

Wirtschaftswissenschaftliche Fakultät
Ruprecht-Karls-Universität Heidelberg



Einsatz Künstlicher Neuronaler Netze auf dem Aktienmarkt

Diplomarbeit
im Studiengang
Volkswirtschaftslehre

vorgelegt von

Stefan Selle
aus Wolfsburg

1998

Prediction is difficult,
especially for the future.

Niels Bohr
(dänischer Physiker und Nobelpreisträger)

Kurzfassung

Gegenstand dieser Arbeit ist die Untersuchung der Eignung Künstlicher Neuronaler Netze auf dem Aktienmarkt. Dazu werden zunächst allgemein Aufbau, Funktion und Arbeitsweise von KNN beschrieben und deren prinzipielle Einsatzmöglichkeiten aufgezeigt. Eine spezielle Anwendung KNN ist die Aktienkursprognose. Es werden einige klassische Analysemethoden angegeben und anschließend anhand von typischen Beispielen die Aktienanalyse mittels KNN vorgestellt. Abschließend werden die Fähigkeiten KNN in einer Simulation überprüft.

In dieser Arbeit soll der kurzfristige Verlauf des Deutschen Aktienindex mit Hilfe von KNN vorhergesagt werden. Ausgehend von verschiedenen Wirtschafts- und Börsendaten wird versucht, die Kursentwicklung des DAX des nächsten Tages zu prognostizieren und zwar im Zeitraum vom 1.10.1997 bis 28.2.1998 (70 Börsentage). Insbesondere soll der Kurseinbruch des DAX am 28.10.1997 infolge eines Crashes an der Börse in Hongkong erkannt werden.

Grundlage des Modell bildet die Intermarket-Analyse. Als Eingabedaten dienen, neben dem bisherigen Kursverlauf des DAX, die Historien anderer Aktienindizes, Wechselkurse, Renten, Rohstoffwerte, sowie Handelsvolumen und monetäre Größen im Zeitraum von 1.1.1990 bis 30.9.1997 (1702 Börsentage). Im Rahmen einer Korrelationsanalyse werden 7 der 41 Eingabegößen eliminiert. Aus den übrigen 34 Datenreihen werden 150 Zeitreihen transformiert, die als Eingaben in die Simulation einfließen.

Bei den Simulationen wird ein vollkommen vernetztes Mehrschichten-Perzeptron verwendet. Das Netz wird mit dem Backpropagation-Lernalgorithmus mit Momentum-Term trainiert. Als Optimierungswerkzeuge zur Komplexitätsreduktion kommen die Techniken Optimal Brain Damage kombiniert mit Input-Pruning und Hidden-Merging zum Einsatz. Die Versuche werden mit dem Stuttgarter Neuronale Netze Simulator durchgeführt.

Das beste Generalisierungsergebnis konnte mit einem optimierten Netz erzielt werden, das 20 der 150 Eingaben benutzt und das 4 Neuronen der inneren Schicht enthält. Insgesamt müssen 24 Gewichte angepaßt werden. Verglichen mit einer naiven Prognose ist die Generalisierungsleistung dieses Netzes allerdings schlecht. Dieses Resultat läßt sich auf starke Kursschwankungen während des Untersuchungszeitraums zurückführen. Zwischen Minimum und Maximum liegt ein Kurszuwachs von 32 %. Der Börsencrash konnte mit keinem Netz vorhergesagt werden.

Die Prognoseergebnisse der KNN werden als Grundlage von Anlageentscheidungen herangezogen. Dabei schlagen sowohl das optimierte Netz als auch der Vorgänger ohne Optimierung die einfache Buy-and-Hold-Strategie. Insbesondere zwischen dem 1.10.1997 und 13.1.1998 erzielen die KNN eine Rendite von 15,5 % bzw. 25,5 %, während die Buy-and-Hold-Strategie ein Minus von 0,2 % erwirtschaftet.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Wissenschaftliche Positionierung	1
1.2	Biologische Neuronale Netze	2
1.3	Geschichtliche Entwicklung	3
2	Grundlagen	6
2.1	Das Perzeptron	7
2.2	Approximation von Funktionen	8
2.3	Topologie	9
2.4	Lernen	9
2.5	Klassifikation	12
2.6	Entwicklung und Optimierung	21
3	Ökonomische Anwendungen	25
3.1	Prognose	25
3.2	Datenanalyse	26
3.3	Mustererkennung i.e.S.	27
3.4	Optimierung	28
4	Aktienanalysen	29
4.1	Die Random-Walk-Hypothese	29
4.2	Die fundamentale Aktienanalyse	30
4.3	Die technische Aktienanalyse	34
4.4	Die moderne technische Aktienanalyse	36
4.5	Intermarket Analyse	38
4.6	Nichtlineare dynamische Theorie	39
4.7	Aktienanalysen mit Hilfe von KNN	40
5	Prognose des Deutschen Aktienindex	46
5.1	Ziel der DAX-Prognose	46
5.2	Eingabedaten	47
5.3	Datenpräsentation	50
5.4	Durchführung der Simulationen	53

6	Ergebnisse und Diskussion	57
6.1	MLP ohne Optimierung	57
6.2	MLP mit Optimierung	59
6.3	Naive Prognose	60
6.4	Anlageentscheidungen	61
7	Zusammenfassung und Ausblick	63
8	Literaturverzeichnis	65
A	Graphische Darstellungen der Eingabedaten	78
A.1	Aktienindizes	78
A.2	Devisen	80
A.3	Renten	83
A.4	Rohstoffe	83
A.5	Monetäre Größen	84
A.6	Handelsvolumen	84
B	Suche nach historischen Zeitreihen	85
C	Korrelationsanalyse	86
C.1	Quellprogramm in FORTRAN 77	86
C.2	Korrelationskoeffizienten	91
D	Vorverarbeitung und Datenkonvertierung	99
D.1	Quellprogramm in FORTRAN 77	100
E	Simulationsergebnisse	111
E.1	Depotentwicklung	111
E.2	Topologie des optimierten MLP	112

Abbildungsverzeichnis

1.1	Darstellung zweier miteinander vernetzter Neuronen	2
2.1	Struktur eines künstlichen Neurons	6
2.2	Struktur eines Perzeptrons	7
2.3	Darstellung logischer Schaltungen mittels Perzeptronen	7
2.4	Taylorsche Reihe als Funktionennetz	8
2.5	Gradientenabstiegsverfahren im zweidimensionalen Fehleraum . .	11
2.6	Typische Probleme des Gradientenabstiegsverfahrens	11
2.7	Übertrainieren und Trainingsabbruch	12
2.8	Klassifikation von Netzwerkmodellen	12
2.9	Mehrschichten-Perzeptron	15
2.10	Darstellung der logistischen Funktion und ihrer Ableitung	15
2.11	Optimierungswerkzeuge bei dem Design KNN	22
3.1	Klassifikationsschema für wirtschaftliche Anwendungen KNN	25
5.1	Deutscher Aktienindex	46
5.2	Aktienindizes	49
5.3	Fehlerfunktion	55
6.1	DAX-Prognose mit dem (150-15-1)-MLP	57
6.2	DAX-Prognose mit dem (20-4-1)-MLP	59
6.3	Naive DAX-Prognose	61
A.1	Aktienindizes Nordamerika	78
A.2	Aktienindizes Asien/Pazifik	79
A.3	Aktienindizes Europas	80
A.4	Wechselkurse Nordamerika	80
A.5	Wechselkurse Asien/Pazifik	81
A.6	Wechselkurse Europa	82
A.7	Zinssätze USA und BRD	83
A.8	Rohstoffe	83
A.9	Geldmenge und Diskontsatz USA und BRD	84
A.10	Volumen	84
D.1	Aufbau der Eingabedatei	99

Abkürzungsverzeichnis

Allgemeines

Abb.	Abbildung
BRD	Bundesrepublik Deutschland
Kap.	Kapitel
MEZ	Mitteleuropäische Zeit
US	United States (of America)
bzw.	beziehungsweise
ca.	circa
d.h.	daß heißt
engl.	englisch
i.a.	im allgemeinen
i.d.R.	in der Regel
i.e.S.	im eigentlichen Sinne
i.w.S.	im weiteren Sinne
s.o.	siehe oben
s.u.	siehe unten
u.a.	unter anderem
usw.	und so weiter
u.U.	unter Umständen
vgl.	vergleiche
z.B.	zum Beispiel
z.T.	zum Teil

Ökonomie

AG	Aktiengesellschaft
BSP	Bruttosozialprodukt
DM	Deutsche Mark
ECU	European Currency Unit
EWS	Europäisches Währungssystem
GfK	Gesellschaft für Konsumforschung
HKMA	Hong Kong Monetary Authority

Informatik

AI	Artificial Intelligence
ART	Adaptive Resonance Theory

BAM	Bidirectional Associative Memory
BP	Backpropagation
BSB	Brain-State-in-a-Box
EA	Evolutionäre Algorithmen
GA	Genetische Algorithmen
KI	Künstliche Intelligenz
KNN	Künstliche Neuronale Netze
LVQ	Lineare Vektorquantisierung
MB	Mega Byte = $1024 \cdot 1024$ Byte
MLFF	Multilayer Feedforward Network
MLP	Multilayer Perceptron
NP	Nichtdeterministisch Polynomiale Komplexität
OCR	Optical Character Recognition
RAM	Random Access Memory
SOM	Self-organizing Map
TSP	Travelling Salesman Problem
XOR	Exklusive ODER-Funktion
XPS	Expertensysteme
kHz	Kilo Hertz = 1000 (Takte)/Sekunde
ms	Milli Sekunde = $(1/1000)$ Sekunde

Aktienanalysen

ADL	Advance-Dcline-Line
FA	Fundamentale Aktienanalyse
GD	Gleitender Durchschnitt
KGv	Kurs-Gewinn-Verhältnis
MSE	Mean Squared Error
RS	Relative Stärke
RSI	Relativer Stärke Index
SSE	Sum of Squared Errors
TA	Technische Aktienanalyse

Firmen, Produkte

SGZ-Bank	Südwestdeutsche Genossenschafts-Zentralbank
SENN	Software Entwicklungsumgebung für NN
SMPS	Stock Market Prediction System
SNI AG	Siemens Nixdorf Informationssysteme AG
SNNS	Stuttgarter Neuronale Netze Simulator

Börse

AFGX	Affärsvärlden General Index
AMEX	American Stock Exchange Index
CAC	Compagnie des Agents de Change
CBOE	Chicago Board Options Exchange
DAX	Deutscher Aktienindex
FTSE	Financial Times Stock Exchange
FWB	Frankfurter Wertpapierbörse
IBIS	Integriertes Börsenhandels- und Informationssystem
KOSPI	Korea Composite Stock Price Index
MIB	Milano Italia Borsa
NASDAQ	National Association of Securities Dealers Automated Quotation
NYSE	New York Stock Exchange
STI	Straits Times Index
S&P	Standard and Poor's Index
TOPIX	Tokyo Stock Exchange Price Index
TSE	Toronto Stock Exchange

Variablen

C	Komplexitätsterm (engl. complexity term)
E	Fehlerfunktion (engl. error)
F, f	Funktionen
P	Zahl der Muster (engl. pattern)
i, j, k	Laufindizes
t	Gewünschte Ausgabe (engl. teaching input)
v, w	Gewichte (engl. weights)
x, y, z	Ein- und Ausgaben
Θ	Schwellenwert
η	Lerngeschwindigkeitskoeffizient
α	Momentum-Term
ρ	Korrelationskoeffizient
E	Erwartungswert
V	Varianz
COV	Kovarianz
X, Y, Z	Zeitreihen
t	Zeitindex

1 Einleitung

In den letzten Jahren ist das Interesse an neuen und innovativen wirtschaftswissenschaftlichen Problemlösungsverfahren stark gestiegen. Bekannte Verfahren aus dem Operations Research und der Statistik erzielen angesichts hoher Komplexität und fehlender Transparenz der zugrundeliegenden Problemstellung u.U. nur unbefriedigende Lösungen. Als Alternativen dringen deshalb zunehmend Verfahren aus dem Forschungsgebiet der *Künstlichen Intelligenz* (KI, engl. artificial intelligence (AI)) in ökonomische Problemstellungen vor. Künstliche Neuronale Netzwerke (KNN) sind intelligente Problemlösungsverfahren, die sich besonders für nicht-konservative Aufgabenstellungen eignen, bei denen kein exaktes Modell der Ursache-Wirkungszusammenhänge vorliegt.

