fehler, dann zeigt sich weiterhin, daß sich

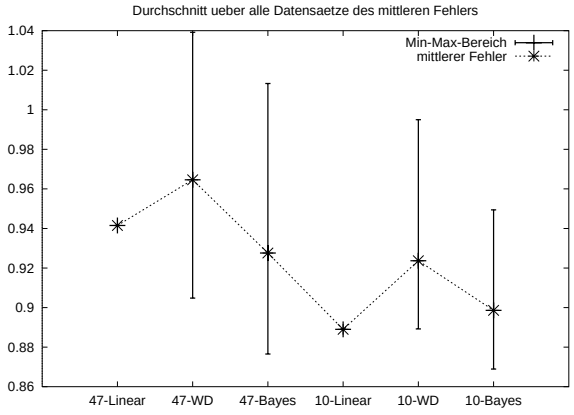


(a) (b)

Abbildung 7.9: Untersuchung des Informationsgehaltes. (a) Gemittelter Informationsgehalt $I(X_i, Y)$ zwischen den einzelnen Merkmalen X_i und der Ausgabe Y für die Prognose der Verkaufszahlen. Der Informationsgehalt für jedes Merkmal wurde für jeden Datensatz bestimmt und dann über alle gemittelt. Es wurden nur die 15 Merkmale betrachtet, die nach der Korrelationsanalyse noch verblieben sind. (b) Ergebnisse der Rückwärtssuche basierend auf Mutual Information. Um einen Pruning-Schritt auszuführen, wird zuerst für alle Datensätze das Merkmal berechnet, das weggelassen werden kann. Dann wird dasjenige bestimmt, daß am häufigsten ausgewählt wurde und bei allen Datensätzen entfernt. Der angegebene Wert ist das Mutual Information des verbleibenden Eingabevektors gemittelt über alle Datensätze.

der Einsatz der mächtigeren Verfahren nur lohnt, wenn man viele Modelle trainiert. Nur in diesem Fall kann das Evidenz-Kriterium wirklich genutzt werden. Trainiert man nur wenige Netze pro Händler, dann ist es bei der gegebenen Aufgabenstellung günstiger, ein lineares Modell zu verwenden.

Abbildung 7.10: Die Abbildung zeigt Ergebnisse für die Prognose der Absatzzahlen für den 47-dimensionalen und den 10-dimensionalen Eingabevektor. In beiden Fällen wird die Leistung des linearen Modells mit dem Training mit Weight Decay sowie dem Bayes'schen Lernen für ein Netz mit 4 versteckten Neuronen verglichen. Das ist die Topologie, wie sie im Moment im realen Einsatz verwendet wird. Für jeden Datensatz wurden 50 Netze trainiert. Angegeben ist der Mittelwert (über alle Händler) des minimalen, mittleren und maximalen Fehlers. Der mittlere Fehler gibt an, mit welcher Leistung man rechnen kann, wenn man Netze trainiert und das oft wiederholt. Das entspricht dem bisherigen Anwendungsfall, wenn die Topologie fest vorgegeben ist.



Für die eben vorgestellten Experimente war die Topologie für alle Händler einheitlich vorgegeben. Die Beispiele in Kapitel 5 zeigten, daß für jeden Datensatz die optimale Zahl an versteckten Neuronen unterschiedlich sein kann. Den vollen Nutzen aus der Bayes'schen Methode wird man für die gegebene Anwendung nur ziehen können, wenn die Optimierung

der Zahl der versteckten Neuronen und der Merkmale ebenso betrachtet wird. Im nächsten Abschnitt wird dazu das Konzept der evolutiven Optimierung angewendet, nachdem an zwei Beispielen demonstriert wurde, daß dieser Weg erfolgversprechend ist.

7.3 Optimierung der Evidenz

Die Abhängigkeit der Güte des Modells von der Zahl der versteckten Neuronen und der Merkmale erfordert für diese Anwendung einen Ansatz, der automatisch eine möglichst optimale Lösung bestimmt. Es ist völlig unmöglich, diese für jeden Datensatz gesondert zu optimieren. Die evolutive Optimierung, wie sie in Kapitel 5 vorgestellt wurde, ist gerade für diese Art der Problemstellung hervorragend geeignet. An zwei Zeitreihen soll exemplarisch gezeigt werden, daß die Methode erfolgreich eingesetzt werden kann. Dazu wird das Ergebnis von 10 Versuchen der evolutiven Optimierung wieder zur empirischen Abhängigkeit in Relation gesetzt. Für den realen Anwendungsfall wird dann nur noch ein Versuch pro Händler durchgeführt. Früher hatten wir bereits gesehen, daß die evolutive Optimierung die Varianz der Lösungen deutlich reduziert. Das heißt, die Gefahr, daß man ein schlechtes Modell auswählt, ist deutlich geringer. Daher sollte sich im Mittel über viele Zeitreihen die Leistungssteigerung durch die Optimierung der Evidenz bemerkbar machen.

Abbildung 7.11 zeigt für zwei Datensätze (Händler Nr. 0117 und Nr. 1659) die empirische Abhängigkeit der Evidenz bzw. des Testfehlers von der Zahl der versteckten Neuronen. Für jede Topologie außer der linearen wurden 50 Netze trainiert, 750 Netze insgesamt.

Im ersten Fall (Abbildungen 7.11a und c) ergibt sich ein Maximum der Evidenz bei Verwendung von zehn Neuronen. Die mittlere Evidenz sinkt über den ganzen Bereich deutlich ab. Außerdem nimmt mit zunehmender Komplexität die Varianz ebenfalls zu. Das heißt, es treten auch viele schlechte Modelle auf. Der Testfehler erreicht ein Minimum bei zehn Neuronen. Das lineare Modell liegt aber nur 2% darüber. Man kann bei diesem Datensatz also von einem nahezu linearen Zusammenhang ausgehen. Die Korrelation von Evidenz und Testfehler über alle 750 Netze beträgt -0.57. Die Kreuze markieren das Ergebnis der evolutiven Optimierung aus 10 Versuchen. Alle Netze gruppieren sich in der Nähe des Maximums der Evidenz. Der mittlere Testfehler dieser 10 Netze beträgt 0.943 bei einem Min-Max Bereich von $[0.935; 0.967]$. Damit liegt die Leistung der Netze nicht wesentlich über dem linearen Modell, das anscheinend schon nahezu optimal ist.

Für den anderen Händler ergibt sich ein deutlicherer Unterschied. Hier liegt das Maximum der Evidenz bei drei Neuronen und der minimale Testfehler bei 7 bis 12 Neuronen. Der mittlere Testfehler sinkt zuerst deutlich ab, bleibt dann relativ konstant und steigt schließlich wieder an. Die besten nicht-linearen Modelle haben einen um etwa 20% kleineren Fehler als das lineare Modell. Die Korrelation von Evidenz und Testfehler über alle 750 Netze beträgt -0.60. Die evolutiv optimierten Netze gruppieren sich vor allem bei drei und vier Neuronen. Sie erreichen einen mittleren Testfehler von 0.50 bei einem Min-Max Bereich von $[0.47; 055]$. Damit sind alle Netze besser als das lineare Modell, im Mittel um 10%.

Nach dieser erfolgversprechenden Voruntersuchung wurde für jeden der 43 stationären Datensätze (von den 50 ausgewählten) die evolutive Optimierung ein einziges Mal angewendet. Im realen Betrieb hat man keine Möglichkeit, ein Ergebnis auszuwählen, da auf allen Daten

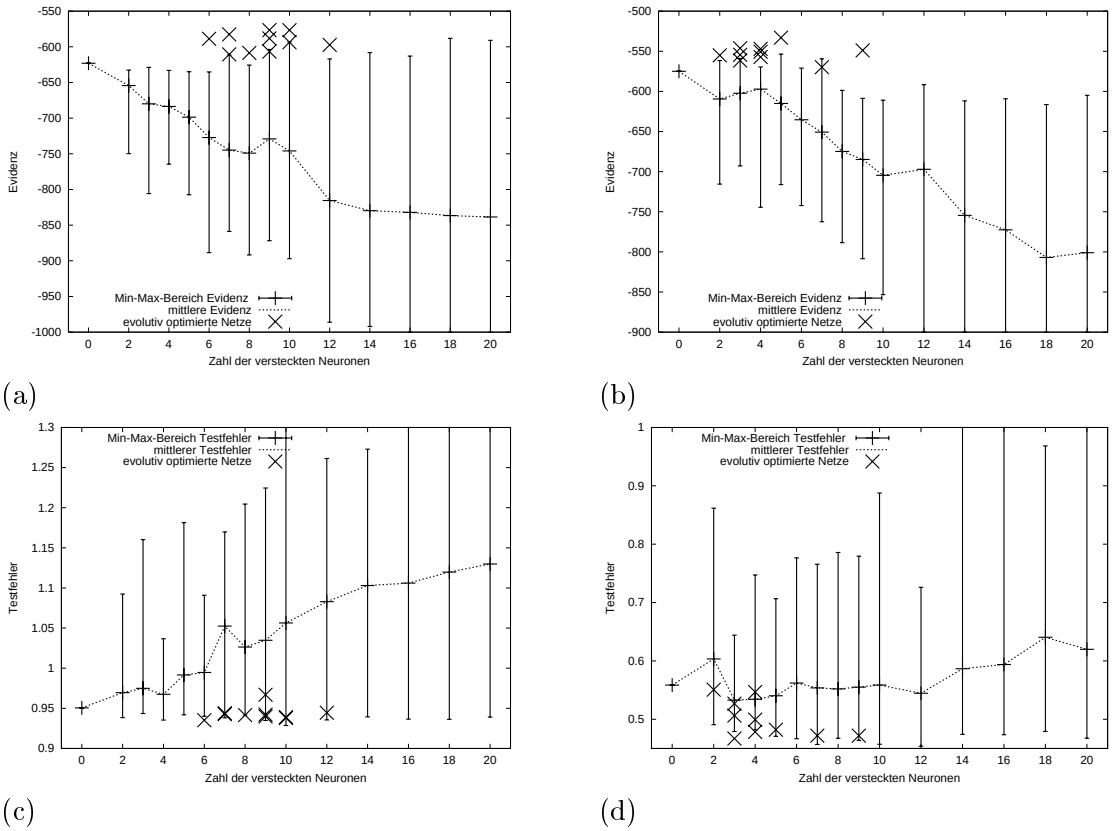


Abbildung 7.11: Die Abbildung zeigt die empirische Abhängigkeit der Evidenz bzw. des Testfehlers von der Zahl an versteckten Neuronen für den Datensatz der Händlers Nr. 0117 und Nr. 1659. Um den Zusammenhang zu ermitteln, wurden pro Topologie 50 Netze, insgesamt also 750 Netze, trainiert. Die Kreuze geben die Ergebnisse für jeweils 10 Versuche der evolutiven Optimierung wieder. Die Netze gruppieren sich wieder sehr schön im Bereich der maximalen Evidenz. a) Mittlere, minimale und maximale Evidenz für Händler Nr. 0117. Die maximale Evidenz wird für ein Netz mit zehn Neuronen erreicht. Die mittlere Evidenz sinkt zuerst kontinuierlich ab und erreicht dann einen Minimalwert. b) Bereich der mittleren und maximalen Evidenz für Händler Nr. 1659. Die maximal Evidenz wird hier für ein Netz mit 5 Neuronen erreicht. Die mittlere Evidenz sinkt ebenso kontinuierlich. c) Mittlerer, minimaler und maximaler Testfehler für Händler Nr. 0117. Der mittlere Testfehler steigt kontinuierlich an. Der minimale Fehler wird von einem Netz mit zehn Neuronen erreicht. Die Korrelation von Evidenz und Testfehler beträgt über alle Netze -0.57. Das beste nicht-lineare Netz liegt nur um 2% unter dem Fehler des linearen Modells. Für Nr. 0117 ergibt sich insgesamt ein eher linearer Zusammenhang. d) Mittlerer, minimaler und maximaler Testfehler für Händler Nr. 1659. Der mittlere Testfehler fällt hier bis zu einer Anzahl von 3 Neuronen. Der minimale Fehler wird von einem Netz mit 7 bzw. 12 Neuronen erreicht. Die Korrelation von Evidenz und Testfehler beträgt über alle Netze -0.60. Dieser Händler zeigt einen deutlichen nicht-linearen Anteil. Das beste nicht-lineare Modell hat einen um 20% geringeren Fehler als das lineare Modell.

trainiert wird. Es ist also wichtig, daß bei einer einmaligen Durchführung im Mittel über alle Händler ein besseres Ergebnis erreicht wird.

Abbildung 7.12 zeigt, daß in einigen Fällen die lineare Lösung deutlich verbessert, für andere der Testfehler aber größer wird. Der mittlere Fehler liegt mit 0.838 signifikant unter dem der linearen Modelle (0.840). Um auf Signifikanz zu testen, wurde geprüft ob die Verteilung der Differenz der Fehler für die einzelnen Händler Mittelwert 0 hat oder nicht. Diese Vor-

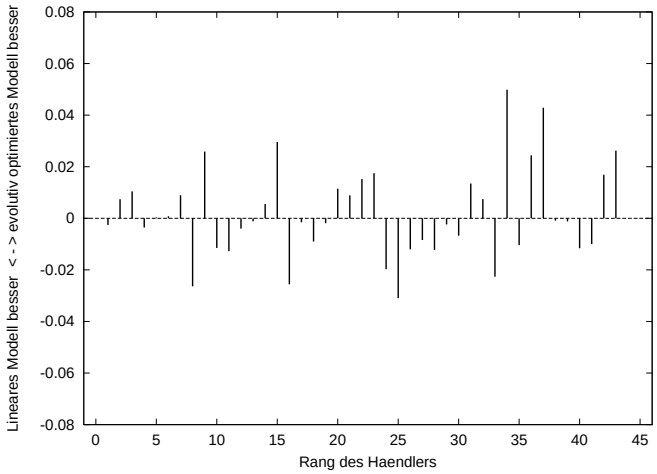


Abbildung 7.12: Die Abbildung vergleicht die durchschnittliche Leistung der linearen Netze mit den evolutiv optimierten Modellen für die 43 stationären der 50 zufällig ausgewählten Datensätze. Ausschläge nach oben zeigen einen Vorteil für das evolutiv optimierte Modell an, nach unten entsprechend für das lineare. Der durchschnittliche Fehler der evolutiv optimierten Netze beträgt 0.838 und liegt leicht unter dem Wert für die linearen Modelle (0.840).

gehensweise wurde gewählt, da die Prognosefehler eine hohe Varianz aufweisen. Das heißt, es gibt Händler mit kleinem und solche mit sehr großem Fehler (vgl. dazu auch Abbildung 7.16b). Es macht keinen Sinn, den t-Test auf den Mittelwerten dieser Fehler auszurechnen, da die entscheidende Frage ist, ob die Veränderungen für jeden Händler signifikant sind. Die Differenzen wie in Abbildung 7.14 geben diese wieder.