1.1 Wissenschaftliche Positionierung

Die *Kognitionswissenschaft* bezeichnet eine interdisziplinäre Forschungsrichtung (Neurowissenschaften, Psychologie, Philosophie, Sprachwissenschaften und Informatik), die durch die Intelligenzleistungen des menschlichen Gehirns inspiriert worden ist. Im Mittelpunkt der Forschung steht die Annahme, daß sich kognitive Prozesse¹ als Prozesse der Informationsverarbeitung betrachten lassen.

Aus biologisch motivierten Modellen und der informationstechnischen Umsetzung auf Computern entwickelte sich die Forschungsrichtung des *Konnektionismus*, der innerhalb der Informatik zum Teilgebiet der Künstlichen Intelligenz² gezählt wird. Bei der Erforschung kognitiver Prozesse sind konnektionistische Systeme ein wesentliches Modellierungsinstrument. Das Ziel konnektionistischer Modelle ist es, die Struktur- und Funktionsprinzipien des menschlichen Gehirns abzubilden und damit dessen technische Nachbildung in KNN zu ermöglichen.

Neben der Erforschung biologischer Vorgänge, beginnen konnektionistische Modelle als universelle Problemlösungsverfahren andere Aufgabenbereiche zu erobern. Probleme mit unbekannten Ursache-Wirkungszusammenhängen lassen sich nicht explizit modellieren, sondern u.a. im Rahmen der KI mit Hilfe von intelligenten Informationssystemen lösen. Nach Definition sind intelligente Informationssysteme durch fünf idealtypische Merkmale gekennzeichnet: Lernfähigkeit³, Anpassungsfähigkeit (Adaption), Flexibilität, Nachvollziehbarkeit und Entdeckungsfähigkeit [Goo95]. Daher kommen als Problemlösungsverfahren

¹Unter kognitiven Prozessen versteht man Leistungen des menschlichen Gehirns ausgehend von Sinneswahrnehmungen.

²Die Modelle des Konnektionismus entsprechen allerdings nicht dem in der klassischen KI manifestierten Paradigma der Symbolverarbeitung.

³Zwischen der Lernfähigkeit von Mensch und Maschine muß unterschieden werden [Pen91]. Maschinelles Lernen wird auch als zielgerichtete Selbstmodifikation bezeichnet.

neben KNN auch Expertensysteme (XPS), Fuzzy Logic, Evolutionäre Algorithmen (EA) und deren Kombinationen in Betracht.

1.2 Biologische Neuronale Netze

Biologische Nervensysteme arbeiten massiv parallel, sind weitgehend fehlertolerant, verhalten sich adaptiv und als lernende Systeme, die ihre eigenen Parameter bestimmen und anpassen können. Das Wesen der Funktion des Nervensystems besteht in der Kontrolle durch Kommunikation. Das menschliche Gehirn besteht aus etwa 10^{10} bis 10^{12} miteinander vernetzten Nervenzellen, den Neuronen. Etwa 10 % der Neuronen dienen der Eingabe (affarent) und Ausgabe (efferent). Die restlichen 90 % sind mit anderen Neuronen verknüpft, die Informationen speichern oder bestimmte Umwandlungen des Signals vornehmen, das sich durch das Netzwerk fortpflanzt.

Neuronen sind komplexe Zellen, die auf elektrochemische Signale reagieren. Sie setzen sich zusammen aus einem Zellkern, einem Zellkörper, mehreren Dendriten, die über Synapsen Eingabeverknüpfungen zu anderen Neuronen herstellen (siehe Abb. 1.1), sowie einem Axonstrang, der über Endkolben oder Synapsen ein Aktionspotential ausgibt. Ein Neuron kann mit hunderten bis tausenden anderen Neuronen verbunden sein. Die Verbindungen erfolgen über zwei allgemeine Synapsentypen: exzitatorische (erregende) und inhibitorische (hemmende).

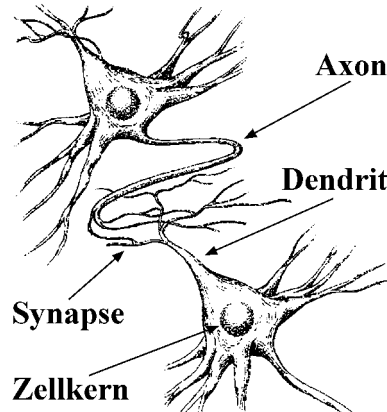


Abbildung 1.1: Darstellung zweier miteinander vernetzter Neuronen [NKK96].

Die neuronale Aktivität wird bestimmt durch die Entstehung eines internen elektrischen Potentials, dem Membranpotential. Dieses Potential kann durch die Eingabeaktivitäten seitens anderer Zellen über die Synapsen verstärkt oder abgeschwächt werden. Wenn die kumulativen Eingaben das Potential über einen Schwellenwert heben, sendet das Neuron Impulse aus, indem es eine Folge von Aktionspotentialen über das Axon ausschüttet. Diese Impulse bewirken, daß eine

chemische Substanz, der Neurotransmitter, an die Synapsen ausgegeben wird, die wiederum andere Neuronen erregen oder hemmen können.

Das Axonsignal ist wegen des Schwellenwerts von Natur aus binär. Die nicht-binären Informationen, die im Nervensystem verarbeitet werden, sind nicht durch die Größe der Spannungen, sondern durch die zeitlichen Abstände des Aktionspotential codiert. Das Nervensystem arbeitet demnach mit Frequenzmodulation.

Die Zeit, die ein Reiz zum Durchqueren einer Synapse benötigt, beträgt etwa 1 ms. Nach dem Feuern entsteht eine unempfindliche Phase, die etwa 10 ms dauert und während derer das Neuron nicht feuern kann. Pro Sekunde können fünfzig bis mehrere hundert Ausschüttungen auftreten. Die Taktfrequenz des biologisch neuronalen Netzes liegt damit maximal im unteren kHz-Bereich und ist um mehrere Dimensionen kleiner als die Geschwindigkeit der Prozessoren eines konventionellen Computersystems (von-Neumann-Architektur). Die Leistungen des menschlichen Gehirns beruhen daher in erster Linie auf der hohen Parallelität bei der Informationsverarbeitung.

Synapsen können wachsen, verkümmern oder ganz verschwinden. Umgekehrt kann ein Axon neue Zweige mit den zugehörigen Synapsen ausbilden und dadurch mit weiteren Nervenzellen in Kontakt treten. Diese Wachstumsprozesse sind für Gedächtnis und Lernen verantwortlich.

1.3 Geschichtliche Entwicklung

Die Entwicklung der KNN begann in den 40er Jahren. Der Neurophysiologe Warren S. McCulloch und der Mathematiker Walter Pitts abstrahierten von den biologischen Vorgängen und schufen 1943 ein Modell des essentiellsten Gehirnbausteins auf Basis logischen Kalküls, das künstliche Neuron [MP43]. Dieses McCulloch-Pitts-Neuron oder Schwellenwert-Neuron genügt fünf Annahmen:

- Ein Neuron ist ein binäres Schaltelement, es ist entweder aktiv oder inaktiv.
- Jedes Neuron besitzt einen festen Schwellenwert.
- Ein Neuron empfängt Eingaben von exzitatorischen Synapsen gleichen Gewichts.
- Ein Neuron empfängt außerdem Eingaben von inhibitorischen Synapsen, deren Effekt absolut ist: eine aktive inhibitorische Synapse verhindert die Aktivierung des Neurons.
- Es gibt ein Zeitquantum für die Integration der synaptischen Eingaben. Wenn keine inhibitorische Synapse aktiv ist, werden die exzitatorischen Eingaben addiert und das Neuron wird aktiv, wenn sein Schwellenwert dadurch überschritten wird.

1949 formulierte Donald O. Hebb, ebenfalls Neurophysiologe, ein Modell des menschlichen Lernens, das die Lernvorgänge mit der Anpassung von Neuronen-

verbindungen bei gleichzeitiger Aktivität miteinander verbundener Neuronen begründet [Heb49]. Das Modell wurde als Hebbsche Lernregel bekannt und ist heute noch in Verwendung.

Ein Jahr später erschien Karl Lashleys Forschungsbericht über die Lokalität des Gedächtnisses im Gehirn [Las50]. Lashley ging davon aus, daß bestimmte Gehirnregionen für Erinnerungen zuständig sind. Im Laufe der Zeit wurde dies widerlegt, denn es stellte sich heraus, daß zwar bestimmte Lokalitäten im Gehirn spezielle Aufgaben übernehmen, aber Erinnerungen dezentral gespeichert werden. Aus diesen Überlegungen entstand die Idee der verteilten Informationsspeicherung [SHG90].

Die Arbeiten in den 40er Jahren interessierten Forscher verschiedener Gebiete, darunter den Entdecker des modernen Computermodells, John von Neumann, und den Mitbegründer der Kybernetik, Norbert Wiener. Aufbauend auf der Hebbschen Lernregel kombinierte der Psychologe Frank Rosenblatt das McCulloch-Pitts-Neuron in einem Netzwerk und beschrieb 1958 das Perzeptron [Ros58]. Alle grundlegenden Merkmale heutiger KNN sind enthalten: Lernfähigkeit, Selbstorganisation, Generalisierungsfähigkeit und Fehlertoleranz. Durch die richtige Wahl des Schwellenwerts konnten diese Konstrukte logische Operationen realisieren (vgl. Kap. 2.1).

Das erste praktische einsetzbare KNN konstruierten 1960 Bernhard Widrow und Marcian E. Hoff [WH60]. Ihr Adaline (engl. adaptive linear neuron) konnte einfache Klassifizierungsaufgaben ausführen und kam zum Beispiel für die Dämpfung des Echos in Telefonleitungen zur Anwendung. Wichtigster Schritt für die Entwicklung der KNN war jedoch eine neue Lernregel, die gegenüber dem Perzeptron Vorteile aufwies. Adaline benutzt den kleinsten quadratischen Fehler zwischen gewünschtem und erzeugtem Output als Fehlermaß. Die Widrow-Hoff-Regel ist auch als Delta-Regel bekannt.

1969 erlitt die Erforschung Künstlicher Neuronaler Netze einen Einbruch. Marvin Minsky und Seymour Papert, die zuvor den Begriff Künstliche Intelligenz geprägt haben, veröffentlichten eine vernichtende Kritik an den existierenden Neuronen-Modellen [MP69] mit dem Erfolg, daß zwischen 1970 und 1982 nur noch recht wenige Forscher voneinander getrennt in einzelnen Disziplinen weiterarbeiteten. Dieses Buch war eine elegante mathematische Analyse der Perzeptronen mit ihren Vor- und Nachteilen. Im wesentlichen wurde dabei gezeigt, welche logischen Funktionen einfache Perzeptronen verarbeiten konnten, welche nicht. Ein Perzeptron ist grundsätzlich nicht in der Lage, die zur Klasse der nichtlinearen separablen Funktionen gehörende exklusive Oder-Funktion (XOR) zu realisieren.

Dennoch entstanden in den 70er Jahren richtungsweisende Arbeiten, die aber aufgrund der Kritik Minskys und Paperts zunächst unbeachtet blieben. Der Elektronik-Ingenieur Teuvo Kohonen entwickelte 1972 ein Neuronen-Modell für Steuerungsaufgaben, den Korrelations-Matrix-Speicher [Koh72]. Der Neurophy-

siologe James A. Anderson entwickelte zeitgleich den linearen Assoziator und das Brain-State-in-a-Box Netzwerk [And70, And72] in Anlehnung an natürliche biologische und psychologische Anpassungsvorgänge. Paul J. Werbos legte 1974 in seiner verkannten Dissertation den Grundstein für den heute bekanntesten Lernalgorithmus (Backpropagation) [Wer74]. Stephen Grossberg entwarf die mathematisch fundierte Adaptive Resonanztheorie (ART), die stark am biologischen Vorbild orientiert ist und als weit entwickeltes KNN gilt [Gro76].

Die Renaissance der KNN begann zwischen 1979 und 1982. Wissenschaftler verschiedener Disziplinen (Biologen, Neurologen, Physiker, Mathematiker) entdeckten ein gemeinsames Interesse an der Forschung nach neuen Neuronen-Modellen. Allerdings entstand erst 1982 ein merklicher Schub für die neuronale Forschung als der renommierte Physiker und Nobelpreisträger John J. Hopfield Neuronen-Modelle für die Erklärung physikalischer Phänomene benutzte [Hop82]. Das Hopfield-Modell führte zur Entwicklung weiterer Netzwerkmodelle, die sich an den physikalischen Energiegesetzen orientierten. Ebenfalls im Jahr 1982 stellte Teuvo Kohonen seinen Ansatz über die selbstorganisierende Bildung topologisch korrekter Merkmalskarten vor [Koh82]. Die Idee Kohonens besteht darin, eine Schicht von Neuronen zu bilden, die auf eine Eingabe mit der Aktivierung einer bestimmten Region reagieren. Ähnliche Eingaben sollen benachbarte Regionen erregen.

Die 1985 von David H. Ackley, Geoffrey E. Hinton und Terence J. Sejnowski vorgestellte Boltzmann-Maschine verwendet Hopfield-Neuronen und hat ein physikalisches Analogon: dem langsamen Abkühlen einer Kristallschmelze [AHS85]. Die Boltzmann-Maschine war das erste KNN, das in der Lage ist, Neuronen innerer Schichten zu trainieren.

Der entscheidende Durchbruch KNN kam 1986 als zeitgleich und unabhängig voneinander David E. Rumelhart, Geoffrey E. Hinton und Ronald J. Williams [RHW86a, RHW86b] sowie David B. Parker [Par85] und Yann Le Cun [Le 86] eine neue Lernregel⁴, die generalisierte Delta-Regel, vorstellten, die die Kritik am Perzeptron aufhob. Diese heute als Backpropagation bekannte Lernregel ist in der Lage, die Verbindungsgewichte zu inneren Einheiten in mehrschichtigen KNN zu bestimmen. Die Idee besteht darin, den Fehler, den das Netz bei der Erzeugung einer Ausgabe macht, rückwärts durch das Netz, also von der Ausgabe zur Eingabeschicht, weiterzureichen (propagieren) und zur Gewichtsveränderung zu verwenden (vgl. Kap. 2.5).

KNN befinden sich zur Zeit in einem fortgeschrittenen Stadium der experimentellen Forschung. Die rasante Entwicklung der Computer in den vergangenen Jahren, insbesondere der technische Fortschritt und die damit verbundene Kostenreduzierung bei der Realisierung von Parallelrechnern, ermöglicht eine effiziente Umsetzung einer Vielzahl neuronaler Ansätze [Füs96].

⁴Die Lernregel wurde bereits von Paul J. Werbos entwickelt (s.o.).

2 Grundlagen

Die minimale Struktur eines informationsverarbeitendes Elements des KNN, das künstliche Neuron, besteht aus mehreren gerichteten Eingabeleitungen (vgl. Kap. 1.2: Dendriten), die mit Gewichten (Synapsen) versehen sind, einem Berechnungskörper (Zellkörper) und einer Ausgabeleitung (Axon). Das künstliche Neuron arbeitet wie folgt (siehe Abb. 2.1). Die n Eingabesignale (externer Input oder Ausgabesignale vorgelagerter Neuronen) x_i werden mit Gewichtungsfaktoren w_i multipliziert und mittels einer Propagierungsfunktion $p(x_1, \dots, x_n)$ zu einem Gesamt- oder Netto-Eingabesignal zusammengefaßt. Eine Aktivierungsfunktion bzw. Transferfunktion $f(p(x_1, \dots, x_n))$ sorgt dafür, daß unter Berücksichtigung des Netto-Eingabesignals ein Ausgabesignal (externer Output oder Eingabesignal für nachgelagerte Neuronen) y generiert wird.

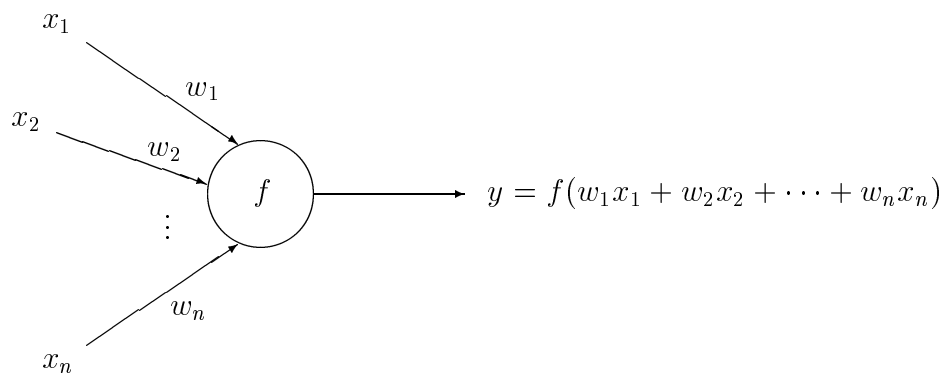


Abbildung 2.1: Struktur eines künstlichen Neurons

Die Vernetzung dieser Neuronen bildet das KNN. Dabei faßt man Neuronen zu Schichten zusammen. Es gibt eine Eingabe- und eine Ausgabeschicht, sowie innere, verborgene (engl. hidden) Schichten, wobei letztere keine direkte Verbindung mit der Außenwelt haben. Je nach Netztyp (vgl. Kap. 2.3) können die verborgenen Schichten ganz wegfallen und die Eingabeschicht kann mit der Ausgabeschicht identisch sein.

Als Propagierungsfunktion wird i.d.R. für alle Neuronen des KNN dieselbe Funktion benutzt, und zwar die gewichtete Summe aller Eingaben

$$p(x_1, \dots, x_n) = \sum_{i=1}^n w_i x_i \quad . \quad (2.1)$$

Ebenso wird nur eine Aktivierungsfunktion verwendet, zumindest für die Neuronen in einer Schicht. Die Wahl der Aktivierungsfunktion ist allerdings sehr stark modellabhängig. Mehrschichtige Modelle benutzen sehr oft eine sigmoide Aktivierungsfunktion (vgl. Kap. 2.5) und weisen daher kontinuierliche Aktivierungszustände auf, die meist aus $[0, 1]$ gewählt werden.

2.1 Das Perzeptron

Das Perzeptron (engl. perception = Wahrnehmung) ist ein einfaches KNN, das lediglich aus einer einzigen Verarbeitungseinheit besteht (siehe Abb. 2.2). Es besteht aus n Eingabeleitungen x_i , die jeweils mit den Gewichten w_i behaftet sind. Das Perzeptron kann nur Schwellenwertentscheidungen treffen, d.h. die Ausgabe der Zelle ist 1, falls $w_1x_1 + w_2x_2 + \dots + w_nx_n \geq \Theta$ gilt, wobei Θ der sogenannte Schwellenwert der Zelle ist. Falls $w_1x_1 + w_2x_2 + \dots + w_nx_n < \Theta$ gilt, wird eine Null ausgegeben.

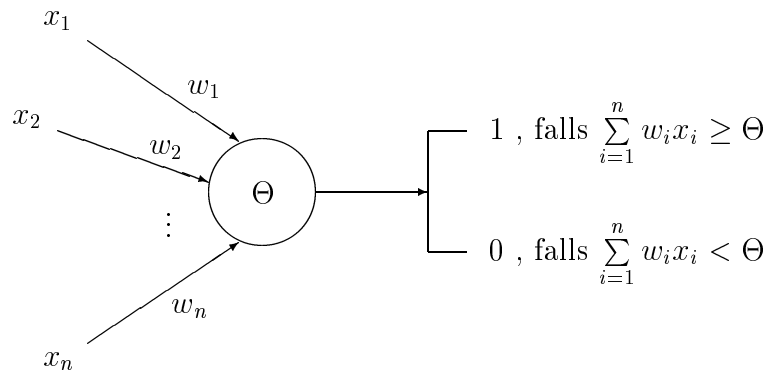


Abbildung 2.2: Struktur eines Perzeptrons

Mit Perzeptronen können logische Funktionen ohne weiteres realisiert werden (siehe Abb. 2.3). Die AND-Verknüpfung zweier binärer Variablen x_1 und x_2 kann durch ein Perzeptron mit zwei Eingabeleitungen implementiert werden, bei der w_1 und w_2 beide gleich 1 sind und $\Theta = 2$ gilt. Das Perzeptron wird nur dann eine 1 feuern, wenn $x_1 + x_2 \geq 2$ erfüllt ist, d.h. bei binären Eingaben genau dann, wenn $x_1 = x_2 = 1$. Bei der OR-Verknüpfung wird entsprechend $\Theta = 1$ gesetzt. Zur Darstellung der NOT-Verknüpfung genügt eine Eingabeleitung x_1 , die mit dem Gewicht $w_1 = -1$ versehen wird. Der Schwellenwert ist hier null.

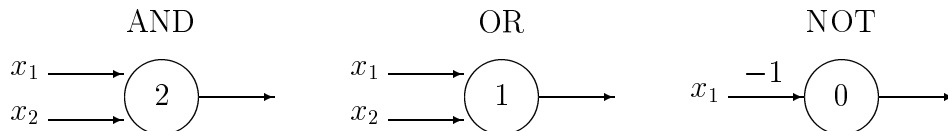


Abbildung 2.3: Darstellung logischer Schaltungen mittels Perzeptronen

Da die drei Funktionen AND, OR und NOT Basis für die Implementierung aller möglichen Booleschen Funktionen bilden, ist klar, daß mit einem Netz von Perzeptronen beliebige endliche logische Funktionen berechnet werden können.