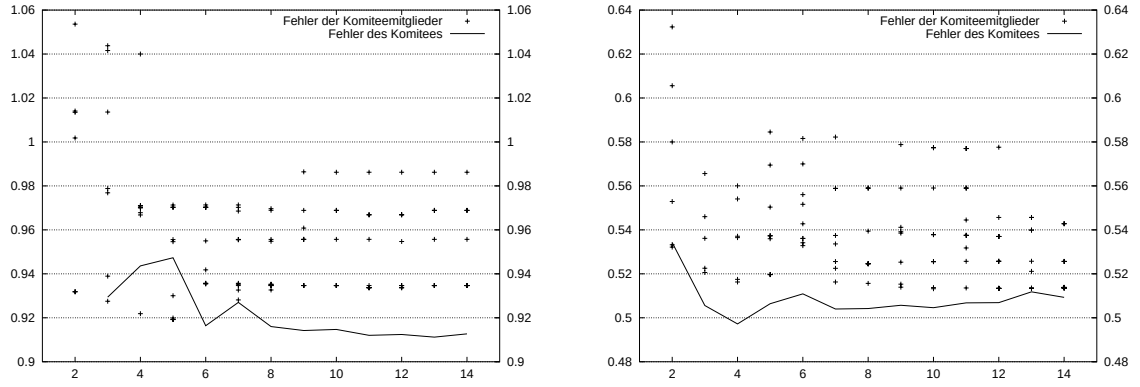
Das schlechtere Abschneiden in einigen wenigen Fällen ist nicht dadurch bedingt, daß die bessere Lösung nicht gefunden werden könnte. Vielmehr ist am Ende der Evolution in der Population eine ganze Auswahl an konkurrierenden Lösungen vorhanden, die eine ähnliche Evidenz, aber einen unterschiedlichen Testfehler haben. In Kapitel 5 zeigte das Beispiel des Sinus-Regressionsproblems bei stärkerem Rauschen ein ähnliches Verhalten. In Kapitel 6.2 wurde gezeigt, daß durch Verwendung eines Komitees von Netzen der Fehler deutlich reduziert werden kann. Diese Technik wird im nächsten Abschnitt auf die Prognose von Absatzzahlen angewendet.

7.4 Prognose mit Komitees von neuronalen Netzen

Die evolutive Optimierung erreichte für einige Zeitreihen einen deutlich geringeren Fehler als die linearen Modelle. Erhofft hätte man sich, daß in jedem Fall das Optimum gefunden wird, ob linear oder nicht-linear. Die Komiteebildung sollte hier weiterhelfen, da es, wie in Abbildung 7.11 für die zwei Beispiele ersichtlich war, offenbar mehrere Maxima der Evidenz gibt, die gefunden werden können.

Die Suchen nach unabhängigen Modellen mit hoher Evidenz wird wieder am Beispiel der beiden Händler Nr. 0117 und Nr. 1659 demonstriert. Die Parameter der Evolution wurden genauso eingestellt wie für das Sinus-Regressionsproblem aus Kapitel 6. Es ist weder nötig noch möglich, diese speziell an die Anwendung anzupassen. Abbildung 7.13 zeigt für die beiden Datensätze die Entwicklung des Komitees während der Evolution. Aufgetragen sind die Fehler der einzelnen Komiteemitglieder und der Komiteefehler über die Generationen. Man kann sehr schön erkennen, wie sich mehrere Netze mit unterschiedlichem Testfehler etablieren. Diese entsprechen gerade wieder Bereichen hoher Evidenz - eine Situation, die

am Beispiel des Sinus-Regressionsproblems bereits ausführlich untersucht wurde. Für beide Händler läßt sich der Fehler gegenüber dem linearen Modell deutlich reduzieren. Im ersten Fall gilt das auch bezüglich der Leistung der evolutiven Optimierung. Man kann also hoffen, daß die Komiteebildung aus unabhängigen Netzen durch die Integration über mehrere mögliche Erklärungen eines Datensatzes im Mittel signifikant besser abschneidet.



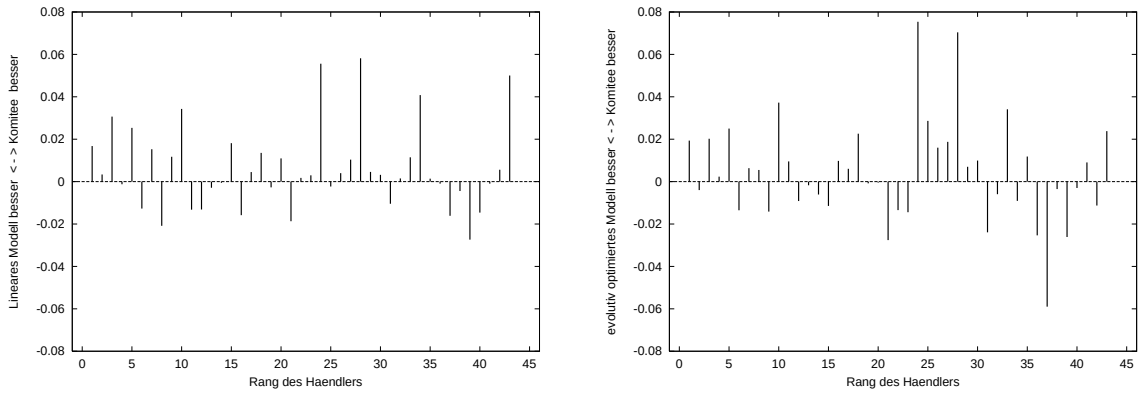
(a) (b)

Abbildung 7.13: Verlauf der Evolution unabhängiger Netze für die beiden Händler Nr. 0117 und Nr. 1659. Die Kreuze geben jeweils die individuellen Fehler der Komiteemitglieder und die Linie den Komiteefehler wieder. (a) Für Nr. 0117 pendelt sich der Komiteefehler ab der 8. Generation deutlich unter dem der einzelnen Netze ein. Die Verbesserung gegenüber dem linearen Modell beträgt ca. 4.4%. (b) Für Nr. 1659 erreicht der Komiteefehler schnell ein niedriges Niveau, das etwa dem Wert des besten individuellen Komiteemitgliedes entspricht. Die Verbesserung gegenüber dem linearen Modell beträgt hier ca. 10%.

Für die 43 Zeitreihen wurde jeweils ein Komitee durch das integrierte Verfahren bestimmt. Die Parametereinstellungen waren für alle Datensätze bis auf eine Ausnahme identisch. Die sortierte Liste für die bevorzugte Löschung von Merkmalen wurde für jeden Händler individuell berechnet. Abbildung 7.14 zeigt die Ergebnisse der Komiteebildung im direkten Vergleich für alle 43 Datensätze mit dem linearen Ansatz (linke Abbildung) und der evolutiven Modelloptimierung.

Die Verbesserung gegenüber dem linearen Modell ist in 4 Fällen größer als 0.04 und in 7 Fällen größer als 0.02. In zwei Fällen gibt es eine Verschlechterung dieser Größenordnung. Gegenüber der evolutiven Modelloptimerung gibt es zwei deutliche Verbesserungen und eine deutliche Verschlechterung. Eine Verbesserung um mindestens 0.02 tritt neunmal auf, eine Verschlechterung fünfmal. Die mittlere Leistung der unabhängigen Komitees ist im Vergleich zu den evolutiv optimierten und den linearen Modellen signifikant besser.

Abbildung 7.15 zeigt den mittleren Fehler über alle 43 Datensätze für verschiedene Optimierungsverfahren und Methoden zur Komiteebildung. Um die Ergebnisse für Boosting und Bagging besser einschätzen zu können, wurden neben der bisher verwendeten Topologie auch lineare Modelle zugrunde gelegt. Für diese schnitten die Verfahren deutlich besser ab. Trotzdem verschlechtert Boosting den Fehler signifikant. Hier zeigt sich ganz deutlich, daß das Verfahren für Daten mit hohem Rauschanteil nicht geeignet ist (vgl. dazu auch (Rätsch *et al.*, 1998)). Liegt ein nicht-lineares Modell zugrunde, dann nimmt der Overfitting-Effekt deutlich zu. Die evolutive Modelloptimierung und Bagging auf Basis linearer Modelle reduzieren den Fehler signifikant gegenüber dem linearen Modell. Für den Vergleich der Komi-

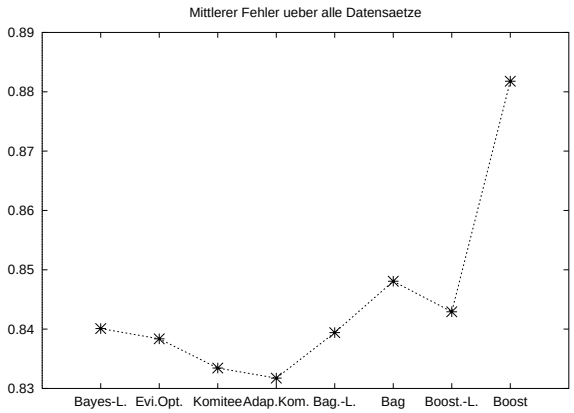


(a) (b)

Abbildung 7.14: Vergleich der Prognoseleistung des Komitees mit dem linearen Modell und der evolutiven Modelloptimierung. Für jeden der 43 Datensätze wurde ein Modell mit jeder Methode generiert. Die Impulse nach oben zeigen einen Vorteil für das Komitee an, im anderen Fall einen Vorteil für die andere Methode. Für beide Vergleiche ist der Mittelwert der Differenzen signifikant größer als Null (t-Test bei Schwelle $t_{0.95;40} = 1.68$) und damit die Komiteebildung günstiger. (a) Vergleich mit den linearen Modellen. Die Komiteebildung bringt in 7 Fällen eine klare Verbesserung (Differenz > 0.02), in zweien eine Verschlechterung. Für 4 Datensätze liegt die Differenz zugunsten des Komitees über > 0.04. Ansonsten liegen die Fehler etwa gleich. (b) Vergleich mit der evolutiven Modelloptimierung. Für 9 Datensätze wird wieder eine klare Verbesserung erreicht, während es aber auch für 5 eine Verschlechterung gibt.

teebildung mit allen anderen Verfahren ist der Mittelwert der Differenzen immer signifikant größer als 0. Damit wurde auch für dieses schwierige Optimierungsproblem eine deutliche Leistungssteigerung realisiert.

Abbildung 7.15: Die Abbildung zeigt Ergebnisse für die Prognose der Absatzzahlen für die verschiedenen Varianten der Komiteebildung bzw. Modelloptimierung. Im Unterschied zu Abbildung 7.10 sind hier nur die stationären Datensätze berücksichtigt. Bagging und Boosting wurden für lineare Modelle und für ein Netz mit 4 versteckten Neuronen verglichen, wie es im Moment im realen Einsatz verwendet wird. Die meisten Zeitreihen haben einen linearen Zusammenhang. Aus diesem Grund ist es sehr schwer, den Mittelwert aller linearen Netze zu unterbieten. Die Verbesserung durch die unabhängigen Komitees ist nach dem t-Test signifikant. Für die evolutive Optimierung und Bagging mit linearen Modellen liegt der Wert knapp über der Schwelle (siehe dazu auch Text). Das Boosting-Verfahren schneidet signifikant schlechter ab. Dies ist auf den hohen Rauschanteil in den Daten zurückzuführen, der durch Boosting nicht richtig modelliert wird.



Die Verbessserung durch Einbeziehung der Konfidenzwerte beträgt knapp 0.2% bezogen auf den mittleren Fehler. Berechnet man wieder den t-Test dafür, ob der Mittelwert der

Differenzen größer als 0 ist, dann wird dieser knapp abgelehnt (1.53 bei einer Schwelle $t_{0.95;40} = 1.68$). Aufgrund des Rauschens und der dadurch implizierten Unsicherheit über die Ausgabe hätte man hoffen können, daß der Zugewinn deutlicher ausfällt. In Kapitel 6.3 wurde am Beispiel des Sinus-Regressionsproblems gezeigt, daß die Verbesserung der evolutiv gefundenen Komiteemitglieder schwierig war, da diese den erzeugenden Prozeß bereits sehr schön approximierten. Ob dies für diese Aufgabe der Fall ist, läßt sich nicht prüfen, wäre aber erstmal zu bezweifeln. Ein weiterer möglicher Grund dafür, daß die Verbesserung gering ausfällt, liegt darin, daß bei dieser Aufgabenstellung der größte Anteil der Prognoseaufgabe bereits durch den speziellen, gleitenden Durchschnitt geleistet wird, zu dem das neuronale Netz nur eine Fehlerkorrektur berechnet. Für die Verwendung der Konfidenzwerte ist es vielversprechend, dem Netz die ganze Aufgabe zu übertragen, d.h. die absoluten Differenzen vorherzusagen. In diesem Fall darf man erwarten, daß die berechneten Konfidenzwerte wesentlich besser mit dem Fehler korreliert sind als für die Fehlerkorrektur. Die Information über die Konfidenzintervalle wird durch den gleitenden Durchschnitt einfach nicht berechnet und geht verloren. Setzt man den Bayes'schen Ansatz bereits auf dieser Ebene ein, dann kann man den vollen Nutzen aus der statistischen Modellierung ziehen. Weitere Gründe, die dafür sprechen, werden im folgenden diskutiert. In jedem Fall lohnt es sich, die Konfindenzwerte zur Kontrolle der Netze während der Arbeitsphase heranzuziehen. Wird die Varianz für ein neues Muster zu groß, dann kann der Anwender eine manuelle Intervention durchführen. Dies geschieht bisher beispielsweise, wenn besondere Ereignisse vorkommen, die nicht prognostizierbar sind, aber zu einem höheren Verkauf führen.

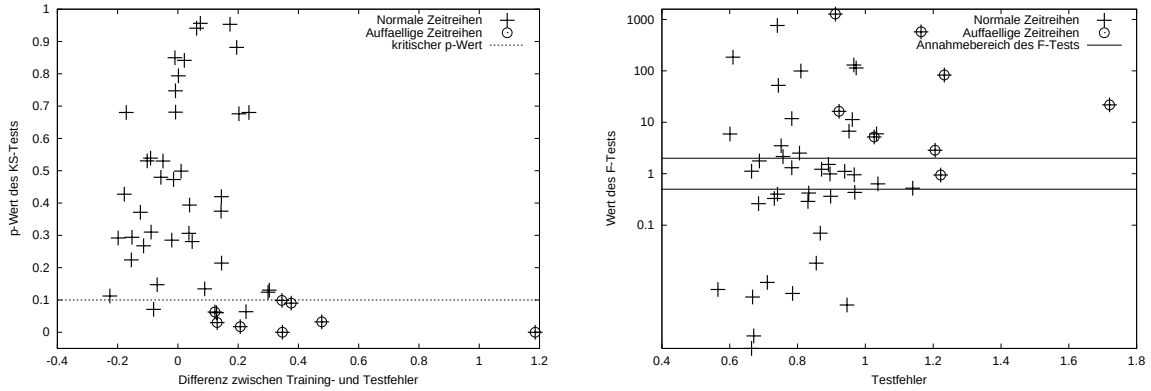
7.5 Weiterführende Fragen

In diesem Abschnitt gehe ich auf die durch die statistischen Tests aufgeworfenen Fragen nochmals ein und zeige, daß die Tatsache, daß einige der Zeitreihen nicht-stationär sind, bei der Prognose unbedingt Beachtung finden muß. Weiterhin werden einige Überlegungen ausgeführt, wie die Systemleistung weiter verbessert werden könnte. Diese beruhen insbesondere darauf, Ähnlichkeiten zwischen verschiedenen Händlern auszunutzen.

Von den ca. 120 Datensätzen aus Abbildung 7.7 war etwas mehr als ein Sechstel (22 Zeitreihen) nicht-stationär. Die Hälfte der nicht-stationären Zeitreihen war derart, daß die Varianz zugenommen hat. Das bedeutet, die Testmuster liegen außerhalb des Trainingsbereichs. Von diesen 11 Zeitreihen waren 7 unter den am Anfang zufällig zum Training ausgewählten. Dies waren die Händler Nr. 1204, 1218, 1248, 1425, 1647, 1841 und 1923. Dazu kommt noch ein Datensatz (Händler Nr. 1585), bei dem 5 von 6 Wochentagszeitreihen nicht-stationär waren.