2.2 Approximation von Funktionen

Ein altes Problem der Approximationstheorie besteht darin, eine vorgegebene Funktion durch die Komposition primitiver Funktionen exakt oder annähernd auszudrücken. Ein klassisches Beispiel ist die Approximation von eindimensionalen Funktionen mittels Polynomen. Die Taylorsche Reihe für eine Funktion $F(x)$, die am Punkt x_0 approximiert wird, hat die Form

$$F(x) = \sum_{i=1}^n w_i (x - x_0)^i, \quad (2.2)$$

wobei die Konstanten w_i vom Wert der Funktion F und ihrer Ableitung am Punkt x_0 abhängen. Für $n \rightarrow \infty$ wird der Funktionswert exakt durch die Taylorsche Reihe ausgedrückt. Abb. 2.4 zeigt, wie die polynomielle Approximation einer Funktion $F(x)$ als Funktionennetz dargestellt werden kann. An den Knoten des Netzes werden die Funktionen $z \rightarrow z^0, z \rightarrow z^1, \dots, z \rightarrow z^n$, also die Summanden der Taylorschen Reihe, berechnet. Der Ausgabeknoten sammelt additiv die ankommenden Informationen und gibt den Wert des ausgewerteten Polynoms aus. Die Netzgewichte w_i können mit Hilfe eines Lernalgorithmus bestimmt werden.

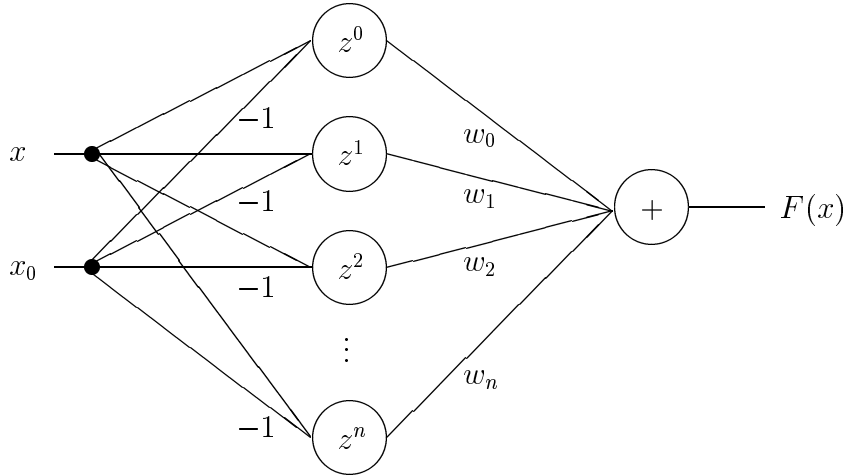


Abbildung 2.4: Taylorsche Reihe als Funktionennetz

Nach dem Theorem von Andrej N. Kolmogorov [Kol57, HN87b] ist es möglich, mehrdimensionale stetige Funktionen als ein endliches Netz primitiver eindimensionaler Funktionen darzustellen. Die Bestimmung der Netzparameter in einem KNN ist i.a. ein NP-vollständiges Problem⁵. Mit Hilfe des Erfüllbarkeitsproblems kann jedoch gezeigt werden, daß sich ein NP-vollständiges Problem in polynomieller Zeit auf ein Lernproblem für KNN reduzieren läßt [GJ79].

⁵NP steht für nichtdeterministisch polynomielle Komplexität. Das bedeutet, daß kein Algorithmus bekannt ist, der alle Instanzen des Problems in polynomieller Komplexität lösen kann.

2.3 Topologie

Künstliche Neuronale Netze lassen sich hinsichtlich ihrer Topologie klassifizieren:

- *Vorwärtsgekoppelte Netze* (engl. feedforward networks) sind Netze, in denen die Verbindungen nur in eine Richtung gehen, von der Eingabe zur Ausgabe.
- *Zyklische Netze* sind Netze, in denen sich einige Neuronen gegenseitig reizen.
- *Vollkommen verbundene Netze* sind Netze, in denen jedes Neuron mit allen anderen Neuronen verbunden ist (meistens bis auf sich selbst).
- *Geschichtete Netze* sind Netze, in denen die Neuronen in getrennten, nicht verbundenen Mengen angeordnet sind.
- *Symmetrische Netze* sind Netze, in denen die Verbindung zwischen zwei beliebigen Neuronen in beiden Richtungen gleich ist.
- *Selbstassoziative Netze* sind Netze, in denen Eingabe- und Ausgabe-Neuronen übereinstimmen.
- *Stochastische Netze* sind Netze, in denen eine gewisse Wahrscheinlichkeit besteht, daß ein Neuron nicht aktiviert wird, obwohl es Reize bekommen hat.
- *Asynchrone Netze* sind Netze, in denen die Neuronen nicht alle auf einmal (synchron), sondern zufällig eins nach dem anderen aktiviert werden.

2.4 Lernen

Ein wesentliches Merkmal KNN ist die allgemeine Lernfähigkeit, d.h. KNN lernen aus Erfahrungen, die sie durch präsentierte Trainingsdaten gewinnen. Lernen heißt Selbstanpassung der Gewichtungsfaktoren zwischen den Neuronen des KNN, so daß das Netz das gewünschte Verhalten zeigt. Lernmethoden für KNN können in drei grundsätzliche Kategorien eingeteilt werden: überwacht, bestärkt oder nicht-überwacht.

Beim *überwachten Lernen* (engl. supervised learning) ist während des Lernprozesses ein Lehrer anwesend, und jedes Beispielmuster für das Training des Netzwerks beinhaltet ein Eingabemuster sowie ein Ziel oder ein gewünschtes Ausgabemuster, die korrekte Antwort. Während des Lernprozesses kann ein Vergleich zwischen der vom Netzwerk berechneten und der korrekten Ausgabe angestellt werden, um den Fehler festzustellen. Der Fehler kann dann verwendet werden, um die Netzparameter (Gewichtungsfaktoren) entsprechend anzupassen, so daß der Netzwerk-Fehler reduziert wird. Nachdem die Gewichtungen für alle Trainingsmuster iterativ angepaßt wurden, konvergieren die Gewichtungswerte gegen eine Wertemenge, mit der die erforderlichen Mustervorgaben erzielt werden können. Das Lernen wurde dann erreicht, wenn die Fehler für alle Trainingsmuster auf einen akzeptablen Wert für neue Muster, die sich nicht in der Trainingsmenge befinden, minimiert wurde. Überwachtes Lernen wird auch induktives oder assoziatives Lernen genannt.

Beim *bestärkenden Lernen* (engl. reinforcement learning) wird ebenfalls die Anwesenheit eines Lehrers vorausgesetzt, aber dem Netzwerk wird die korrekte Antwort nicht präsentiert. Statt dessen erhält es nur einen Hinweis darauf, ob die berechnete Antwort richtig oder falsch ist. Anhand dieser Information muß das Netzwerk seine Performance verbessern. Normalerweise erhält es eine Belohnung, indem die Gewichtungen für Einheiten, die die richtige Antwort erzeugt haben, erhöht werden, während für die Einheiten mit der falschen Antwort die Gewichtungswerte reduziert werden.

Beim *nicht-überwachten Lernen* (engl. unsupervised learning) bzw. entdeckendem Lernen erhält das Netzwerk kein Feedback über die gewünschte oder die korrekte Ausgabe. Es gibt keinen Lehrer, der Zielmuster präsentiert. Deshalb muß das System durch Entdecken und die Übernahme strukturierter Eigenschaften der Eingabemuster lernen, d.h. durch die Anpassung an statistische Gleichmäßigkeiten oder Muster-Cluster aus den Trainings-Eingabebeispielen. Dieses Lernen kann durch die Verstärkung ausgewählter Gewichtungen bewerkstelligt werden, so daß eine Übereinstimmung mit zentralen, prototypischen Trainingsmustern erzielt wird, welche eine repräsentative Gruppe ähnlicher Muster oder Cluster darstellen.

Das *Hebbsche Lernen* stellt eine Form der korrelativen Gleichgewichtsangpassung dar, bezogen auf die prä- und postsynaptischen Kräfte eines Neurons. Die dafür grundlegende Theorie wurde von Donald Hebb vorgestellt [Heb49]. Wenn ein Axon der Zelle A nah genug an einer Zelle B liegt, um diese zu erregen, und wiederholt oder andauernd feuert, erfolgt in einer oder beiden Zellen ein Wachstumsprozeß oder eine metabolische Veränderung, so daß sich A's Einflußeffizienz auf B erhöht. Für diese Hebbsche Lernregel wurden zahlreiche Varianten vorgeschlagen, unter anderem eine Gewichtsangpassung basierend auf der Minimierung einer Energie- oder Entropiefunktion.

Beim *konkurrierenden Lernen* werden die Gewichtungen so angepaßt, daß Neuronen, die am stärksten auf einen Eingabereiz reagieren, bevorzugt werden. Die Gewichtsangpassung ist normalerweise eine modifizierte Form der Hebbschen Anpassung. Die Neuronen in dieser sogenannten Wettbewerbsschicht konkurrieren. Das Neuron, das die größte Übereinstimmung seiner Gewichtsangpassung mit dem Eingabemuster feststellt, gewinnt (Siegerneuron, engl. winner-takes-all), alle anderen verlieren.

Mehrere Lernparadigmen basieren auf der Reduzierung des Netzwerk-Fehlers mit Hilfe der *Methoden des steilsten Gradienten*. Diese Methoden machen es erforderlich, daß die Aktivierungsfunktion differenzierbar ist. Ausgehend vom aktuellen Fehler wird die Richtung ermittelt, in der sich der Fehler am schnellsten verringert. Damit entspricht der Lernalgorithmus dem Suchen des globalen Minimums in einem nichtlinearen Gleichungssystem. Abb. 2.5 zeigt das Prinzip des steilsten Gradientenabstiegs im zweidimensionalen Fehlerraum.

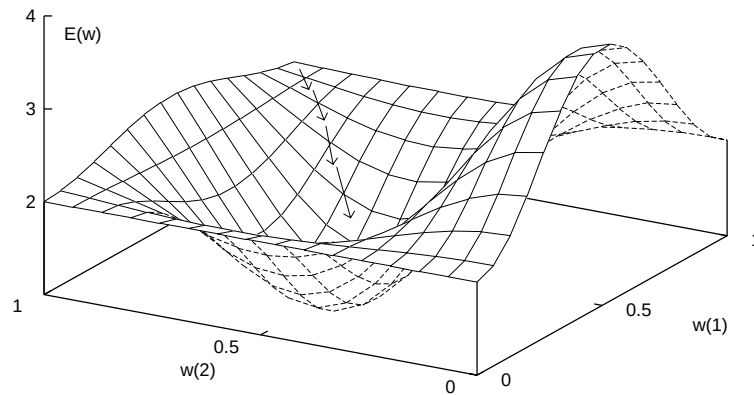


Abbildung 2.5: Gradientenabstiegsverfahren im zweidimensionalen Fehlerraum

Die Probleme, die bei dieser Lernmethode auftreten können, sind Oszillationen der Fehlerfunktion, eine hohe Anzahl von Iterationsschritten des Algorithmus aufgrund von Plateaus innerhalb der Fehlerfunktion, Abbruch des Algorithmus in einem lokalen Minimum der Fehlerfunktion und das Verlassen des globalen Minimums der Fehlerfunktion (siehe Abb.2.6).

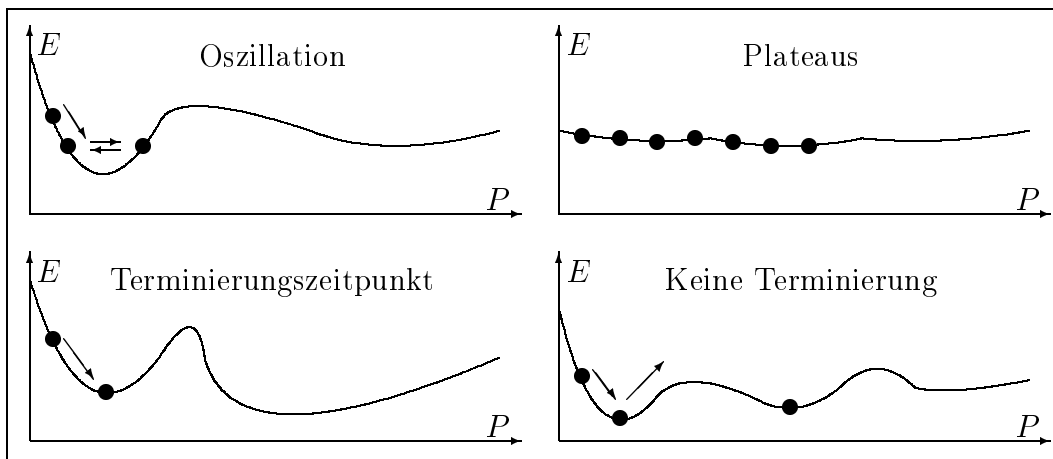


Abbildung 2.6: Typische Probleme des Gradientenabstiegsverfahrens [Füs96]

Präsentiert man dem Netzwerk zu häufig die gleichen Trainingsmuster, dann kann es passieren, daß das Netzwerk diese Muster auswendig lernt. Muster, die nicht zu dieser Trainingsmenge gehören, werden dann schlechter verarbeitet. Um dieses *Überlernen* (engl. overlearning) zu verhindern, wird während des Lernens der Netzfehler auf einer Trainings- und einer Validierungsmenge⁶ gemessen. Wenn der Netzfehler der Validierungsmenge größer wird, ist das Training abubrechen.

⁶Die Validierungsmenge ist nicht mit der Testmenge zu verwechseln, die dem KNN als dritte disjunkte Menge erst nach Ende des Trainings präsentiert wird.

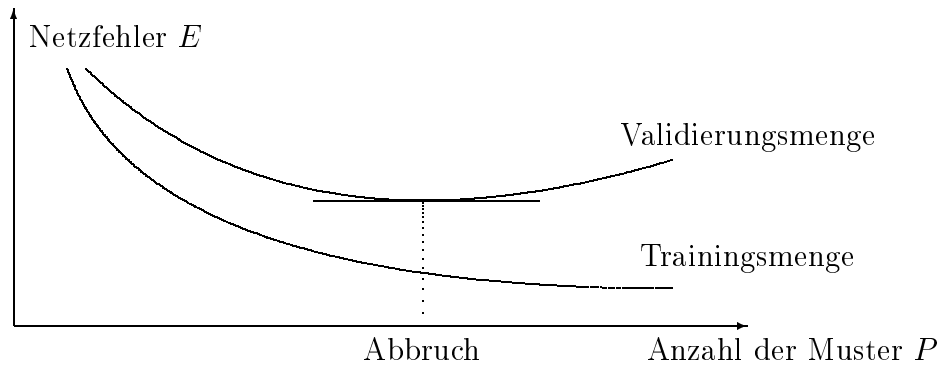


Abbildung 2.7: Übertrainieren und Trainingsabbruch

2.5 Klassifikation

Ein vollständiger Überblick zu allen existierenden Modellen KNN ist angesichts der unüberschaubaren Anzahl bekannter, modifizierter und vollkommen neuer Typen kaum mehr möglich. Eine übersichtsartige Klassifikation nach den wesentlichen Eigenschaften häufig in wirtschaftlichen Problemstellungen eingesetzter Netzwerkmodelle zeigt Abb. 2.8.

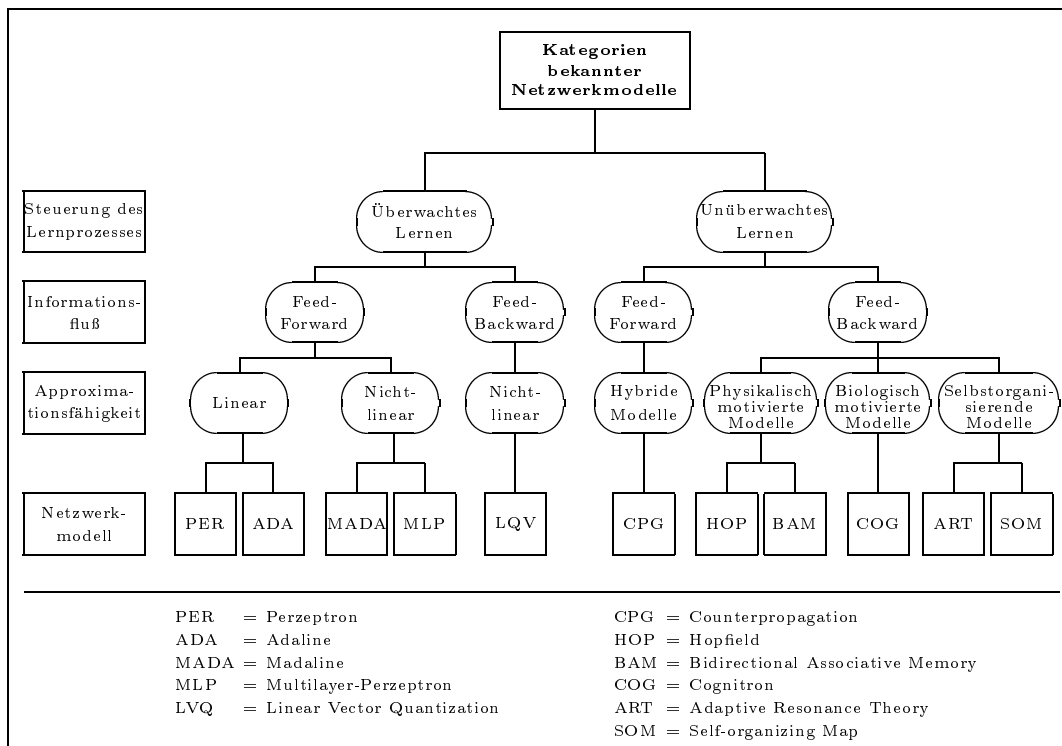


Abbildung 2.8: Merkmalsorientierte Klassifikation von Netzwerkmodellen [SS97]

Perzeptron Das Perzeptron wurde bereits in Kap. 2.1 behandelt. Die Lernregel eines Perzeptrons läßt sich folgendermaßen beschreiben:

- Wenn die Ausgabe Eins (aktiv) ist und Eins sein soll oder wenn sie Null (inaktiv) ist und Null sein soll, dann werden die Gewichtungen nicht verändert.
- Wenn die Ausgabe Null ist, aber Eins sein sollte, werden die Gewichtungen für alle aktiven Eingabeverknüpfungen erhöht.
- Wenn die Ausgabe Eins ist, aber Null sein sollte, werden die Gewichtungen für alle aktiven Eingabeverknüpfungen verringert.

Mathematisch läßt sich der Perzeptron-Lernalgorithmus als Änderung der Gewichtungen

$$\Delta w_i = \eta(t - y)x_i \quad (2.3)$$

schreiben, wobei x_i die i -te Eingabe, y die tatsächlich realisierte Ausgabe und t die gewünschte bzw. erwartete Ausgabe (engl. teaching input) sind. Der Parameter η ist der Lerngeschwindigkeitskoeffizient. Wenn η sehr klein ist, erfolgt das Lernen zwar sehr langsam, aber stabil. Der Vorteil des Perzeptron-Lernalgorithmus besteht darin, daß er bei gegebener linearer Separabilität⁷ der Lernaufgabe garantiert zu einer geeigneten Gleichgewichtskonfiguration hin konvergiert. Dieses sogenannte Perzeptron-Konvergenztheorem läßt sich mit Hilfe der Cauchy-Schwarzschen Ungleichung beweisen [Pat96].

Adaline Das Adaline (engl. adaptive linear neuron) [WH60] ist ein einzelnes Neuron mit Schwellenwertlogik und einer bipolaren Ausgabe mit den Werten $\{+1, -1\}$. Eingaben für diese Einheit erfolgen normalerweise auch bipolar, müssen aber nicht. Die Gewichtungen werden nach der Lernregel von Widrow-Hoff, die auf der Minimierung des Netzwerkfehlers beruht, angepaßt. Der Gesamtfehler des Netzwerks ist gerade die Summe der Einzelfehler

$$E_{\text{tot}} = \sum_{p=1}^P E^p \quad , \quad (2.4)$$

wobei P die Anzahl der Trainingsmuster (engl. pattern) darstellt. Der Einzelfehler ergibt sich aus dem Fehlerquadrat

$$E^p = \frac{1}{2}(t^p - y^p)^2 \quad . \quad (2.5)$$

⁷Gegeben sei ein n -dimensionaler Raum X . Zwei Teilmengen X_1 und X_2 von X sind (absolut) linear separabel, wenn es einen Wert $\Theta \in \mathbb{R}$ und n reelle Werte $w_1, \dots, w_n$ gibt, so daß für alle Punkte $(x_1, \dots, x_n) \in X_1$: $\sum_{i=1}^n w_i x_i > \Theta$ und für alle Punkte $(x_1, \dots, x_n) \in X_2$: $\sum_{i=1}^n w_i x_i < \Theta$ gilt.

Die Ausgabe ist gerade die gewichtete Summe der n Eingaben abzüglich des Schwellenwertes Θ , also

$$y^p = \left(\sum_{i=1}^n w_i x_i^p \right) - \Theta \quad . \quad (2.6)$$

Die Gewichtsänderung ist proportional zum Gradienten des Gesamtfehlers, also

$$\Delta w_i = -\eta \frac{\partial E_{\text{tot}}}{\partial w_i} = \eta \sum_{p=1}^P (t^p - y^p) x_i^p \quad . \quad (2.7)$$

Die Proportionalitätskonstante η kann hier ebenfalls als Lerngeschwindigkeitskoeffizient interpretiert werden. Der Ausdruck $(t^p - y^p)$ wird häufig mit δ^p bezeichnet und deshalb trägt der Lernalgorithmus auch den Namen Delta-Regel. Für η wird oft der Wert $1/n$ gewählt. Die Delta-Regel konvergiert, der Fehler behält allerdings einen Wert größer null bei. Adaline-Netzwerke werden zur Mustererkennung und als Assoziativspeicher benutzt. Ein gegebenes KNN mit n Eingabeeinheiten kann unter Verwendung der Delta-Regel maximal n linear unabhängige Muster fehlerfrei speichern.

Madaline Durch die Kombination mehrerer Adaline in einem Netzwerk erhält man ein Madaline-Netzwerk (engl. multiple adaline) [WL90]. Die Lernregel basiert auf dem Prinzip der minimalen Störung. Dabei werden die Eingaben jedes Elements gestört, indem die Eingabe um einen kleinen Betrag Δs verändert wird und die Änderung in der quadratischen Fehlersumme der Ausgabe beobachtet wird. Man erhält so

$$\Delta w_i = \eta \sum_{p=1}^P (t^p - y^p) x_i^p \frac{\Delta f}{\Delta s_i^p} \quad , \quad (2.8)$$

wobei Δf die Änderung der Ausgabe (Aktivierungsfunktion) ist. Madaline-Netzwerke erlauben die Realisierung von Abbildungen mit nichtlinearer Teilbarkeit, wie z.B. die XOR-Funktion.