Die Nicht-Stationarität der Datensätze hängt von dem Verhalten der 6 Einzelzeitreihen ab. Berechnet man die p-Werte der Einzelzeitreihen, dann ist der minimale p-Wert der Einzelzeitreihen und der Testfehler mit $\rho = -0.55$ korreliert. Das heißt, einschneidende Veränderungen an einem Wochentag genügen, um die Prognose insgesamt zu erschweren. Weiterhin ist es aber auch von Bedeutung, wie homogen die p-Werte der 6 Zeitreihen sind. Die Varianz dieser Werte ist mit $\rho = 0.28$ mit dem Testfehler korreliert. Das heißt, je unterschiedlicher das Verkaufsverhalten an den einzelnen Tagen, desto schwieriger ist die Prognose.

In Abbildung 7.16a ist für die linearen Modelle die Differenz zwischen Trainings- und Testfehler gegenüber dem p-Wert des KS-Tests aufgetragen. Die 8 Netze, die auf den bedenklichen Zeitreihen trainiert wurden, sind zusätzlich mit Kreisen markiert.



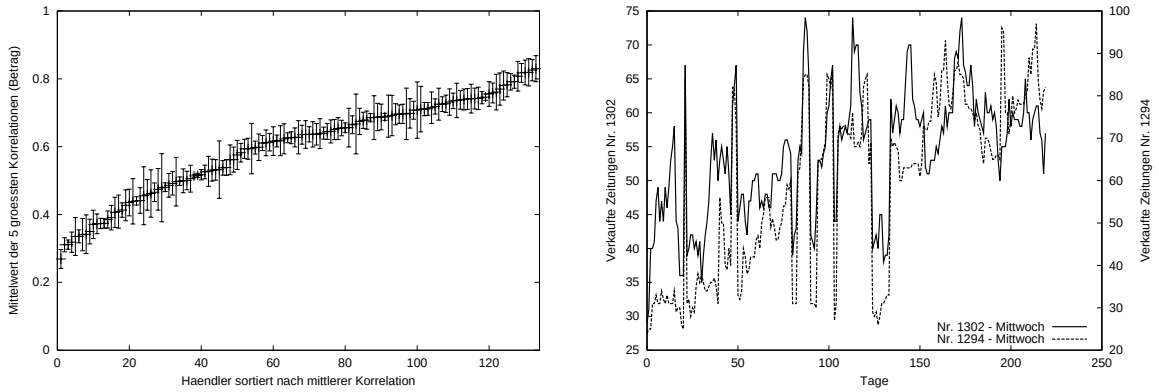
(a) (b)

Abbildung 7.16: Die Abbildung zeigt den Zusammenhang zwischen Stationarität der Zeitreihe und der Leistung der Netze. Die Kreise markieren bis auf einen Fall die Netze, die auf den nicht-stationären Zeitreihen mit zunehmender Varianz trainiert wurden. Für alle diese Netze gilt, daß der Fehler gegenüber der Trainingsmenge deutlich ansteigt. In 5 Fällen liegt er über einem Wert von 1, d.h. die Fehlerkorrektur hat entgegen der Absicht den Fehler vergrößert. Der Ausnahmefall betrifft den Händler Nr. 1585, der als einziger fast nur nicht-stationäre Zeitreihen an den verschiedenen Wochentagen hat. Siehe auch Text. (a) Zusammenhang zwischen Differenz von Trainings- und Testfehler (für die linearen Modelle) und dem p-Wert des KS-Tests. Die nicht-stationären Zeitreihen haben einen zunehmenden Fehler auf der Testmenge. (b) Zusammenhang zwischen dem Testfehler und dem F-Test auf Gleichheit der Varianz. Der Wert des F-Tests ist logarithmisch aufgetragen. Die beiden Linien umfassen den Annahmebereich des Tests. Ein großer Testfehler tritt immer dann auf, wenn die Varianz auf den Testdaten ansteigt. Es sind dann vor allem für 'schwierige' Muster Prognosen zu machen.

Für die meisten Netze gilt, daß die Differenz zwischen Trainingsfehler und Testfehler in dem Intervall $[-0.2; 0.2]$ liegt. Für fast alle Zeitreihen mit kleinem p-Wert liegt die Differenz allerdings über 0. Die Zeitreihen, die mit einem Kreis markiert sind, erfüllen gleichzeitig noch die Bedingung, daß die Varianz der Daten zunimmt. In Abbildung 7.16b befinden sie sich alle in der rechten oberen Ecke, d.h. aus der Nicht-Stationarität mit zunehmender Varianz folgt, daß der Testfehler größer als 1 ist. In diesen Fällen ist es günstiger, auf die Fehlerkorrektur durch das neuronale Netz (oder ein anderes Modell) zu verzichten. Stattdessen verwendet man als Prognose den gleitenden Durchschnitt. Im anderen Fall, wenn die Varianz kleiner geworden ist, liegt auch der Testfehler immer unter 1. Bei dem markierten Netz, dessen Varianz konstant geblieben ist, handelt es sich um den Händler Nr. 1585. Daß er trotz niedrigem p-Wert eine konstante Varianz hat, liegt daran, daß diese für manche der Wochentagszeitreihen zugenommen, für andere abgenommen hat.

Nachdem nun Ursachen für ein schlechtes Prognoseverhalten bestimmt sind, wende ich mich im folgenden möglichen Verbesserungsmöglichkeiten zu. Die erste betrifft die Tatsache, daß die Zeitreihen der verschiendenen Händler zum Teil stark miteinander korreliert sind. Abbildung 7.17 zeigt auf der linken Seite den Mittelwert und Min-Max Bereich für die fünf stärksten Korrelationen eines Händlers. Der Mittelwert über alle Händler liegt hier bei 0.60 und streut zwischen 0.3 und 0.85. In jedem Fall ist es vielversprechend, diese Information für die Prognose zu nutzen. Dazu können auf Basis der Korrelationen neue Merkmale defi-

niert werden. Günstiger ist es aber, die Zielwerte der stark korrelierten Händler mit in die Ausgabe zu nehmen. Für diesen Multi-Tasking Ansatz benötigt man die Daten der anderen Händler dann nur während der Trainingsphase und nicht im Realbetrieb. Auf der rechten Seite von Abbildung 7.17 sind zwei Verkaufszeitreihen gezeigt, die zu zwei unterschiedlichen Händlern gehören. Die Korrelation beträgt etwa 0.8.



(a) (b)

Abbildung 7.17: Die Abbildung zeigt die Korrelationen zwischen den Verkaufszeitreihen der Einzelhändler. (a) Es wurden zu jedem Datensatz die fünf anderen bestimmt, die die größte Korrelation hatten. Von diesen fünf Werten ist der Mittelwert und der Min-Max Bereich aufgetragen. Die Datensätze sind der besseren Übersicht wegen nach dem Mittelwert sortiert. Der Mittelwert streut zwischen 0.3 und 0.8. (b) Die Abbildung zeigt die Verkaufszeitreihe der Händler Nr. 1302 (linke y-Achse) und Nr. 1294 (rechte y-Achse). Die Korrelation beträgt 0.8.

Wie wir bereits gesehen haben, ist es von grundlegender Bedeutung, daß sich die einzelnen Zeitreihen ähnlich verhalten. Weicht das Verkaufsverhalten an einem Wochentag völlig von den anderen ab, dann führt das zu Schwierigkeiten bei der Prognose. In Abbildung 7.18 ist die Ähnlichkeit zwischen den Einzelzeitreihen für alle Händler dargestellt. Für manche Händler ist das Verkaufsverhalten an allen Tagen gleich, während es bei einigen völlig unterschiedlich ist. Negative Korrelationen kommen äußerst selten vor. Die mittlere Korrelation über alle Händler beträgt ca. 0.46. In jedem Fall ist diese Information, wie sich der Verkauf an den anderen Tagen verhalten hat, nützlich für eine Prognose. Diese Information wird inzwischen in Merkmalen kodiert zum Training der Netze verwendet. Auch hier könnte man versuchen, ein Multi-Tasking Modell zu entwickeln. Die Zielwerte der anderen Tage bilden dann gerade die zusätzlichen Ausgaben. Dadurch erreicht man, daß sich in den versteckten Neuronen allgemeinere Merkmale herausbilden, die weniger durch das (unbekannte) Rauschen in den Zielwerten beeinflußt sind.

Andererseits kann man aufgrund der Unterschiedlichkeit der einzelnen Zeitreihen auch vermuten, daß der Ansatz, pro Händler nur ein Netz zu trainieren, Probleme aufwirft. Die Entscheidung dafür oder dagegen ist nicht offensichtlich. Durch die Skalierung liegen alle Zeitreihen auf einem Niveau, was die Verschiedenartigkeit eventuell relativiert. Weiterhin hat man den Vorteil, wesentlich mehr Muster zum Training zur Verfügung zu haben. Inwieweit es sich positiv oder negativ auswirkt, alle Zeitreihen eines Händlers mit einem Netz zu lernen, wird Gegenstand weiterer Untersuchungen im laufenden Projekt sein.

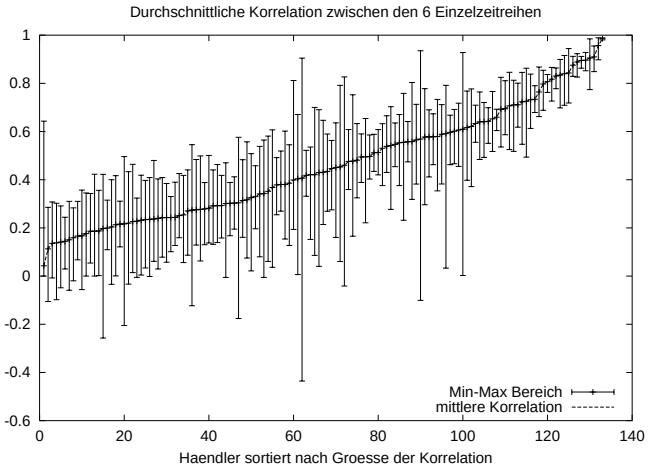


Abbildung 7.18: Die Abbildung zeigt die Korrelation zwischen den Einzelzeitreihen jedes Händlers. Dazu wurden alle Paare der Einzelzeitreihen gebildet und die 15 Werte berechnet. Gezeigt ist der Mittelwert dieser Werte sowie der Min-Max Bereich. Es gibt zum Teil starke Korrelationen zwischen den einzelnen Zeitreihen. Andererseits gibt es aber auch starke Schwankungen. Diese Unterschiede sind zum Teil für eine schlechte Prognoseleistung verantwortlich.

7.6 Zusammenfassung

Die Prognose von Absatzzahlen für viele Verkaufsstellen ist ein interessantes und facettenreiches Problem, bei dem vielfältige Schwierigkeiten auftreten, wie sie nur eine reale Anwendung bietet.

In diesem Fall ist nicht nur eine Lösung für einen Datensatz zu finden, sondern für viele gleichzeitig. Eine Methode, die bei Händler A funktioniert mag bei Händler B das Gegenteil bewirken. Aufgrund der Vielzahl der Händler ist es unmöglich, Parameter so zu justieren, daß immer bei allen bessere Ergebnisse erreicht werden. In diesem Sinne ist eine Methode gut, wenn sie robust ist. Das heißt, nachdem man einmal die Parameter fixiert hat, wird sie auf alle Zeitreihen angewendet und sollte dann im Mittel bessere Ergebnisse bringen.

Voraussetzung dafür ist natürlich, daß die Zeitreihen prognostizierbar sind. Um dies zu gewährleisten, wurden die Daten mehreren statistischen Tests unterworfen. Die erste Untersuchung sollte zeigen, daß es sich bei den Differenzen-Zeitreihen nicht um einen Zufallsprozeß handelt. Mit einem Iterationstest konnte belegt werden, das dies im allgemeinen ausgeschlossen werden kann. Der Kolmogorow-Smirnow-Test wurde verwendet, um zu prüfen, ob die Verteilungen, die den Trainings- und Testdaten zu Grunde liegen, identisch sind. War das nicht der Fall, dann wurde in einem zweiten Schritt geprüft, ob die Varianz der Testdaten größer als auf dem Trainingszeitraum ist. In diesem Fall kann mit Sicherheit keine sichere Prognose geleistet werden, da das neuronale Netz dabei extrapolieren muß. Ein neuronales Netz (oder auch ein anderes Modell) sollte dann nicht zur Prognose verwendet werden, sondern nur der gleitende Durchschnitt.

Aufbauend auf den Ergebnissen von Kapitel 5 konzentrierte sich der nächste Schritt der Optimierung auf die Selektion von Eingabemerkmalen. Nach einer visuellen Vorverarbeitung wurden auf Basis der Rückwärtssuche mit Korrelationskoeffizienten weitere 24 Merkmale entfernt. Eine Mittelung über alle Datensätze diente dazu, eine einheitliche Kodierung beizubehalten, statt für jeden Händler eine spezielle Merkmalskombination zu favorisieren. Dies ist möglich, weil die Korrelationen hoch sind, d.h. es geht keine Information verloren. Es ist sinnvoll, weil es die Pflege des Systems und die nachfolgenden Untersuchungen extrem

vereinfacht. Für den verbleibenden 15-dimensionalen Vektor brachte der Einsatz des Mutual Information Kriteriums eine weitere Reduktion um 5 Merkmale, so daß im folgenden eine einheitliche Kodierung mit 10 Merkmalen verwendet werden konnte. Netze, die mit der kleineren Kodierung trainiert wurden, schnitten signifikant besser ab als solche, die alle Merkmale zur Verfügung hatten.

Der besondere Charme dieses Problems liegt darin, daß man keine optimale Netzgröße bestimmen kann, die für alle Händler Gültigkeit hat. Vielmehr hat jeder Datensatz seine eigene Charakteristik. Neben einfachen linearen Zusammenhängen treten auch komplexe auf. Es ist also vielversprechend, die Methode der evolutiven Optimierung anzuwenden. Die durchschnittliche Leistung läßt sich damit leicht steigern. Trotzdem gibt es auch Datensätze, bei denen die Prognose gegenüber dem linearen Modell schlechter wird. Im allgemeinen wird nicht nur eine *einzige* Lösung gefunden. Vielmehr gibt es meistens mehrere Netze mit ähnlicher Evidenz, aber deutlich verschiedenem Testfehler. Eine Situation, die an das Sinus-Regressionsproblem mit hohem Rauschanteil erinnert. Es bleibt festzuhalten, daß die evolutive Optimierung für die meisten Zeitreihen ein Optimum gefunden hat. Außer Bagging mit linearen Modellen und der Komiteebildung aus unabhängigen Netzen zeigten alle anderen Methoden einen deutlichen Overfitting-Effekt. Es wurden Zusammenhänge gelernt, die durch die Daten offenbar nicht gestützt waren.

Die eben beschriebene Sachlage ist geradezu prädestiniert, durch Komiteebildung die auftretenden Unsicherheiten bei der Modellfindung zu reduzieren. So konnte für die meisten Zeitreihen die Leistung der einzelnen Modelle deutlich übertroffen werden. Bildet man die Differenzen gegenüber allen anderen Verfahren wie in Abbildung 7.14, dann ergibt sich, daß der Mittelwert signifikant größer als 0 ist. Weiterhin waren durch Einbeziehen der Konfidenzwerte weitere Leistungssteigerungen möglich. Für eine Kontrolle der Modelle während der Arbeitsphase ist die Berechnung und Auswertung derselben in jedem Fall empfehlenswert.