Mehrschichten-Perzeptron Das mehrschichtige vorwärtsgekoppelte Netzwerk (engl. multilayer feedforward network (MLFF)) mit Backpropagation-Lernen (BP) wird auch als Mehrschichten-Perzeptron (engl. multilayer perceptron (MLP)) bezeichnet, weil es Perzeptron-Netzwerken mit mehreren Schichten sehr ähnlich ist. Es handelt sich um ein vollverknüpftes Netzwerk, das aus einer Eingabeschicht, einer oder mehreren inneren Schichten und einer Ausgabeschicht besteht (siehe Abb. 2.9).

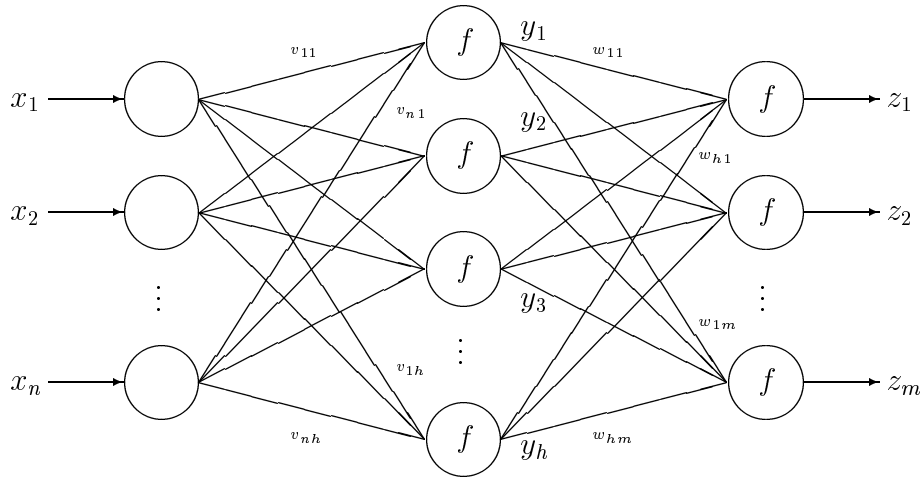


Abbildung 2.9: Schematische Darstellung eines MLP mit einer inneren Schicht (übersichtshalber sind nicht alle Gewichtungen eingezeichnet)

Als Aktivierungsfunktion wird häufig die sigmoide, S-förmige, monoton steigende und differenzierbare logistische Funktion (siehe Abb. 2.10)

$$f(x) = \frac{1}{1 + \exp(-b \cdot x)} \quad (2.9)$$

benutzt, wobei b eine Konstante ist, die die Steilheit der Kurve bestimmt. Der Wertebereich der Funktion liegt zwischen 0 und 1. Die Ableitung hat die Form

$$f'(x) = b \cdot f(x)(1 - f(x)) \quad (2.10)$$

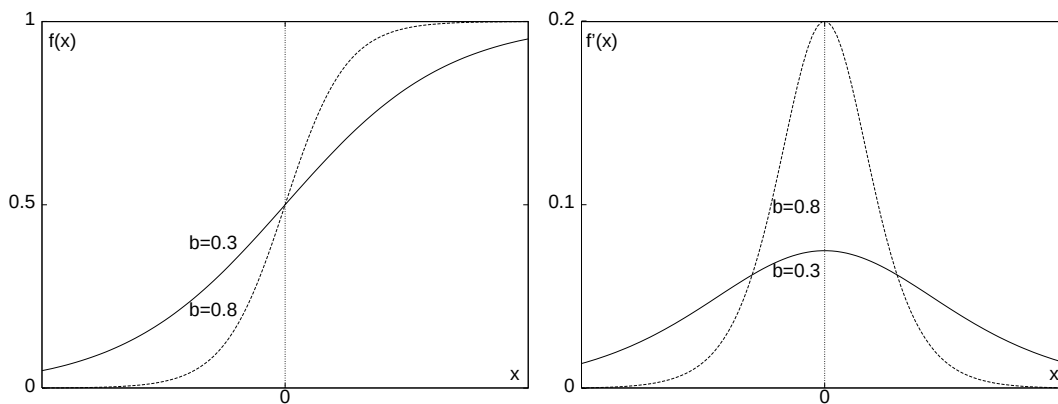


Abbildung 2.10: Darstellung der logistischen Funktion und ihrer Ableitung für zwei unterschiedliche Parameter b

Die Bestimmung der Gewichte der verborgenen Neuronen erfolgt mit der Backpropagation-Lernmethode. Es handelt sich um eine Optimierungsprozedur (Minimierung des Netzfehlers), die auf dem steilsten Gradientenabstieg basiert (vgl. Kap. 2.4). Die berechneten Fehler werden als Eingaben für Feedback-Verknüpfungen verwendet, welche dann rückwärts Schicht für Schicht, also iterativ, die Gewichtungen anpassen. Betrachtet man ein Netz mit n Eingabeneuronen x_i , h Neuronen der verborgenen Schicht y_j und m Neuronen der Ausgabeschicht z_k . Die Gewichte zwischen Eingabe- und verborgener Schicht sind v_{ij} , die zwischen verborgener und Ausgabeschicht w_{jk} . Folglich gilt für die Ausgabesignale

$$z_k = f \left(\sum_{j=1}^h w_{jk} y_j \right) = f \left(\sum_{j=1}^h w_{jk} f \left(\sum_{i=1}^n v_{ij} x_i \right) \right) . \quad (2.11)$$

Die zu minimierende Energiefunktion lautet

$$E_{\text{tot}} = \frac{1}{P} \sum_{p=1}^P E^p \quad \text{mit} \quad E^p = \frac{1}{2} \sum_{k=1}^m (t_k^p - z_k^p)^2 . \quad (2.12)$$

Die Gewichtsänderungen sind proportional zum Gradienten der Energiefunktion:

$$\Delta w_{jk} = -\eta \frac{\partial E_{\text{tot}}}{\partial w_{jk}} \quad \text{und} \quad \Delta v_{ij} = -\eta \frac{\partial E_{\text{tot}}}{\partial v_{ij}} . \quad (2.13)$$

Mit den Definitionen

$$H_j^p = \sum_{i=1}^n v_{ij} x_i^p \quad , \quad I_k^p = \sum_{j=1}^h w_{jk} y_j^p \quad \text{und} \quad (2.14)$$

$$\delta_k^p = (t_k^p - z_k^p) f'(I_k^p) \quad , \quad \delta_j^p = f'(H_j^p) \sum_{k=1}^m \delta_k^p w_{jk} \quad (2.15)$$

folgt aus Gl. 2.13

$$\Delta w_{jk} = \frac{\eta}{P} \sum_{p=1}^P \sum_{k=1}^m \delta_k^p y_j^p \quad \text{und} \quad \Delta v_{ij} = \frac{\eta}{P} \sum_{p=1}^P \sum_{j=1}^h \delta_j^p x_i^p . \quad (2.16)$$

Zu Backpropagation existieren viele Erweiterungen. Mit dem *Momentum Term* verhindert man, daß das Lernen in einem flachen Bereich des Fehlergebirges nur noch langsam vorangeht (vgl. Kap. 2.4). Man gibt dem Verfahren einen Schwung mit, indem man einen Bruchteil der letzten Gewichtsänderung nochmals addiert:

$$\Delta w_{kj}^p = \eta \sum_{k=1}^m \delta_k^p y_j^p + \alpha \Delta w_{kj}^{p-1} \quad \text{und} \quad \Delta v_{ij}^p = \eta \sum_{j=1}^h \delta_j^p x_i^p + \alpha \Delta v_{ij}^{p-1} . \quad (2.17)$$

Aufgrund der Vielseitigkeit und Abbildungsfähigkeit von MLPs gibt es zahlreiche Anwendungen in den Bereichen Funktionenapproximation, Klassifizierung, Diagnose, Steuerung, Optimierung, Vorhersage und Mustererkennung.

Hopfield-Netz Hopfield-Netzwerke [Hop82] sind einschichtige, rekursive Netzwerke mit symmetrischen Gewichtungsmatrizen, d.h. es gilt $w_{ij} = w_{ji}$ für alle $i, j = 1, 2, \dots, n$. Hopfield-Netze speichern eine Anzahl P von Prototypenmustern, die sogenannten Fixpunkt-Attraktoren. Die gespeicherten Muster können durch eine direkte Berechnung spezifiziert werden, wie etwa mit der Hebbischen Lernregel

$$\Delta w_{ij} = \eta x_i y_j \quad , \quad (2.18)$$

oder sie werden anhand eines Aktualisierungsschemas wie der Delta-Regel erlernt. Nachdem ein Netzwerk P Prototyp-Muster gelernt hat, können diese zum assoziativen Wiederfinden herangezogen werden. Um ein spezielles Muster zu finden, arbeitet das Netzwerk rekursiv, indem die Ausgabesignale des Netzwerks wiederholt in die Eingaben einfließen, und zwar zu jedem Aktualisierungszeitpunkt t , bis das Netzwerk sich schließlich stabilisiert. Die Lösung kann zyklisch sein mit fester Periodendauer T . Damit das Netzwerk als Assoziativspeicher dient, muß die Lösung jedoch in endlicher Zeitdauer gegen einen Fixpunkt $\vec{x}(t+1) = \vec{x}(t)$ konvergieren. Der Status des Netzwerks läßt sich durch die Energiefunktion

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j \quad (2.19)$$

beschreiben. Hopfield konnte zeigen, daß die Energie irgendwann einen stabilen Zustand erreichen muß, wenn sich das KNN gemäß seiner Dynamik entwickelt, weil die definierte Energiefunktion E nicht nach jeder Aktualisierung anwachsen kann. Sie muß kleiner werden oder zumindest gleich bleiben. Weil es eine endliche Anzahl Systemzustände gibt, muß das Netzwerk irgendwann gegen ein lokales Minimum konvergieren. Den Energieminima entsprechen Fixpunkt-Attraktoren, den gespeicherten Mustern. Der Status, den das System bei Konvergenz annimmt, bestimmt das Ausgabemuster.

Hopfield-Netze wurden hauptsächlich in Optimierungsanwendungen eingesetzt, wie etwa beim NP-Problem des Handlungsreisenden (engl. travelling salesman problem (TSP)), bei der Erstellung von Zeitplänen oder bei der Funktionsoptimierung.

Bidirektionaler assoziativer Speicher Der bidirektionale assoziative Speicher (engl. bidirectional associative memory (BAM)) [Kos87] ist einerseits ein generalisiertes, heteroassoziatives arbeitendes Hopfield-Netzwerk, das andererseits auch Ähnlichkeiten mit dem ART-Modell (s.u.) hat. Es besteht aus zwei Schichten, wobei die Elemente der Schichten nur binäre Zustände annehmen können. Die beiden Schichten sind in beide Richtungen vollständig miteinander verbunden, wobei die Gewichte symmetrisch sind. Das Lernziel eines BAM ist es, die

Eingabevektoren der Dimension n mit m -dimensionalen Ausgabevektoren innerhalb der Gewichtungsmatrix adäquat abzubilden. Der Lernschritt innerhalb eines BAM-Netzwerkes ähnelt dem des Hopfield-Netzes, d.h. es kann nach der Regel von Hebb oder der Delta-Regel trainiert werden. Das BAM läßt sich gut zur Musterergänzung einsetzen, wobei in jede der beiden Schichten ein Teilmuster eingegeben wird.

Brain-State-in-a-Box Das Brain-State-in-a-Box (BSB) [And72] ist ein rekursives, autoassoziatives Netzwerk, wobei der Systemzustand innerhalb des quadratischen, schachtelähnlichen Hyper-Bereichs, der von den Kanten $[+1, -1]$ begrenzt ist, gefangen ist. Das Netzwerk besteht aus einer einzelnen Schicht von n Einheiten. Es ist in der Struktur und Arbeitsweise ähnlich den Hopfield-Netzen. Die Einheiten dürfen Eigen-Feedback-Verbindungen $w_{ii} \neq 0$ haben, und einige Gewichtungsverbindungen dürfen weggelassen werden, d.h. für einige (i, j) darf gelten $w_{ij} = 0$. Für die Anpassung der Gewichtungen in einem BSB-Netzwerk kann das Hebbsche oder das Delta-Regel-Lernen angewendet werden. Wie auch bei den Hopfield-Netzen werden BSB-Netzwerke zur Mustererkennung und Optimierung eingesetzt.

Die Boltzmann-Maschine Die Boltzmann-Maschine [AHS85] ist ein rekursives, stochastisches Netzwerk. Die Statuszustände, die das Netzwerk annehmen kann, werden durch die Boltzmann-Verteilung bestimmt, einer exponentiellen Form der Wahrscheinlichkeitsverteilung, die für die Modellierung der Statuszustände eines physikalischen Systems bei thermischem Gleichgewicht genutzt wird. Wie das Hopfield-Netzwerk hat eine Boltzmann-Maschine eine symmetrische Gewichtungsmatrix, die Konvergenz gegen einen stabilen Status garantiert. Anders als das Hopfield-Netzwerk kann die Boltzmann-Maschine jedoch verborgene Einheiten haben. Während des Arbeitens und des Trainings der Boltzmann-Maschine wird eine Lernregel benutzt, die man als kontrolliertes Abkühlen (engl. simulated annealing) bezeichnet. Diese Methode ermöglicht dem Netzwerk lokale Minima zu verlassen und gegen einen globalen Gleichgewichtszustand zu konvergieren. Die Boltzmann-Maschine kann als Assoziativspeicher oder zur Lösung von Optimierungsproblemen verwendet werden.

Selbstorganisierende Karten Selbstorganisierenden (sensorischen) Karten (engl. self-organizing map (SOM)) werden auch nach ihrem Entwickler Kohonen-Netze genannt [Koh82]. Es sind einschichtige, rückkopplungsfreie Netzwerke, die i.a. bis zu zwei Dimensionen aufweisen, wobei jeder Eingang mit allen Elementen verbunden ist. Sie arbeiten auf der Basis des konkurrierenden, nicht-überwachten Lernens (vgl. Kap. 2.4). Eine topologische Struktur wird auf der Wettbewerbsschicht (Kohonen-Schicht) definiert, d.h. Neuronen der Wettbewerbsschicht wer-

den in Form eines Hyperquaders angeordnet. Das Ziel ist es, selbstorganisierende Karten so zu trainieren, daß benachbarte Cluster durch benachbarte Neuronen der Wettbewerbsschicht repräsentiert werden. Während des Lernprozesses wird dem Netzwerk eine Folge von Eingabemustern präsentiert, die in der Regel durch eine Wahrscheinlichkeitsverteilung erzeugt wurden. Für die Änderung der Gewichtungsfaktoren des aktivierten j -ten Neurons gilt

$$\Delta w_{ij} = \eta(x_i - w_{ij}) \quad (2.20)$$

unter der Nebenbedingung (Normierung)

$$\sum_{i=1}^n w_{ij} = 1 \quad . \quad (2.21)$$

SOM-Netzwerke werden zur Datenkomprimierung, kombinatorische Optimierung, Robotorsteuerung sowie Sprach- und Mustererkennung eingesetzt.

Lineare Vektor Quantisierung Die Vektorquantisierung ist ein Prozeß, stetig, reellwertige Vektoren $\vec{x}$ aus einer Menge $A \subseteq \mathbb{R}^n$ auf den nächsten Referenz-Gewichtungswert $\vec{w}_i$ aus der Menge $B \subseteq \mathbb{R}^m$ abzubilden. Die Eingabevektoren $\vec{x}$ der Dimension n werden in einer endlichen Anzahl von Klassen transformiert, wobei jede Klasse durch einen Prototyp-Vektor $\vec{w}_i$ dargestellt ist. Normalerweise versteht man unter nächstliegend den euklidischen Abstand

$$d(\vec{x}, \vec{w}_i) = \sqrt{\sum_{j=1}^n (w_{ij} - x_j)^2} \quad . \quad (2.22)$$

Vektorquantisierungs-Netzwerke arbeiten als konkurrierende Netzwerke. Wenn das Lernen überwacht erfolgt, nennt man sie auch lineare Vektorquantisierung (engl. linear vector quantization (LVQ)). Wenn die Klasse korrekt erkannt wurde, wird der Gewichtungsvektor der gewinnenden Einheit in Richtung des Eingabevektors verschoben. Wenn ein falscher Prototyp ausgewählt wurde, wird der Gewichtungsvektor vom Eingabevektor weg verschoben. Diese Verschiebungen sind abhängig von dem euklidischen Abstand (Gl. 2.22). Die LVQ wird in den Aufgabenbereichen Mustererkennung und Datenkomprimierung eingesetzt.

Das Kognitron Das Kognitron-Netzwerk [Fuk80] stellt eine hierarchische Struktur dar, die aus mehreren kaskadenförmig angeordneten, modularen Einheiten besteht, die Signale in Vorwärtsrichtung verarbeiten. Diese Netzwerkarchitektur wurde ursprünglich für Aufgaben der Bildverarbeitung entworfen und orientiert sich daher stark an das System des menschlichen Sehens. Die Eingabeschicht (Sensorschicht) ist ein rechteckiges Feld aus Rezeptorneuronen mit

lichtsensiblen Zellen und ist für die Eigenschaften auf Zell- oder Pixelebene verantwortlich. Zellen in höheren Schichten lernen, Low-Level-Eigenschaften (z.B. das Dach des Buchstaben A in der Zeichenerkennung) zu integrieren, und sind deshalb für globalere Eigenschaften zuständig. Die Integration setzt sich bis zur Ausgabeschicht fort, die vollständige Objekte im Bild identifiziert, wobei für jedes identifizierte Objekt eine Zelle erforderlich ist. Das Lernen wird schichtweise und konkurrierend ausgeführt, wobei die Zelle, die am meisten auf das Trainingsmuster reagiert, repräsentativ für diese Zellebene wird und deren Gewichtung so angepaßt wird, daß sie noch mehr auf das Muster reagiert. Diese Netzwerke werden hauptsächlich in Anwendungen für die invariante Zeichen- und Objekterkennung eingesetzt.

Counterpropagation Counterpropagation-Netzwerke [HN87a] sind dreischichtige, vorwärtsgekoppelte Netzwerke, die in nahezu perfekter Kombination zwei Lernstrategien vereinigen und die durch diese Kopplung wesentlich leistungsfähiger geworden sind, als Netzwerke mit den isolierten Einzelstrategien. Die Eingabeschicht dient dem Counterpropagation-Netz lediglich der Verteilung der Eingangsaktivitäten, sie führt keine Berechnungen aus. Jedes Eingabeneuron ist mit jedem Neuron der Kohonen-Schicht über ein Gewicht w_{ij} und letztere sind wiederum über ein Gewicht w_{jk} mit den Neuronen der Ausgabeschicht (Grossberg-Schicht) verbunden. Die Kohonenschicht wird nach dem Ansatz des konkurrierenden Lernen trainiert (winner-takes-all). Die Grossberg-Schicht dagegen wird durch überwachtes Lernen trainiert. Counterpropagation-Netzwerke werden im Bereich der Mustererkennung, Mustervervollständigung und Signalverbesserung genutzt.

Adaptive Resonanztheorie Die adaptive Resonanztheorie (ART) wurde als Erweiterung konkurrierender-kooperierender Lernsysteme entwickelt [Gro76]. Dabei wurde versucht, das Stabilitäts-Elastizitäts-Problem⁸ sowie andere instabile Lerneigenschaften von konkurrierenden Netzwerken zu umgehen.

ART-Netzwerke bilden n -dimensionale Eingabemuster auf Ausgabekategorien oder Klassen ab, die auf den Eigenschaften des Eingabemusters basieren. Ähnliche Eingabemuster (nächster Nachbar) werden in derselben Klasse gruppiert, nicht-ähnliche Muster in separaten, verschiedenen Klassen. Das Lernen in ART-Netzwerken erfolgt während der normalen Arbeitsweise des Netzwerkes in Echt-

⁸Damit KNN vergleichbar mit den biologischen Vorbildern arbeiten, müssen sie in der Lage sein, sinnvolle Informationen im Gedächtnis zu behalten, während gleichzeitig neue wichtige Informationen erlernt, irrelevante ignoriert und veraltete oder unwichtige vergessen werden. Mit andern Worten sollen diese Netzwerke einen hohen Stabilitätsgrad aufweisen, wenn sie adaptiv neue Kategorien oder Konzepte lernen, und gleichzeitig anpaßbar sein, um diese neuen Kategorien oder Konzepte zu erkennen und zu erlernen, also auch ein hohes Maß an Elastizität aufweisen. Diese beiden Ziele konkurrieren miteinander.

zeit. Dabei handelt es sich um eine Form stetigen, nicht-überwachten adaptiven Lernens, wobei automatisch eine neue Kategorie gebildet wird, wenn dem Netz ein neues Eingabemuster präsentiert wird. Es werden solange neue Kategorien für neue Eingabemuster erzeugt, bis das Netzwerk seinen Pool noch nicht verwendeter Ausgabekategorie-Neuronen erschöpft hat. Alle weiteren neuen Eingabemuster werden zurückgewiesen. Eingabemuster, die ähnlich bereits eingerichteten Kategorien sind, werden sofort erkannt, indem an dem Neuron der selektierten Kategorie eine hohe Ausgabe erzeugt wird. Eingaben, die mit existierenden Kategorien übereinstimmen, initiieren außerdem ein gewisses Lernen für diese Kategorie, ohne daß dabei die Stabilität der erlernten Kategorien erhöht wird. ART-Netzwerke werden in viele wichtige Anwendungen eingesetzt, u.a. für die Sensordatenfusion, Diagnose, Steuerung und Datenbankverwaltung für Entwurfszeichnungen.