Abschließend wurden noch Möglichkeiten aufgezeigt, wie man die Gesamtleistung des Systems weiter verbessern könnte. Neben der Feststellung, daß man für die nicht-stationären Zeitreihen am besten die Fehlerkorrektur nicht verwendet, die das neuronale Netz für den gleitenden Durchschnitt berechnet, hat sich gezeigt, daß es zwischen verschiedenen Händlern zum Teil hohe Korrelationen im Verkaufsverhalten gibt. Diese Information könnte man in Merkmale kodiert dem Netz zur Verfügung stellen. Eine günstigere Lösung wäre aber, ein Multi-Tasking Modell zu verwenden, bei dem die stark korrelierten Zeitreihen die zusätzlichen Ausgaben liefern. Das vermeidet einerseits eine Vergrößerung des Eingaberaumes, und andererseits spart es den Aufwand, die Daten im Einsatzfall aufbereiten zu müssen, da die zusätzlichen Ausgaben nicht ausgewertet werden.

Kapitel 8

Zusammenfassung und Ausblick

8.1 Zusammenfassung

Die vorliegende Arbeit diskutiert die Lösung von Problemen durch Komitees neuronaler Netze, die durch einen integrierten Optimierungsprozeß gewonnen wurden. Dabei bedeutet *integriert*, daß wesentliche Entwurfsaspekte in der Optimierungsmethodik Berücksichtigung finden. Dazu gehören

- die geeignete Merkmalsselektion zur Vermeidung hochdimensionaler Eingaberäume,
- automatische Regularisierung durch Bayes'sches Lernen,
- Optimierung der Zahl der versteckten Neuronen,
- Reduktion der Datenabhängigkeit durch Sampling und Komiteebildung,
- die gezielte Suche nach möglichst unabhängigen Netzen zur Komiteebildung anstelle der Auswahl eines einzelnen 'besten' Modells.

Die Entwurfsmethodik ist weitestgehend unabhängig von der verwendeten Modellklasse, hier also den neuronalen Netzen. Aufgrund des modularen Aufbaus kann jede andere Methode der Funktionsapproximation verwendet werden, solange sie nur ein analytisches Gütekriterium zur relativen Bewertung der Modelle bereitstellt, entsprechend der Evidenz im Bayes'schen Ansatz.

Die beiden zentralen Bausteine dieser Arbeit bilden die *evolutive Modelloptimierung* und die *Clusterung der Modelle* auf Basis ihrer Ähnlichkeit. Grundlegend hierfür ist die Dekomposition des Komiteefehlers in eine Art 'Bias-Term' und einen 'Varianz-Term' in schöner Analogie zum Bias-Varianz Dilemma.

Der erste Term, der durchschnittliche Fehler der Netze, sollte möglichst klein sein, d.h. jedes Netz sollte bezüglich seiner Lernaufgabe möglichst optimal sein. Daraus folgt, daß auf der

Ebene der Modelloptimierung sowohl Regularisierung als auch die Optimierung des Eingaberaumes und der Topologie wesentliche Bestandteile sein müssen. Bishop merkt an, daß die Optimierung der Merkmalsstruktur und der Modelle idealerweise zusammen durchgeführt wird, dies aber in der Praxis aus Aufwandsgründen oft nicht möglich ist (Bishop, 1995; S. 305). Mit der hier vorgestellten Methode - Bayes'sches Lernen, Merkmalsselektion und Topologieoptimierung in einem Evolutionsprozeß effizient zu verzahnen - wird erstmals ein Werkzeug vorgestellt, daß diese Aufgabe für neuronale Netze leistet. Dabei kommt der Prozeß, und das ist wichtig, ohne Kreuzvalidierung aus. Stattdessen wird die Evidenz, die das Bayes'sche Verfahren berechnet, als Fitneßkriterium in der evolutionären Suche verwendet. Das Verfahren sucht systematisch nach Modellen mit einer hohen a posteriori Wahrscheinlichkeit, ohne den Aufwand gegenüber dem normalen Training mit der Bayes'schen Methode zu erhöhen.

Der zweite Term mißt die 'Streuung' der Netze um das Komitee und sollte gemäß der Dekomposition möglichst groß werden, um den Generalisierungsfehler des Komitees zu minimieren. Für diese Optimierungsaufgabe leitete ich ein Kriterium her, das unter anderem besagt, daß der Varianzterm maximal wird, wenn die einzelnen Netze stochastisch unabhängig sind (Gleichung (3.11)). In das Kriterium fließt auch die Anzahl der Netze ein. Entscheidend ist deshalb, daß die stochastische Unabhängigkeit für möglichst viele Modelle möglichst groß wird. Um aus einer gegebenen Menge von Modellen diejenigen auszuwählen, die das obige Kriterium maximieren, wird ein hierarchisches, agglomeratives Clusterverfahren eingesetzt. Ob man die geeigneten Modelle zur Komiteebildung gezielt suchen kann, bleibt auf dieser Ebene zunächst offen.

Tatsächlich lassen sich beide Bausteine elegant in die evolutionäre Suche einbetten. Anstatt die Suche zunehmend auf das Netz maximaler Evidenz zu fokussieren, werden mehrere vielversprechende Suchpunkte verfolgt und beibehalten. Dazu werden die Eltern vor der Selektion zur Erzeugung der Nachkommen mittels der Clusterung in Gruppen zusammengefaßt. Das Ranking der Netze erfolgt innerhalb der Gruppen nach ihrer Evidenz. Für jede Gruppe werden gemäß ihrem Anteil an der Population Nachkommen generiert. Auf zwei Ebenen besteht jetzt ein Selektionsdruck: Innerhalb einer Klasse pflanzen sich die Individuen wahrscheinlicher fort, die eine höhere Evidenz haben. Andererseits ergeben *unabhängigere* Suchrichtungen wahrscheinlicher eine eigene Klasse. Für ein Netz wird dadurch die Fortpflanzungswahrscheinlichkeit erhöht. Da die Zahl der Eltern immer etwas kleiner ist als die Zahl der Nachkommen, werden die schlechtesten Modelle kontinuierlich aussortiert. Durch diese Reduktion vor der Clusterung wird verhindert, daß sich Modelle mit niedriger Evidenz in der Population etablieren können, wenn sie nur hinreichend von allen anderen verschieden sind. Am Ende des Prozesses liegen Netze mit hoher Evidenz vor, die maximal voneinander unabhängig sind.

Ist das Komitee zusammengestellt, dann läßt sich durch die Konfidenzintervalle, die im Bayes'schen Ansatz berechnet werden, für jedes Muster die Gewichtung der Mitglieder adaptiv berechnen. Jedes Netz wirkt im (umgekehrten) Verhältnis zu seiner *Varianz* bzw. seiner *Verlustwahrscheinlichkeit* für das aktuelle Muster an der Ausgabe des Komitees mit.

8.2 Vorgehensweise und Ergebnisse

Für die Approximation einer Funktion auf Basis empirischer Daten wird in dieser Arbeit ein integriertes Konzept vorgeschlagen, das ein Komitee aus mehreren neuronalen Netzen generiert. Im folgenden sollen die einzelnen Bausteine, die im Rahmen dieses Konzeptes entwickelt wurden, anhand der beim Modellentwurf auftretenden Teilprobleme beschrieben und die Ergebnisse eingeordnet werden (siehe auch nochmals Abbildung 1.5).

Modelloptimierung

Bevor man anfängt, Modelle zu trainieren, sollte zuerst eine geeignete Merkmalsauswahl getroffen werden. Dazu wurden in dieser Arbeit basierend auf dem Korrelationskoeffizienten und dem Infomationsgehalt zwei Algorithmen bereitgestellt, die durch eine iterative Rückwärtssuche die Zahl der Merkmale kontinuierlich verkleinern. Die optimale Größe und Kombination ist so nicht zu ermitteln. Es läßt sich aber der Suchraum deutlich einschränken, wie an der Schilddrüsenklassifikation und später in Kapitel 7 an der Prognose von Absatzzahlen demonstriert wurde. In beiden Fällen hatten die kleineren Modelle eine höhere Generalisierungsleistung. Durch Experimente wurde weiterhin belegt, daß sich die Information, die durch die Vorverarbeitung gewonnen wurde, zur Initialisierung der Population für die evolutive Suche nutzen läßt. Auch dadurch ließ sich der mittlere Fehler signifikant verringern.

Beim Bayes'schen Lernen wird die Gewichtung des Regulariserungsterms durch einen iterativen Algorithmus automatisch berechnet. In jeder Iteration werden zuerst die Hyperparameter optimal eingestellt und dann ein Minimum der aktuellen Fehlerfunktion gesucht. In Kapitel 4 wurde vorgeschlagen, die globale Suche nach guten Modellen mit dem Bayes'schen Ansatz zu verzahnen. Die Evidenz wird dazu in einem evolutionären Algorithmus als Fitneßwert verwendet und kann damit als Leitfaden durch den Suchraum betrachtet werden. Basierend auf einer Plausibilitätsüberlegung und Experimenten konnte gezeigt werden, daß diese Verzahnung am besten so erfolgt, daß pro Suchschritt, d.h. pro Generation, zwei Anpassungsschritte der Hyperparameter erfolgen. Die Suchschritte werden durch Mutationsoperatoren berechnet, die weitgehend auf statistischen Vefahren aufbauen, wie sie auch schon zur Vorverarbeitung eingesetzt wurden.

Für zwei Benchmarkprobleme, die durch eine mit Rauschen überlagerte Sinusfunktion gewonnen wurden, konnte gezeigt werden, daß mit der evolutiven Suche genau die Bereiche hoher Evidenz gefunden werden. Durch diese Verzahnung wird der Parameterraum wesentlich effizienter durchsucht. Die Leistung der evolutiv optimierten Modelle liegt deshalb auch signifikant höher, als wenn man die Netze konventionell trainiert und ein Modell auswählt.

In Kapitel 5.2 wurde für mehrere Anwendungen ein empirischer Zusammenhang zwischen Evidenz bzw. Testfehler und der Zahl an versteckten Neuronen sowie der Zahl der Merkmale hergestellt. Die Ergebnisse wurden als Höhenliniengrafik aufbereitet, anhand derer zu erkennen war, daß sich Bereiche hoher Evidenz und hoher Generalisierungsleistung weitestgehend decken. Für das Add10-Regressionsproblem und die Schilddrüsenklassifikation war die Korrelation nahezu perfekt. Umgekehrt wurde aus den weniger starken Korrelationen für die beiden anderen Beispiele geschlossen, daß die kleinen Testmengen nur begrenzt eine

Schätzung der Güte zulassen. Nach Berechnungen aus der Statistik sind für eine Datenmenge von etwa 500 Mustern maximal 4 bis 6 Merkmale zu verwenden. In diesen Bereichen lag auch das Maximum der Evidenz. Die evolutive Modelloptimierung findet durch Mutation der Eingabeneuronen und der versteckten Neuronen für alle Anwendungen die Bereiche mit hoher Evidenz. Die resultierenden Netze hatten eine signifikant höhere Leistung als mit konventionellem Training zu erreichen gewesen wäre. Mit diesen Untersuchungen wurde auch empirisch nachgewiesen, daß die Zahl der Merkmale für praktische Anwendungen in der Regel zu groß gewählt wird und der Einsatz eines Optimierungsverfahrens in jedem Fall sinnvoll ist.

Betrachtet man den Verlauf der Suche genauer, dann zeigt sich, daß mit zunehmender Annäherung an die optimale Kombination von Merkmalen und der Zahl an versteckten Neuronen die Leistung deutlich zunimmt. Die evolutive Suche folgt auch in diesem komplexen Suchraum einem Gradientenpfad zum Maximum. Für Anwendungen wie in Kapitel 7 ist so eine Eigenschaft besonders wichtig. Der Algorithmus findet nahezu jedesmal ein vernünftiges Modell für einen gegebenen Datensatz. Mit der evolutiven Modelloptimierung wurde erstmals eine Methode bereitgestellt, die für neuronale Netze die Optimierung der Merkmalsstruktur und der Modellparameter zusammen und automatisch durchführt, ohne daß dabei, z.B. durch Kreuzvalidierung und menschliche Interaktion, Parameter bestimmt werden müssen. Dies ist für die Absatzprognose besonders wichtig, da hier für verschiedene Zeitreihen lineare oder auch nichtlineare Modelle geeignet sein können.

Abschließend wurde in Kapitel 5 gezeigt, daß die Modelloptimierung an ihre Grenzen stößt, wenn das Rauschen in den Daten stark wird. Durch die zunehmende Unsicherheit gewinnen andere Modelle an Plausibilität gegenüber dem eigentlich erzeugenden Prozeß. Die Korrelation zwischen Evidenz und Testfehler nimmt dann ab, und es ist schwerer, das beste Modell anhand des Gütekriteriums zu bestimmen.

Komiteebildung durch Clusterung

In Kapitel 3 wurde ausgehend von der Bias-Varianz Dekomposition des Komiteefehlers ein Kriterium hergeleitet, das man optimieren sollte, wenn man aus einer gegebenen Menge von Modellen ein möglichst leistungsfähiges Komitee bilden möchte. Da es im allgemeinen unmöglich ist, den Wert des Kriteriums für alle Kombinationen an Netzen zu berechnen, wurde vorgeschlagen, diese Aufgabe durch ein agglomeratives hierarchisches Clusterverfahren zu lösen. Dazu wird zuerst die Ähnlichkeitsmatrix berechnet, die für alle Paare von Netzen ihre stochastische Abhängigkeit enthält. Jedes Netz bildet zuerst eine eigene Klasse. Durch fortwährendes Verschmelzen der beiden ähnlichsten Klassen erhält man jeweils eine neue Klasseneinteilung, für die der Wert des Kriteriums (3.12) berechnet wird. Das Verfahren wird abgebrochen, wenn der Wert des Kriteriums maximal groß ist. Die Klassenzahl wird also automatisch bestimmt und nicht etwa durch den Benutzer vorgegeben.

In Kapitel 6 wurde an einem Beispiel gezeigt, daß der Generalisierungsfehler des zugehörigen Komitees minimal wird, wenn das Kriterium seinen Maximalwert erreicht. Basierend auf dieser Technik kann man also bereits auf einfache Art und Weise Komitees bilden, die eine deutliche Leistungssteigerung ermöglichen.

Evolution unabhängiger Modelle

Die evolutive Entwicklung eines Komitees muß die beiden wichtigen Aspekte - Optimierung der Evidenz und Maximierung der Unabhängigkeit - berücksichtigen. Um beide Ziele in einer Selektionsstrategie für den evolutiven Prozeß gemeinsam zu verfolgen, werden zuerst die potentiellen Eltern der nächsten Generation auf Basis ihrer Evidenz bestimmt. Bevor jetzt Nachkommen gebildet werden, wird das Clusterverfahren auf die Population angewendet. Dadurch ergeben sich mehrere Gruppen. Innerhalb einer Gruppe werden die Individuen nach ihrer Evidenz angeordnet. In Kapitel 4 wurden einige Möglichkeiten vorgeschlagen, wie die Selektion von Netzen nun im Detail aussehen kann, um beiden Zielen Rechnung zu tragen.

Die Experimente in Kapitel 6 zeigten dann, daß es nicht sinnvoll ist, 'unabhängigere Klassen' bei der Selektion zu bevorzugen. Auch eine zu starke Einengung auf das jeweils beste Netz einer Klasse verschlechtert die Ergebnisse. Die günstigste Strategie war, aus jeder Klasse anteilmäßig Nachkommen zu generierern und dabei die Netze mit höherer Evidenz zu bevorzugen.

Die Klassen bilden gewissermaßen die Suchrichtungen in jeder Generation. Es zeigte sich, daß diese auch nicht zu sprunghaft gewechselt werden sollten. Vielmehr ist es günstig, das Prinzip 'Groß variieren, aber klein vererben' anzuwenden. Das heißt, wenn es eine starke Veränderung der Klassenzahl gibt, dann bricht man die Clusterung nicht beim maximalen Wert des Kriteriums ab, sondern verändert die letzte Klassenzahl leicht in die Richtung des neuen Wertes und bildet so die neuen Suchrichtungen. Damit wird für eine gleichmäßigere Suche gesorgt, die durch zufällige Ausschläge in eine Richtung nicht völlig vom Weg abgebracht wird. Die Parameter der evolutiven Suche wurden an dem Sinus-Regressionsproblem aus Abbildung 1.1b eingestellt und dann für alle Experimente übernommen. Eine spezielle Anpassung an eine Problemstellung ist im allgemeinen nicht nötig.

Verstärkt man das Rauschen oder ändert man die zugrundeliegende Verteilung, dann stößt die evolutive Modelloptimierung an ihre Grenzen, da es dann meist mehrere verschiedene Erklärungen gibt. Die Suche nach unabhängigen Modellen schneidet hier wesentlich besser ab. Dies liegt darin begründet, daß das Kollabieren auf eine Lösung verhindert wird und mehrere verschiedene Bereiche gleichzeitig durchsucht werden. Dadurch werden Modelle mit hoher Evidenz gefunden, die sich gegenseitig bestmöglichst ergänzen. Für die Anwendungen, die wir in Kapitel 5 betrachtet haben, wurde ebenfalls das integrierte Konzept angewendet. Insgesamt kann man feststellen, daß die gemeinsame Optimierung von Evidenz und Unabhängigkeit ein geeignetes Mittel ist, um die Diversität in der Population in sinnvoller Weise aufrecht zu erhalten und geeignete Modelle zur Komiteebildung zu finden.

Bezieht man für die Komiteemitglieder noch ihre Verlustwahrscheinlichkeiten mit ein, dann läßt sich die Gewichtungsmatrix des Komitees für jedes Muster adaptiv berechnen. An einem Beispiel wurde die Wirkungsweise demonstriert und für einige Datensätze gezeigt, daß sich damit weitere Leistungssteigerungen des Komitees realisieren lassen.

Reale Anwendung - Prognose von Absatzzahlen

Das siebte Kapitel widmete sich einer konkreten Anwendung. Basierend auf historischen Daten soll der Absatz an Zeitungen für jeden Einzelhändler für den nächsten Verkaufstag

vorhergesagt werden, um die gedruckten Exemplare möglichst optimal verteilen zu können. Anhand dieser Anwendung sollen die vorgeschlagenen Methoden auf ihre Leistungsfähigkeit und Grenzen hin untersucht werden. Die Besonderheit der Anwendung liegt darin, daß Prognosemodelle für eine Vielzahl von Zeitreihen zu lernen sind, von denen jede ihre eigene Charakteristik hat. Das macht es unmöglich, die Parameter des Verfahrens speziell an eine Zeitreihe anzupassen, so daß damit bessere Ergebnisse erzielt werden. Vielmehr läßt sich durch die Vielzahl an Datensätzen eine Leistungssteigerung durch die Methoden an sich belegen. Die automatische Modelloptimierung findet mit wenigen Ausnahmen die optimale Lösung. Es hat sich allerdings gezeigt, daß ähnlich wie bei dem Sinus-Regressionproblem für manche Zeitreihen unterschiedliche Modelle der Daten gefunden werden, die aber alle eine ähnliche Evidenz haben. Mit zunehmender Generationenzahl fokussiert die Evolution aber die Suche auf ein Modell, das am Ende ausgewählt wird. In diesen Fällen werden auch regelmäßig Netze mit höherem Testfehler ausgewählt werden.

An dieser Stelle kommen die Vorzüge der Suche nach unabhängigen Modellen voll zum Tragen. Statt ein einzelnes Netz zu favorisieren, werden mehrere Entwicklungslinien gleichmäßig verfolgt. Durch Bildung eines Komitees aus den unabhängigsten Netzen mit hoher Evidenz können die Nachteile der evolutiven Modelloptimierung kompensiert werden. Die Generalisierungsleistung des Komitees war selten schlechter als die des einzelnen Modells, und wenn, dann nur unwesentlich. Dafür war der Fehler aber mehrfach deutlich geringer. Die Lösung der Aufgabe, für eine Vielzahl von Verkaufszeitreihen automatisch ein geeignetes Modell zu bestimmen, wird durch die integrierte Optimierung ermöglicht.

8.3 Ausblick

Das vorgeschlagene integrierte Optimierungskonzept zur Entwicklung von Komitees darf nicht als Maschine verstanden werden, die, wenn sie einmal gestartet ist, am Ende die optimale Lösung für ein Problem ausgibt. Vielmehr ist es ein Werkzeug, das dem Entwickler zum einen durch Automatisierung Arbeitsaufwand erspart, zum anderen aber Overfitting vermeidet, weil es ausschließlich auf analytischen Optimierungskriterien basiert. Ein menschlicher Entwickler wird bei iterativen Entwurfsprozessen, die sich auf Kreuzvalidieren und Testen stützen, immer geneigt sein, *günstige* Parameter für den Testfall zu verwenden. Von der Einstellung solcher Parameter wird der Entwickler entlastet.

Das Werkzeug ist auch kein Allheilmittel, das immer eingesetzt werden sollte. Das Beispiel der Prognose von Absatzzahlen für viele Händler zeigt, daß der Entwickler immer die Anwendung im Auge haben sollte. In diesem Fall lohnte es sich, für die gegebene Kodierung zunächst darauf zu verzichten, ein möglichst optimales Modell bzw. ein Komitee zu entwickeln. Stattdessen war es günstiger, zuerst eine einheitliche Eingabestruktur zu suchen und erst dann die evolutive Suche zu starten. Weiterhin ist es auch wichtig, die Modellierung der Aufgabe immer wieder auf den Prüfstand zu stellen, statt nur auf die Mächtigkeit der Werkzeuge zu vertrauen. Für die Prognose der Absatzzahlen können beispielsweise zusätzlich die Korrelationen zwischen den Händlern ausgenutzt werden. Die Kreativität des Menschen ist im Entwurfsprozeß weiterhin die wichtigste Komponente, um die gegebenen Werkzeuge an den geeigneten Stellen mit Gewinn einzusetzen.

Mit der vorgestellten Methodik zur Problemlösung durch Komitees neuronaler Netze konnte die Zielsetzung, wesentliche Entwurfsaspekte bei der Modellentwicklung in einem integrierten Konzept zu behandeln, realisiert werden. In mehreren Experimenten wurde gezeigt, daß damit eine automatische Entwicklung von Komitees möglich ist, die den Anwender von der interaktiven Behandlung wesentlicher Probleme entlastet. Damit kann die Kreativität des Anwenders auf die Bereiche konzentriert werden, in denen maschinelles Lernen keine Hilfe leisten kann.

Ein Defizit evolutionärer Strategien, aber auch heuristischer Suchverfahren im allgemeinen, ist oft das Fehlen eines klaren Abbruchkriteriums für die Suche. Dieses Problem ist bei der Vielzahl von zu untersuchenden Kombinationsmöglichkeiten unvermeidbar. Für andere Verfahren wird oft ein Abbruchkriterium durch Kreuzvalidierung oder nur willkürlich definiert. In meinem Ansatz habe ich die Generationenzahl festgelegt, indem ich den Aufwand der normalen Vorgehensweise als Vergleichsgröße zu Grunde gelegt habe. Da die Optimierung bei verrauschter Zielfunktion kurzfristig auch Verschlechterungen in Kauf nehmen sollte, ist bei der Auswahl am Ende des Prozesses darauf zu achten, daß er in einen nahezu stabilen Zustand bezüglich des Optimierungskriteriums (3.14) konvergiert ist. Dies ist beispielsweise dann der Fall, wenn sich die Werte für die Evidenz nicht mehr wesentlich ändern. Ebenso sollten die Vertreter der Klassen beständig erhalten bleiben. Ist dies nicht der Fall, dann ist die Generationenzahl zu gering oder die Mutationswahrscheinlichkeit zu hoch gewählt. Im letzten Fall werden die Suchpunkte beständig verworfen und neue festgelegt, so daß sich kein Gleichgewicht einstellen kann.

Ein wesentlicher Fortschritt in meiner Arbeit entstand durch die Herleitung des Heterogenitätskriteriums (3.14). Ähnliche Kriterien, die heuristischer Natur waren, brachten nicht den gewünschten Erfolg. Vielversprechendes Potential birgt deshalb nach meiner Auffassung vor allem die theorethische Weiterentwicklung der Komiteebildung insbesondere in einem wahrscheinlichkeitstheoretischen Kontext. Einen wichtigen Schritt in dieser Richtung bildet die Arbeit von Heskes (Heskes, 1998). Er definiert die Bias-Varianz Dekomposition über Verteilungen und die Kullback-Leibler-Distanz. Auf dieser Basis könnte es möglich sein, das Optimierungskriterium für die Komiteebildung auf einer allgemeineren Ebene zu betrachten und dabei andere Fragestellungen miteinzubeziehen. Zum einen wäre die Berechnung einer a posteriori Wahrscheinlichkeit für das Komitee durch einen Bayes'schen Ansatz von prinzipiellem Interesse. Auf der momentanen Betrachtungsebene dürfte das nicht ohne weiteres zu lösen sein. Läßt sich ein Komitee anhand seiner Evidenz bewerten, dann kann man die Suche weiter verzweigen, indem man mehrere Komiteelösungen gleichzeitig verfolgt.

Ebenso kann man versuchen, die Gewichtung der Komiteemitglieder in die Suche miteinzubeziehen. Ein erster Ansatzpunkt wäre, hier die Methodik des *Stacked Generalization* (Wolpert, 1992) in die evolutive Optimierung zu integrieren. Die Gewichtsmatrix für das Komitee wird dann ebenfalls trainiert. Prinzipiell bleibt die Einbeziehung der Gewichtung eine offene Frage, die für grundlegende Fortschritte Raum läßt.

Ein wichtiges Einsatzgebiet ganz anderer Art ist die Kombination des Bayes'schen Ansatzes mit Reinforcement-Lernen, wie wir es für evolutionäre Algorithmen bereits vorgeschlagen haben (Braun & Ragg, 1997). Die automatische Regularisierung ermöglicht es, auch kleine Mustermengen, die Sequenzen von Zuständen entsprechen, einzutrainieren, ohne sich diesen zu speziell anzupassen. Das könnte Instabilitäten des bisherigen Lernprozesses entgegenwirken, wie wir sie zum Beispiel beim Strategielernen beobachtet haben (Ragg *et al.*, 1995).

Aufbauend auf dieser Kombination ist dann das ganze Instrumentarium der evolutiven Modelloptimierung als Basis verfügbar, um neuronale Netze für Reinforcement-Probleme zu entwickeln.

Eine wichtige Eigenschaft des vorgestellten Konzeptes ist die weitestgehende Unabhängigkeit von der verwendeten Modellklasse. Die Operatoren zur Merkmalsselektion und zur Clusterung betrachten nur die Beziehungen zwischen Daten, ohne auf Eigenschaften der neuronalen Netze Rückgriff zu halten. Die Einbeziehung anderer Modellklassen, wie z.B. *Support Vector Machines*, aber auch anderer Methoden der Clusteranalyse, um unabhängige Suchrichtungen zu bestimmen, sind Gegenstand aktueller Arbeiten, die im Rahmen des DFG-Projekts *'Integrierte Entwicklung von Komitees neuronaler Netze'* untersucht werden.

Die einzelnen Bausteine des integrierten Verfahrens, wie z.B. die Merkmalsselektion, automatische Regularisierung, aber auch Clusterung von ähnlichen Funktionen, konnten bei der Prognose von Absatzzahlen in Kooperation mit dem Axel Springer Verlag erfolgreich eingesetzt werden. Die Prognosen werden dazu genutzt, Zeitungen möglichst optimal zu verteilen, so daß die Remissionsquote, d.h. der Anteil der nicht verkauften Zeitungen, minimiert wird, ohne daß die Verkaufsstellen beständig ausverkauft sind. Die vorgestellten Methoden konnten für das Verkaufsgebiet 'Münster' eine signifikante Verbesserung erreichen. Für die meisten Zeitreihen dieses Gebietes ist allerdings eine lineare Lösung optimal. Dies liegt darin begründet, daß viele Verkaufsstellen einen relativ gleichmäßigen Absatz haben. Im Gegensatz dazu liegt bisher im Gebiet 'Berlin' die Remissionsquote deutlich über der von Münster. Das Verkaufsverhalten ist hier wesentlich unregelmäßiger. Die automatische Bestimmung der optimalen Netzkomplexität und Merkmalskombination und darauf aufbauend die Komiteebildung könnte hier eine deutliche Verbesserung bringen. Die Evaluation der Methoden wird im laufenden Projekt auf diese Zeitreihen ausgeweitet.

Anhang A

Anhang

A.1 Statistische Testverfahren

Im folgenden sind die verwendeten statistischen Tests kurz zusammengefaßt. Für eine ausführliche Darstellung siehe (Büning & Trenkler, 1994).

A.1.1 t-Test auf Gleichheit zweier Erwartungswerte

Gegeben seien m bzw. n unabhängige Zufallsvariablen $X_1, \ldots, X_m; Y_1, \ldots, Y_n$, wobei $X_i \sim \mathcal{N}(\mu, \sigma^2)$ $(i = 1, \ldots, m), Y_j \sim \mathcal{N}(\nu, \sigma^2)$ $(j = 1, \ldots, n)$.

Getestet wird die Nullhypothese

$$H_0 : \mu = \nu \tag{A.1}$$

gegen die Alternative

$$H_1 : \mu \neq \nu. \tag{A.2}$$

H_0 wird genau dann abgelehnt, wenn

$$|T_{m,n}| \geq t_{1-\alpha/2} \tag{A.3}$$

gilt, wobei T wie folgt definiert ist:

$$T := \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{(m-1)S_X^2 + (n-1)S_Y^2}{m+n-2}\left(\frac{1}{m} + \frac{1}{n}\right)}} \tag{A.4}$$

mit

$$S_X^2 := \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \tag{A.5}$$

und S_Y^2 entsprechend. t_p bezeichnet das p-te Quantil der t-Verteilung mit $m+n-2$ Freiheitsgraden. In (Büning & Trenkler, 1994) sind die p-Quantile der t-Verteilung tabelliert.

A.1.2 F-Test auf Gleichheit zweier Varianzen

Gegeben seien wieder m bzw. n unabhängige Zufallsvariablen $X_1, \ldots, X_m; Y_1, \ldots, Y_n$, wobei $X_i \sim \mathcal{N}(\mu, \sigma_1^2)$ $(i = 1, \ldots, m), Y_j \sim \mathcal{N}(\nu, \sigma_2^2)$ $(j = 1, \ldots, n)$.

Getestet wird die Nullhypothese

$$H_0 : \sigma_1^2 = \sigma_2^2 \tag{A.6}$$

gegen die Alternative

$$H_1 : \sigma_1^2 \neq \sigma_2^2. \tag{A.7}$$

H_0 wird genau dann abgelehnt, wenn $T < F_{\alpha/2}$ bzw. $T > F_{1-\alpha/2}$ gilt, wobei

$$T := \frac{S_X^2}{S_Y^2}. \tag{A.8}$$

mit S_X, S_Y gemäß Gleichung (A.5). F_p bezeichnet das p-te Quantil der F-Verteilung mit $m-1$ und $n-1$ Freiheitsgraden (Büning & Trenkler, 1994).

A.1.3 Iterationstest auf Zufälligkeit

Der bekannteste Test auf Zufälligkeit ist der sogenannte Iterationstest. Unter einer Iteration versteht man die Folge von einem oder mehreren identischen Symbolen, denen entweder ein anderes oder kein Symbol unmittelbar vorangeht oder folgt. Ein Beispiel für eine Ausprägung der Symbole wäre *Steigt* und *Fällt* in einer Differenzenzeitreihe für positive bzw. negative Werte der Differenz. Bei zufälliger Reihenfolge ist anzunehmen, daß sich die beiden Ausprägungen des Merkmals weder ganz regelmäßig abwechseln (viele Iterationen), noch daß zuerst nur *Fällt* auftritt und dann *Steigt* (wenig Iterationen).

Getestet wird die Nullhypothese

H_0: Die Folge der Beobachtungen an Steigt/Fällt ist zufällig.

gegen die Alternative

H_1: Die Folge ist nicht zufällig.

Die Anzahl R an Iterationen kann durch die Normalverteilung approximiert werden (Büning & Trenkler, 1994). H_0 wird abgelehnt, wenn $|Z| \geq z_{1-\alpha/2}$ ist, wobei die Prüfgröße Z wie folgt definiert ist:

$$Z := \frac{R - 2\alpha n(1-\alpha)}{2\sqrt{n}\alpha(1-\alpha)} \tag{A.9}$$

wobei n_1, n_2 die Anzahl der beiden Ausprägungen angibt und $n = n_1 + n_2$ und $n_1 = \alpha n$ ist.

A.1.4 Kolmogorow-Smirnow-Test auf Gleichheit zweier Verteilungen

Der Kolmogorow-Smirnow-Test ist ein Anpassungstest. Er basiert auf der maximalen Differenz zwischen empirischer und hypothetischer Verteilungsfunktion. Für einen Test auf Gleichheit zweier Verteilungen wählt man als Testkriterium die maximale Differenz der beiden empirischen Verteilungsfunktionen.

Es seien $x_1, \ldots, x_m$ und $y_1, \ldots, y_n$ zwei Stichproben aus Grundgesamtheiten mit stetigen Verteilungsfunktionen F bzw. G. Mit F_m und G_n werden die empirischen Verteilungsfunktionen bezeichnet. Es gilt

$$F_m(z) = \begin{cases} 0 & : \quad z < x_{(1)} \\ i/m & : \quad x_{(i)} \leq z < x_{(i+1)}, i = 1, .., m-1 \\ 0 & : \quad z \geq x_{(m)} \end{cases}$$

und G_n entsprechend.

Getestet wird die Nullhypothese

H_0: $F(z) = G(z)$ für alle $z \in R$

gegen die Alternative

H_1: $F(z) \neq G(z)$ für ein $z \in R$.

H_0 wird abgelehnt, wenn $K_{m,n} = \max_z |F_m(z) - G_n(z)| > k_{1-\alpha}$. Eine Tabellierung der kritischen Werte $k_{1-\alpha}$ findet sich in (Büning & Trenkler, 1994).

A.2 Verkaufszahlen von Zeitungen

Im folgenden sind beispielhaft für die beiden Händler (Nr. 0117 und Nr 1659), die in Kapitel 7 ausführlicher untersucht wurden, alle relevanten Zeitreihen für alle Wochentage abgebildet.

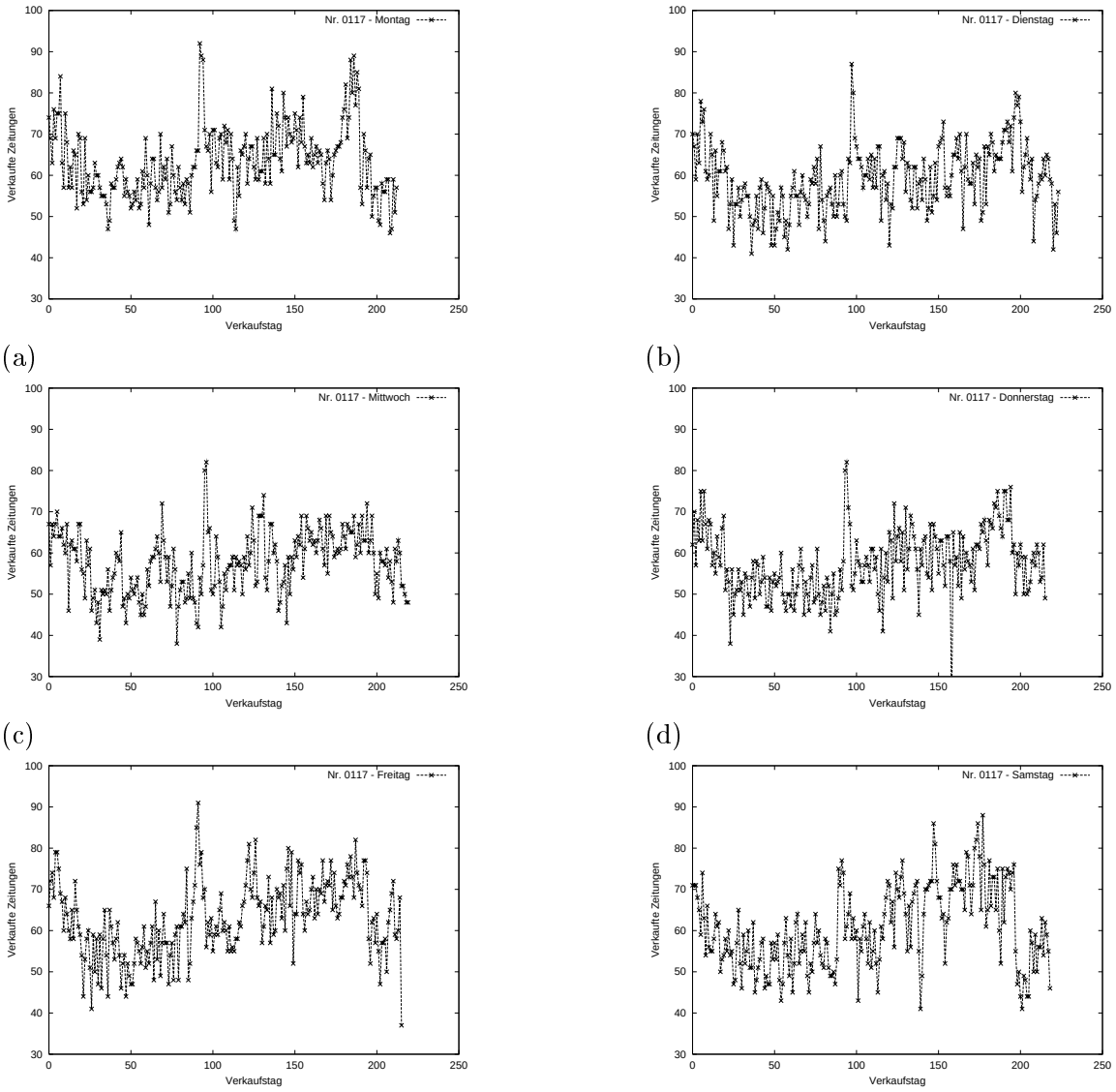


Abbildung A.1: Die Abbildung zeigt die absoluten Verkaufszahlen von Herbst 1992-1996 für den Einzelhändler mit Nr. 0117 aus dem Gebiet 'Münster' an den verschiedenen Wochentagen. (a)-(f) = Montag-Samstag.

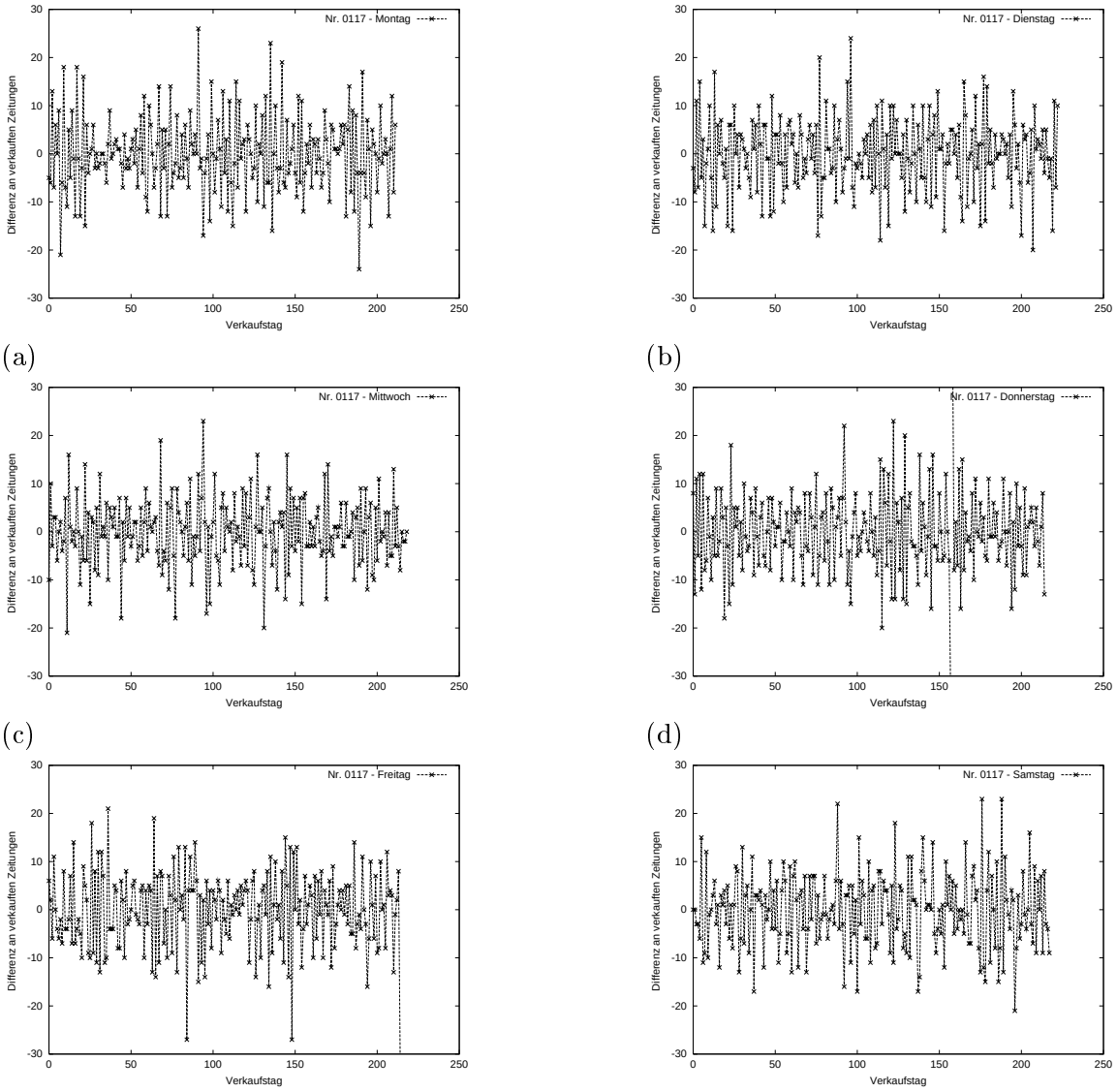


Abbildung A.2: Die Abbildung zeigt die absoluten Differenzen der Verkaufszahlen von Herbst 1992-1996 für den Einzelhändler mit Nr. 0117 aus dem Gebiet 'Münster' an den verschiedenen Wochentagen. Die Differenz wird jeweils zu der Verkaufszahl am gleichen Tag in der nächsten Woche gebildet, beispielsweise von Montag zu Montag. (a)-(f) = Montag-Samstag.

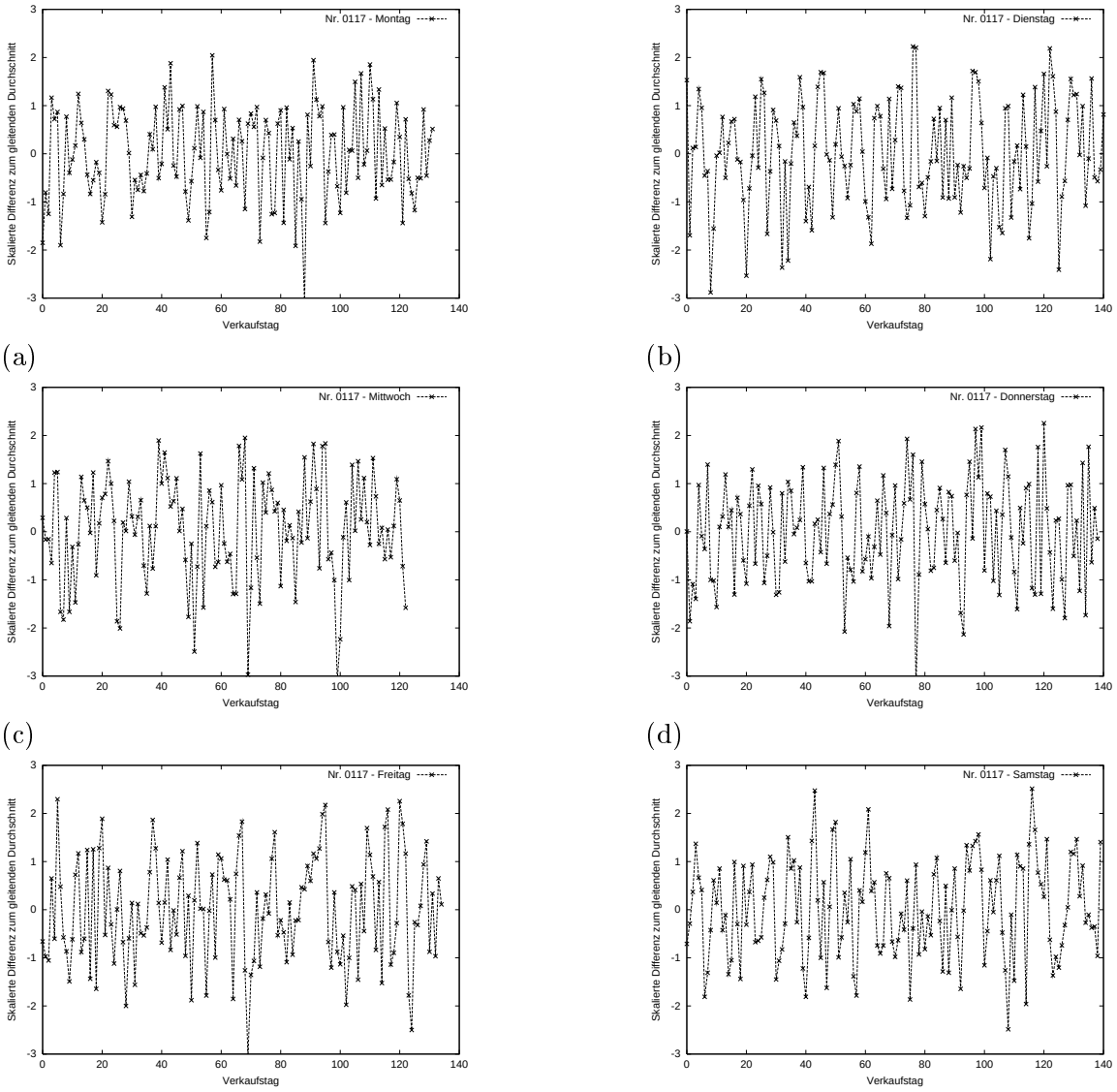


Abbildung A.3: Die Abbildung zeigt die Differenz zu dem gleitenden Durchschnitt von Herbst 1992-1996 für den Einzelhändler mit Nr. 0117 aus dem Gebiet 'Münster' an den verschiedenen Wochentagen. Diese Zeitreihe wird mit den neuronalen Netzen prognostiziert, d.h. die Netze entsprechen einer Art Fehlerkorrekturmodell für den gleitenden Durchschnitt. (a)-(f) = Montag-Samstag.

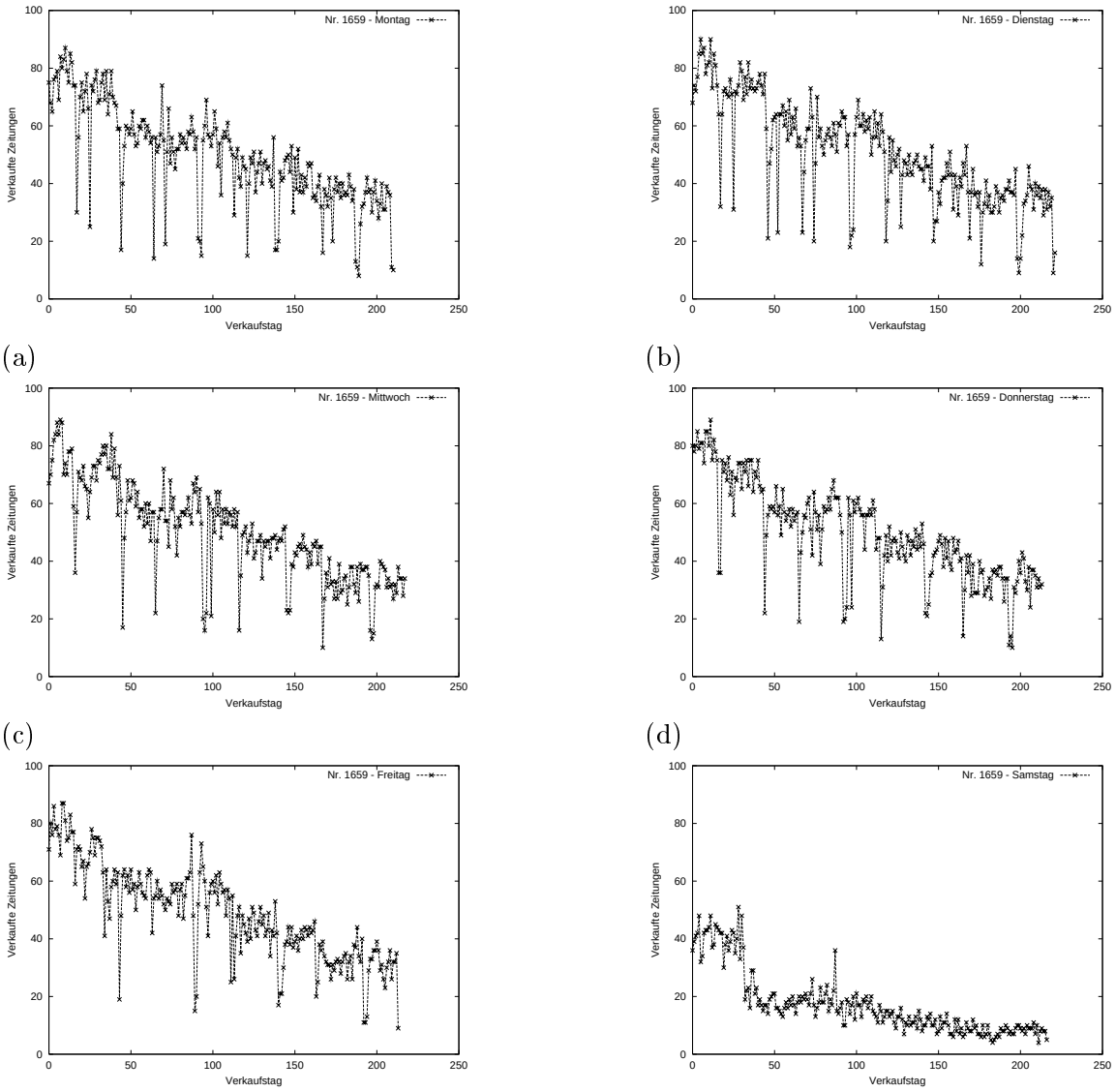


Abbildung A.4: Die Abbildung zeigt die absoluten Verkaufszahlen von Herbst 1992-1996 für den Einzelhändler mit Nr. 1659 aus dem Gebiet 'Münster' an den verschiedenen Wochentagen. (a)-(f) = Montag-Samstag.

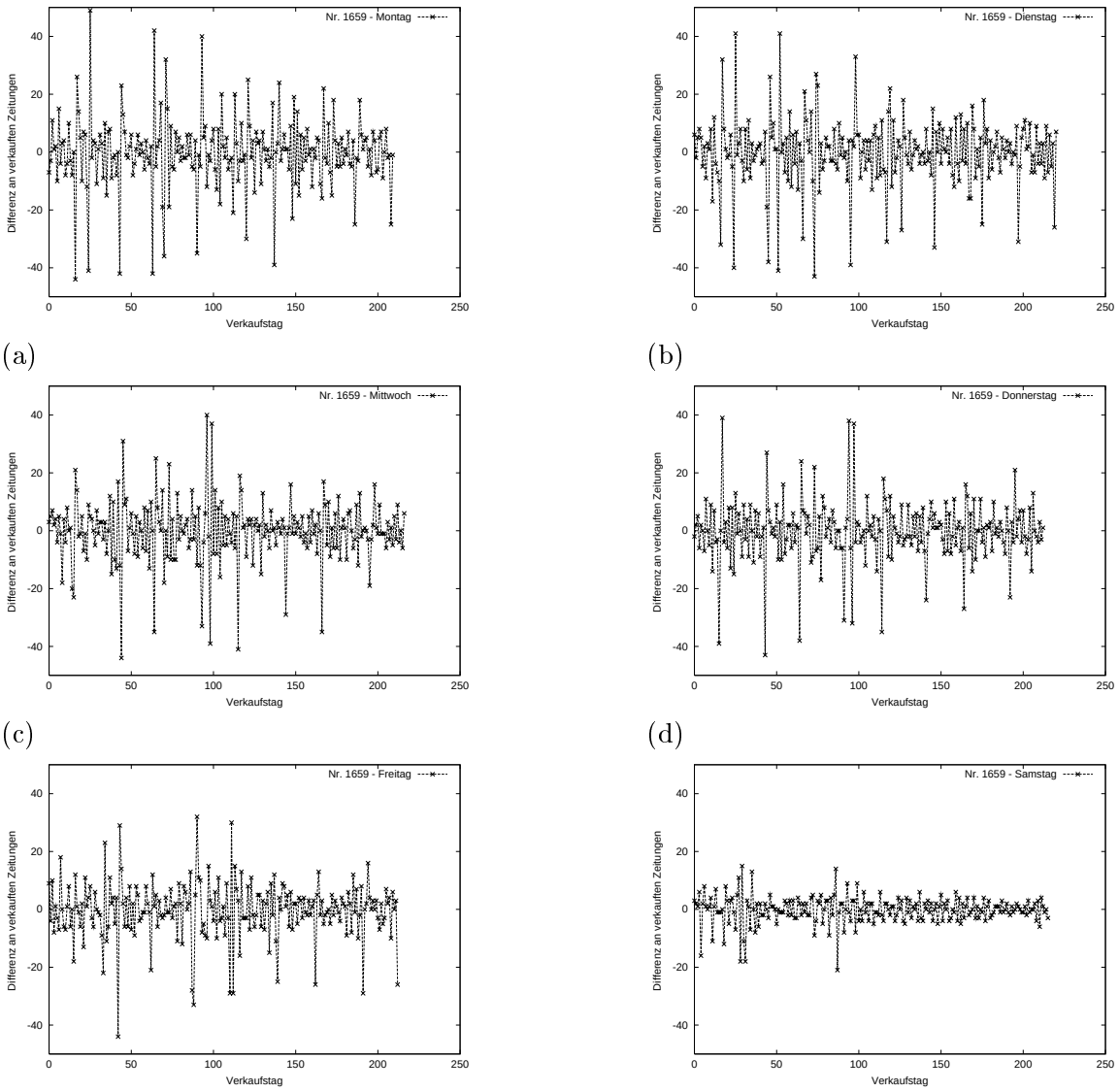


Abbildung A.5: Die Abbildung zeigt die absoluten Differenzen der Verkaufszahlen von Herbst 1992-1996 für den Einzelhändler mit Nr. 1659 aus dem Gebiet 'Münster' an den verschiedenen Wochentagen. Die Differenz wird jeweils zu der Verkaufszahl am gleichen Tag in der nächsten Woche gebildet, beispielsweise von Montag zu Montag. (a)-(f) = Montag-Samstag.

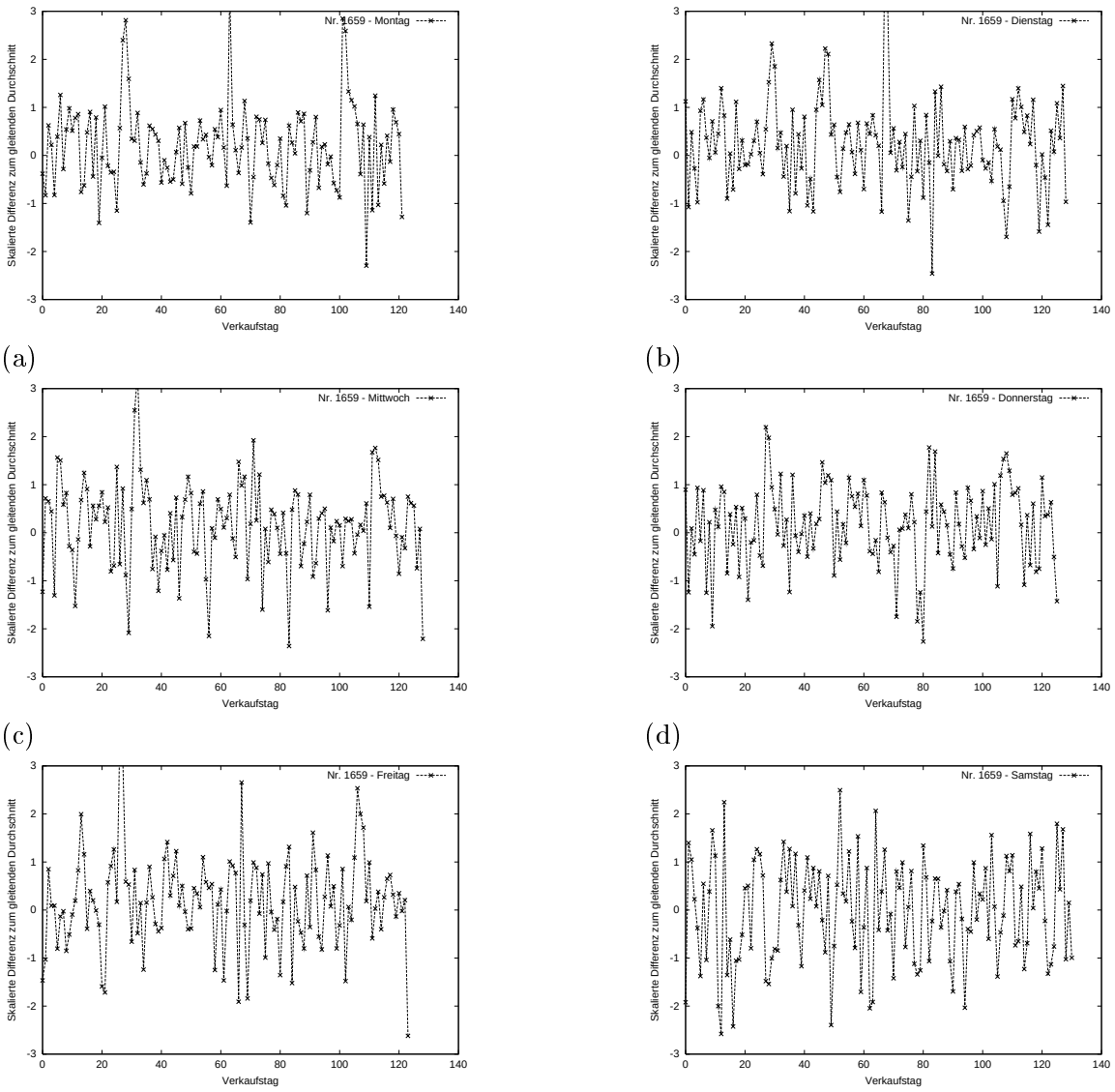


Abbildung A.6: Die Abbildung zeigt die Differenz zu dem gleitenden Durchschnitt von Herbst 1992-1996 für den Einzelhändler mit Nr. 1659 aus dem Gebiet 'Münster' an den verschiedenen Wochentagen. Diese Zeitreihe wird mit den neuronalen Netzen prognostiziert, d.h. die Netze entsprechen einer Art Fehlerkorrekturmodell für den gleitenden Durchschnitt. (a)-(f) = Montag-Samstag.

Literaturverzeichnis

Alander, Jarmo T. (1996) An Indexed Bibliography of Genetic Algorithms and Neural Networks. Technical Report 94-1-NN, Department of Information Technology and Production Economics, University of Vaasa.

Bäck, Thomas (1996) *Evolutionary Algorithms in Theory and Practice.* Oxford University Press.

Badcock, Christopher (1999) *Psychodarwinismus: die Synthese von Darwin und Freud.* Hanser Verlag.

Barber, David and Christopher Bishop (1998) Ensemble Learning for Multi-Layer Networks. In *Advances in Neural Information Processing Systems 10*, pages 395–401.

Baum, E.B. and D. Haussler (1989) What size net gives valid generalization? *Neural Computation*, 1:151–160.

Beck-Bornholdt, Hans-Peter and Hans-Hermann Dubben (1998) *Der Hund, der Eier legt: Erkennen von Fehlinformation durch Querdenken.* Reinbek bei Hamburg : Rowohlt.

Berger, J. O. (1980) *Statistical decision theory and Bayesian analysis.* Springer Verlag.

Bishop, Christopher M. (1995) *Neural Networks for Pattern Recognition.* Oxford Press.

Bishop, Christopher (1998) Variational Learning in Graphical Models and Neural Networks. In *Perspectives in Neural Computing: Proceedings of the ICANN 98*, volume 1, pages 13–22.

Bonnlander, B.V. and A.S. Weigend (1994) Selecting Input Variables Using Mutual Information and Nonparametric Density Estimation. *Proceedings of the ISANN '94, Taiwan*, pages p.42–50.

Box, George E. P., Gwilym M. Jenkins, and Gregory C. Reinsel (1994) *Time series analysis : forecasting and control.* Prentice Hall.

Braun, Heinrich and Thomas Ragg (1996a) ENZO – Evolution of Neural Networks, User Manual and Implementation Guide, http://i11www.ira.uka.de. Technical Report 21/96, Universität Karlsruhe.

Braun, Heinrich and Thomas Ragg (1996b) Evolution Neuronaler Netze. In *Konnektionismus und Neuronale Netze, Beiträge zur Herbstschule HeKoNN96, Münster/Wetsfalen, 1996*, GMD-Studien Nr.300, ISBN 3-88457-300-4, pages 209–230.

Braun, Heinrich and Thomas Ragg (1997) Evolutionary Optimization of Neural Networks for Reinforcement Learning Algorithms. In *Lecture Notes in Computer Science,*

Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms 1997, Norwich, UK, pages 385–389. Springer.

Braun, Heinrich and Joachim Weisbrod (1993) Evolving feedforward neural networks. *Proceedings of the ICANNGA93.*

Braun, Heinrich (1997) *Neuronale Netze: Optimierung durch Lernen und Evolution.* Springer, Heidelberg.

Breiman, L. (1996) Bagging predictors. *Machine Learning*, 24:123–140.

Breiman, L. (1999) Combining Predictors. In Sharkey, Amanda J.C., editor, *Combining Artificial Neural Nets*, pages 31–50. Springer.

Buhmann, J. (1997) Stochastic algorithms for exploratory data analysis: Data clustering and data visualization. In Jordan, M., editor, *Learning in Graphical Models.* Kluwer-Academics.

Büning, Herbert and Götz Trenkler (1994) *Nichtparametrische statistische Methoden.* de Gruyter.

Büning, Herbert (1991) *Robuste und adaptive Tests.* de Gruyter.

Burges, Christopher (1998) A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2:121–168.

Carlin, B. P. and T. A. Louis (1996) *Bayes and Empirical Bayes Methods for Data analysis.* Chapman & Hall.

Caruana, R. (1996) Algorithms and Applications for Multitask Learning . In *Proceedings of 13th International Conference on Machine Learning*, pages 87–95.

Cover, T.M. and J.A. Thomas (1991) *Elements of Information Theory.* Wiley Series in Telecommunications. John Wiley & Sons.

Dawkins, Richard (1994) *Das egoistische Gen.* Spektrum, Akad. Verlag.

Deco, Gustavo and Dragan Obradovic (1996) *An information-theoretic approach to neural computing.* Springer.

Drucker, H. (1999) Boosting Using Neural Networks. In Sharkey, Amanda J.C., editor, *Combining Artificial Neural Nets*, pages 51–78. Springer.

Efron, Bradley and Robert J. Tibshirani (1993) *An introduction to the bootstrap.* Chapman & Hall.

Engelmann, Thomas (1996) Multimodale Evolution Neuronaler Netze. Studienarbeit, Universität Karlsruhe.

Fletcher, Roger (1995) *Practical methods of optimization.* Wiley.

Freund, Yoav and Robert E. Schapire (1996) Experiments with a new boosting algorithm. In *Proceedings of 13th International Conference on Machine Learning*, pages 148–156.

Friedman, J. (1988) Multivariate adaptive regression splines. Technical report, Laboratory for Computational Statistics, Stanford University.

Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition.* Academic Press.

Geman, S., E. Bienenstock, and R. Doursat (1992) Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58.

Glover, Fred (1993) *Tabu Search.* Baltzer.

Goldberg, David (1989) *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley.

Gutjahr, S. (1996) Improving prediction systems by building independent committees. In Weigend, A. S., Y. Abu-Mostafa, and A. N. Refenes, editors, *Decision Technologies for Financial Engineering*, pages 104–110, Pasadena, Kalifornien, USA.

Gutjahr, S. (1998) Improving the determination of the hyperparameters in bayesian learning. In Downs, T., M. Frean, and M. Gallagher, editors, *Proceedings of the Ninth Australian Conference on Neural Networks (ACNN 98)*, pages 114–118, Brisbane, Australien.

Gutjahr, Steffen (1999) *Optimierung Neuronaler Netze mit der Bayes'schen Methode.* Dissertation, Universität Karlsruhe, Institut für Logik, Komplexität und Deduktionssysteme.

Hanke, Martin (1995) The minimal error conjugate gradient method is a regularization method. *Proceedings of the American Mathematical Society*, 123:3487–3498.

Hashem, S. (1999) Treating Harmful Collinearity in Neural Network Ensembles. In Sharkey, Amanda J.C., editor, *Combining Artificial Neural Nets*, pages 101–125. Springer.

Hassibi, Babak and David G. Stork (1992) Second order derivatives for network pruning: Optimal Brain Surgeon. In *NIPS 4.*

Hecht-Nielsen, Robert (1991) *Neurocomputing.* Addison-Wesley.

Henze, N. (1995) *Stochastik I.* Skriptum.

Henze, N. (1997) *Stochastik für Einsteiger.* Vieweg.

Hertz, John, Anders Krough, and Richard G. Palmer (1991) *Introduction to the theory of neural computation*, volume 1 of *Santa Fe Institute, Studies in the sciences of complexity, lecture notes.* Addison-Wesley.

Heskes, Tom (1998) Bias/variance decompositions for likelihood-based estimators. *Neural Computation*, 10:1425–1434.

Heuser, H. (1986) *Lehrbuch der Analysis, Teil 1.* Teubner.

Hofmann, Thomas and Joachim M. Buhmann (1997) Pairwise data clustering by deterministic Annealing. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 1–14.

Hofmann, Helmut (1997) Optimierung von MLP-Netzen mittels Reinforcement-Lernen und Evolution. Diplomarbeit, Universität Karlsruhe.

Jeffreys, H. (1961) *Theory of Probability.* Oxford University Press.

Jordan, Michael I. and Robert A. Jacobs (1994) Hierarchical Mixtures of Experts and the EM Algorithm. *Neural Computation*, 6:181–214.

Kaufmann, Heinz and Heinz Pape (1996) Clusteranalyse. In Fahrmeir, L., A. Hamerle, and G. Tutz, editors, *Multivariate Statistische Verfahren.* de Gruyter.

Krogh, Anders and Jesper Vedelsby (1995) Neural Network Ensembles, Cross Validation and Active Learning. In D.S. Touretzky, G. Tesauro, T.K. Leen, editor, *Advances in Neural Information Processing*, volume 7. MIT press.

LeCun, Y., J.S. Denker, and S.A. Solla (1990) Optimal Brain Damage. In *NIPS 2*.

Liu, Y. and Xin Yao (1998) Simultaneous learning of negatively correlated neural networks. pages 183–187.

MacKay, D. J. C. (1992) Bayesian interpolation. *Neural Computation*, 4(3):415–447.

MacKay, D. (1994) Bayesian methods for backpropagation networks. *Models of Neural Networks III,*.

Mangasarian, O. and W.Wolberg (1990) Cancer diagnosis via linear programming. *SIAM News*, (5).

Menzel, Wolfram (1998) Probmlem Solving with Neural Networks. In Ratsch, U., M.M. Richter, and I.-O. Stamatescu, editors, *Intelligence and Artificial Intelligence*. Springer.

Menzel, Wolfram (1999) Neuronale Netze zur Prognose von Finanzzeitreihen und Absatzzahlen. In Nol, G., G. Nakhaeizadeh, and K.-H. Vollmer, editors, *Datamining und Comutational Finance*, pages 95–114. Physica-Verlag.

Merz, C.J. and P.M. Murphy (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Møller, M. (1993) A Scaled Conjugate Gradient Algorithm for fast Supervised Learning. *Neural Networks*, 6:525–533.

Nauck, Detlef, Frank Klawonn, and Rudolf Kruse (1994) *Neuronale Netze und Fuzzy-Systeme*. Vieweg.

Nautze, C. and S. Gutjahr (1997) Extended bayesian learning. In *Proceedings of ESANN 97, European Symposium on Artificial neural networks, Bruges*, pages 321–326.

Nautze, Christian (1997) Bayes'sches Lernen bei Neuronalen Netzen. Diplomarbeit, Universität Karlsruhe.

Neal, Radford (1994) *Bayesian Learning for Neural Networks*. Doctoral dissertation, University of Toronto, Canada, Departement of Computer Science.

Neal, Radford (1998) Assessing relevance determination methods using DELVE. In Bishop, C. M., editor, *Generalization in Neural Networks and Machine Learning*. Springer.

Penny, W. D. and S. J. Roberts (1998) Bayesian neural networks for classification: how useful is the evidence framework? . Technical report, Department of Electrical Engineering.

Pereira, Fernando, Naftali Z. Tishby, and Lillian Lee (1993) Distributional clustering of English words. In *30th Annual Meeting of the Association for Computational Linguistics*, pages 183–190.

Perrone, M. and L. Cooper (1994) When networks disagree: ensemble methods for hybrid neural networks. *Artificial neural networks for speech and vision.*, pages 126–142.

Prechelt, Lutz (1994) Proben 1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules. Technical Report 21/94, Universität Karlsruhe, Fakultät für Informatik.

Preut, Karl-Heinz (1995) Strukturoptimierung von Neuro-Fuzzy Systemen. Diplomarbeit, Universität Karlsruhe.

Pütz, Udo (1997) Evolutionäre Optimierung neuronaler Netze für Reinforcement Probleme. Diplomarbeit, Universität Karlsruhe.

Puzicha, Jan, Thomas Hofmann, and Joachim M. Buhmann (2000) A Theory of Proximity Based Clustering: Structure Detection by Optimization. *Pattern Recogniton*, 33:617–634.

Ragg, Thomas and Steffen Gutjahr (1997a) Automatic Determination of Optimal Network Topologies based on Information Theory and Evolution. In *IEEE, Proceedings of the 23rd EUROMICRO Conference 1997*, pages 549–555.

Ragg, Thomas and Steffen Gutjahr (1997b) Building High Performant Classifiers by Integrating Bayesian Learning, Mutual Information and Committee Techniques – A Case Study in Time Series Prediction –. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Artificial Neural Networks 1997, Lausanne, Switzerland*. Springer.

Ragg, Thomas and Steffen Gutjahr (1998) Optimizing the Evidence – with an application to Time Series Prediction. In *Proceedings of the International Conference on Artificial Neural Networks 1998, Sweden*, Perspectives in Neural Computing, pages 275–280. Springer.

Ragg, Thomas, Heinrich Braun, and Johannes Feulner (1995) Improving temporal differnce learning for deterministic sequential decision problems. In *Proceedings of the ICANN '95, Paris*, pages 117–122. EC 2.

Ragg, Thomas, Heinrich Braun, and Heiko Landsberg (1997) A Comparative Study of Neural Network Optimization Techniques. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms 1997, Norwich, UK*, pages 343–347. Springer.

Ragg, Thomas, Wolfram Menzel, Walter Baum, and Michael Wigbers (2000) Predicting Sales Rates for Thousands of Retail Traders. In Tsaptsinos, D., editor, *Proceedings of the International Conference on Engineering Applications of Neural Networks, Kingston, England*, pages 199–206.

Ragg, Thomas (1996) Parallelization of an Evolutionary Neural Network Optimizer Based on PVM . In *Parallel Virtual Machine - EuroPVM'96*, Lecture Notes in Computer Science 1156, pages 351–354.

Ragg, Thomas (2000). Evolving Committes of Neural Networks.

Ramsay, J. O. and B.W. Silverman (1997) *Functional data analysis.* Springer.

Rätsch, G., T. Onoda, and K.R. Müller (1998) Soft margins for adaboost. Technical Report NC-TR-1998-021, GMD, Berlin.

Rechenberg, Ingo (1994) *Evolutionsstrategie '94.* Frommann-Holzboog Verlag, Stuttgart.

Reed, Russell (1993) Pruning Algorithms - A Survey. *IEEE Transactions on Neural Networks*, 4:740–747.

Reeves, Colin R., editor (1993) *Modern Heuristic Techniques for Combinatorial Problems.* Advanced topics in computer science. Orient Longman, Department of Statistics and Operational Research, school of mathematical and information sciences, Coventry University.

Riedmiller, M. (1994) Advanced supervised learning in multi-layer perceptrons - from backpropagation to adaptive learning algorithms. *Int. Journal of Computer Standards and Interfaces*, 16:265–278. Special Issue on Neural Networks.

Ripley., B. D. (1996) *Pattern recognition and neural networks.* Cambridge University Press.

Rosen, Bruce E (1996) Ensemble Learning Using Decorrelated Neural Networks. *Connection Science*, 8:373–384.

Rumelhart, D. E., G. E. Hinton, and R. Williams (1986a) Learning Internal Representations by Error Propagation. *Nature.*

Rumelhart, D.E., G.E. Hinton, and R.J. Williams (1986b) Learning internal represantations by error propagation. In Rumelhart, D.E. and J.L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructures of Cocnition*, volume 1, chapter 8, pages 318–362. MIT Press.

Schäfer, Johannes (1994) Evolution Neuronaler Netze zur Erkennung von handgeschriebenen Ziffern. Diplomarbeit, Univerität Karlsruhe, Institut für Logik, Komplexität und Deduktionssysteme.

Schapire, Robert E., Yoav Freund, Peter Bartlett, and Wee Sun Lee (1997) Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings of 14th International Conference on Machine Learning.*

Schiffmann, W., M. Joost, and R. Werner (1992) Synthesis and performance analysis of multilayer neural network architectures. Technical Report 16/1992, University of Koblenz, Insitute for Physics, Rheinau 3–4, D-5400 Koblenz.

Schlittgen, Rainer and Bernd H. J. Streitberg (1999) *Zeitreihenanalyse.* Oldenbourg.

Schmiedle, Frank (1997) Optimierung von Lernmengen für neuronale Netze mittels evolutionärer Algorithmen am Beispiel von HARMONET. Studienarbeit, Universität Karlsruhe.

Schubert, Matthias (1995) Evolutionäre Optimierung Neuronaler Netze zur Zeitreihenvorhersage. Diplomarbeit, Univerität Karlsruhe, Institut für Logik, Komplexität und Deduktionssysteme.

Schürmann, Jürgen (1996) *Pattern Classification.* John Wiley & Sons.

Schwefel, Hans-Paul (1995) *Evolution and Optimum Seeking.* Sixth-Generation Computer Technology Series. John Wiley & Sons.

Scott, D.W. and J.R. Thompson (1983) Probability density estimation in higher dimensions. In Gentle, J.E., editor, *Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface*, pages 173–179.

Shannon, C.E. (1948) A mathematical theory of communication. *Bell Sys. Tech. Journal*, 27:379–423.

Sharkey, Amanda J.C. (1999) Multi-Net Systems. In Sharkey, Amanda J.C., editor, *Combining Artificial Neural Nets*, pages 1–30. Springer.

Silverman, B.W. (1986) *Density Estimation for Statistics and Data Analysis.* Chapman and Hall.

Sprenger, Andreas (1996) Evolutive Optimierung von Neuro-Fuzzy Systemen mit radialen Basisfunktionen. Diplomarbeit, Universität Karlsruhe.

Stahlberger, A. and M. Riedmiller (1997) Fast network pruning and feature extraction by removing complete units. In Jordan, M., editor, *NIPS 9*, volume 9. MIT Press.

Stahlberger, Achim (1996) OBS - Ein Verfahren zum Ausdünnen neuronaler Netze. Verbesserungen und neue Ansätze. Diplomarbeit, Universität Karlsruhe, Institut für Logik, Komplexität und Deduktionssysteme.

Thodberg, Hans Henrik (1993) Ace of bayes: Applications of neural networks with pruning. Technical Report 1132E, Danish Meat Research Institute.

Vapnik, Vladimir (1982) *Estimation of Dependences Based on Empirical Data*. Springer.

Vapnik, Vladimir (1995) *The Nature of Statistical Learning Theory*. Springer.

Weisbrod, Joachim (1992) Einsatz Genetischer Algorithmen zur Optimierung der Topologie mehrschichtiger Feedforward-Netzwerke. Diplomarbeit, Universität Karlsruhe, Institut für Logik, Komplexität und Deduktionssysteme.

White, Halbert (1989) Learning in artificial neural networks: a statistical perspective. *Neural Computation*, 1:425–464.

Williams, C. K. I., C. Qazaz, C. M. Bishop, and H. Zhu (1995) On the relationsship between bayesian error bars and the input data density. In *Proceedings of the Fourth Int. Conference on Artificial Neural Networks*, pages 160–165.

Wilson, Edward O. (1995) *Der Wert der Vielfalt*. Piper.

Wolpert, D. H. (1992) Stacked generalization. *Neural networks*, 2(5):241–259.

Yao, Xin (1999) Evolving artificial neural networks. *Proceedings of the IEEE*, 87:1423–1447.

Zagorski, Peter (1994) Entwicklung Evolutionärer Algorithmen zur Optimierung der Topologie und des Generalisierungsverhaltens von Multilayer Perceptrons. Diplomarbeit, Univerität Karlsruhe, Institut für Logik, Komplexität und Deduktionssysteme.

Zell, Andreas (1994) *Simulation Neuronaler Netze*. Addison-Wesley.