2.6 Entwicklung und Optimierung

Die **Entwicklung** eines KNN kann man in folgenden Phasen einteilen [Füs96]:

- 1 Definition des Problems
- 2 Datenakquisition
- 3 Datenpräsentation
- 4 Wahl des Netzwerkmodells
- 5 Wahl der Netzwerkstruktur
- 6 Wahl der Netzparameter
- 7 Training des Neuronalen Netzes
- 8 Testen des trainierten Netzes
- 9 Anwendung

In Phase 1 muß untersucht werden, ob das Problem geeignet ist, um mit einem KNN bearbeitet zu werden. Die Beschaffenheit der Daten (Phase 2) muß quantitativen (ausreichende Anzahl) und qualitativen Anforderungen (weitgehende Fehlerfreiheit) genügen. In Phase 3 [Ste93] findet eine Codierung (binäre, ganze oder reelle Werte), eine Skalierung (symmetrische Häufigkeitsverteilung um den Mittelwert) und eine Normierung (lineare Abbildung auf das Intervall $[0, 1]$) der Daten statt. Bei Zeitreihen wird oft eine Differenzenbildung der Daten zur Trendbereinigung verwendet. Bei exponentiellem Wachstum der Daten wird zuvor logarithmiert. Die nächsten drei Phasen befassen sich mit der konkreten Ausarbeitung des KNN. Die Fragen, die man hier beantworten muß, sind u.a. wieviele Schichten mit wievielen Neuronen in der jeweiligen Schicht das KNN haben soll, wie die Verbindungen zwischen den Neuronen zu gestalten sind, oder welche Aktivierungsfunktionen verwendet werden können. In Phase 7 steht man möglicherweise vor der Auswahl mehrerer alternativer Lernregeln. Außerdem werden hier die Daten in drei disjunkte Mengen aufgeteilt: Trainings-, Validierungs- und Testmenge (vgl. Kap. 2.4). Die Reihenfolge der Trainingsdaten, die das Netz

zum Lernen benutzt, erfolgt nach Möglichkeit zufallsgesteuert. Beim Testen des Netzes (Phase 8) läßt sich anhand einer Sensitivitätsanalyse feststellen, welche Eingabedaten wirklich relevant, und welche Anzahl von verborgenen Schichten und Gewichten optimal sind. Nach dem Training und Testen gelangt das KNN schließlich zur Anwendung.

Der effektive Einsatz von neuronalen Systemen erfordert eine Feinabstimmung von Parameterwerten. Hierzu sind jedoch nur wenige heuristische Regeln bekannt. Daher ist man weitgehend auf eine experimentelle Vorgehensweise angewiesen.

Die **Optimierung** des Netzwerks wird auch als Netzwerkdesign bezeichnet und nimmt eine zentrale Stellung bei der Entwicklung KNN ein. Zur Bestimmung der notwendigen Anzahl von Neuronen in den inneren Schichten gibt es zwei grundsätzlich verschiedene Strategien, von denen sich die erste inzwischen weitgehend durchgesetzt hat [BH89, Cha89]:

- Start mit einem für die Aufgabenstellung zu groß dimensionierten Netzwerk und Entfernung aller überflüssigen Einheiten (Hecht-Nielson-Argument).
- Start mit einem für die Aufgabenstellung zu klein dimensionierten Netzwerks und bei Bedarf Hinzufügung weiterer Einheiten (network construction algorithms).

Einige unterschiedliche Optimierungsmethoden sind in Abb. 2.11 dargestellt und werden im folgenden kurz vorgestellt.

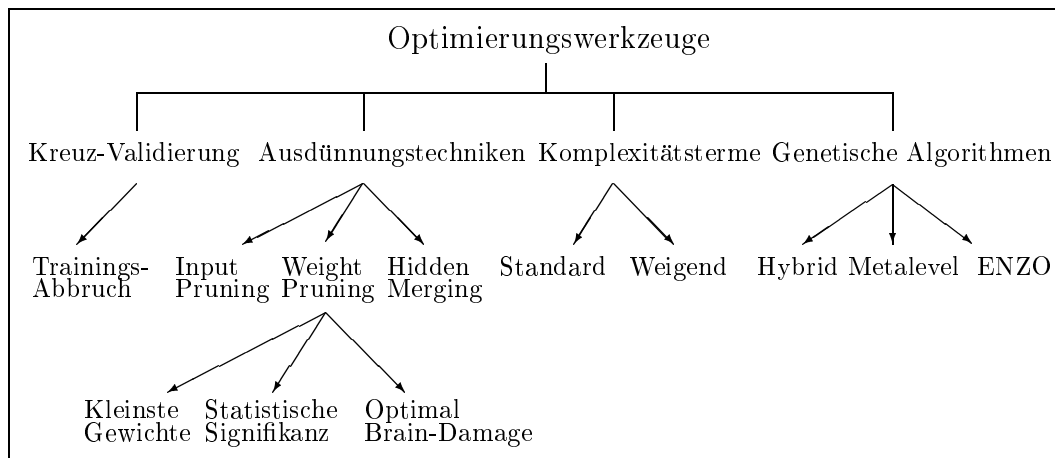


Abbildung 2.11: Optimierungswerkzeuge bei dem Design KNN

Kreuz-Validierung Diese Methode verhindert ein Überlernen des Netzwerks (vgl. Kap. 2.4). Die Datensätze werden dabei in Trainings- und Validierungsmenge unterteilt und der jeweilige Fehler des Netzwerkes zu diesen Mengen berechnet. Beginnt der Fehler der Validierungsmenge anzusteigen, wird das Training abgebrochen. Danach können andere Optimierungsmethoden eingesetzt werden.

Ausdünnungstechniken Es existieren unterschiedliche Möglichkeiten, dichte Komplexität im Netzwerk durch topologieverändernde Eingriffe zu reduzieren (Ausdünnungstechniken, engl. pruning techniques). Ansatzpunkt ist dabei das Gewicht oder die Ausgabe der Neuronen.

Aufgrund einer Sensitivitätsanalyse werden unwichtige Eingabeneuronen, die den Fehler des Netzwerks nur marginal beeinflussen, ausfindig gemacht und entfernt (*Input-Pruning*).

Unter *Weight-Pruning* sollen hier allgemein Verfahren verstanden werden, die geeignet sind, nicht benötigte Gewichte aus einem Netzwerk zu entfernen oder auf Null zu setzen. Die Standardmethode (*Kleinste Gewichte*) benutzt als Testgröße den Betrag des Gewichts. Bei dem Pruning nach *statistischer Signifikanz* [HFZ92] werden Informationen über die Verteilung (z.B. Standardabweichung) der Gewichtsveränderungen für jede Verbindung im Netzwerk betrachtet. Das *Optimal-Brain-Damage* [LDS89] ist eine Methode, die die zweite Ableitung der Fehlerfunktion bei der Gewichtsänderung berücksichtigt.

Eine andere Variante besteht darin, paarweise Korrelationsanalysen der Neuronenaktivitäten in verborgenen Schichten durchzuführen [ZHF92]. Zwei Neuronen mit hohen Korrelationskoeffizienten können dabei zusammengelegt werden (*Hidden Merging*). Die Betrachtung der Varianz der Neuronenaktivitäten erlaubt ferner, Neuronen zu identifizieren, die unabhängig vom anliegenden Eingabemuster stets dieselbe oder annähernd gleiche Ausgabe produzieren. Sie wirken faktisch als Schwellenwert und können komplett entfernt werden.

Komplexitätsterme Das Optimierungswerkzeug der Komplexitätsterme setzt auf der zu minimierenden Fehlerfunktion des Netzwerkes an und wird damit direkter Bestandteil des Lernverfahrens. Es gilt:

$$E_{\text{tot}} = \sum_{p=1}^P E^p + \lambda C(w_1, \dots, w_n) \quad , \quad (2.23)$$

wobei C der noch unspezifizierte, von den Gewichtungen w_i abhängige Komplexitätsterm und $\lambda > 0$ ein Parameter ist, der den Ausgleich zwischen der Anpassung an die Trainingsdaten und dem dabei beobachteten Zuwachs an Komplexität beschreibt. Die älteste Form der Modellierung, bekannt unter dem Namen *Standard-Gewichtszерfall-Verfahren* (engl. weight decay), bestraft größere Gewichte. Der Strafterm wächst quadratisch bei einer trainingsbedingten linearen Gewichtsvergrößerung:

$$C(w_1, \dots, w_n) = \sum_{i=1}^n w_i^2 \quad . \quad (2.24)$$

Demgegenüber bestraft der *Weigend*-Strafterm zwar ebenso größere Gewichte, die Bestrafung wächst jedoch nicht (quadratisch) unendlich bei einer trainingsbedingten linearen Gewichtsvergrößerung, sondern besitzt eine natürliche Obergrenze [WRH91]:

$$C(w_1, \dots, w_n) = \sum_{i=1}^n \left(\frac{\left(\frac{w_i}{w_0}\right)^2}{1 + \left(\frac{w_i}{w_0}\right)^2} \right) \quad . \quad (2.25)$$

Genetische Algorithmen Genetische Algorithmen (GA) stellen globale Suchverfahren dar, die sehr gut in Gradientenabstiegsverfahren eingesetzt werden können. Bei GA wird die Suche des globalen Minimums des Netzwerkfehlers gleichzeitig an verschiedenen Stellen des Fehlerraums aufgenommen und betrieben, denn jedes Populationselement stellt einen solchen Punkt dar. Das bedeutet, daß die Mechanismen der natürlichen Auslese und der biologischen Genetik veranlassen, daß die einzelnen Optimumsucher, mittels eines globalen Suchverfahrens zufallsgesteuert, kreuz und quer durch die Suchlandschaft springen. GA sind in der Lage die Informationen vorhergegangener Generationen effizient auszunutzen, um neue Punkte im Fehlerraum mit guten Erfolgsaussichten zu finden.

*Hybride Lernmodelle*⁹ benutzen GA, um relativ schnell grobe Punkte im Fehlerraum zu finden, um dann mit binärer Codierung diskreter Gewichte oder adaptiver Mutationsrate und verteilter genetischer Suche eine Feinabstimmung zu betreiben, die insgesamt wesentlich schneller zu dem minimalen Netzwerkfehler führt.

Metalevel [Fri93, SN93] ist ein GA, der automatisch mathematische Formeln generiert, die als Repräsentation der Topologie KNN interpretiert werden können. Durch binäre Verschlüsselung der Formeln als Chromosome werden die Voraussetzungen für die Anwendungen GA geschaffen.

Der evolutionäre Netzwerkoptimierer (*ENZO*) bestimmt simultan die optimale Netzwerktopologie eines vorwärtsgekoppelten Netzes und die optimale Anzahl der Gewichte [BW93]. Das Verfahren erzeugt und trainiert zunächst verschiedene Netzwerke. Jedes Gen entspricht dabei genau einer Verbindung. Nach dem Training werden die Netze nach ihrer Eignung aufgereiht. Um zwei Eltern zu rekombinieren, werden alle Gene einzeln überprüft. Sind sie bei beiden Netzen gesetzt, so wird mit einer hohen Wahrscheinlichkeit (je nach Mutationsrate) auch der Nachkomme diese Verbindung aufweisen. Die so entstandenen Nachkommen werden wieder trainiert und in der bestehenden Reihe eingeordnet, wobei die schlechtesten Netzwerke ausscheiden, usw.

⁹Kennzeichnend für ein hybrides System ist die Kombination aus unterschiedlichen Bereichen wie Operation Research, Künstlicher Intelligenz und Konnektionismus, um dadurch Synergieeffekte zu erzielen [Mec95].

3 Ökonomische Anwendungen

Die Anwendungsbereiche Prognose, Datenanalyse und Mustererkennung i.e.S. lassen sich zu einer Gruppe Musterverarbeitung (Funktionsapproximation (vgl. Kap. 2.2), Zuordnungs- oder Klassifikationsproblemen) zusammenfassen (siehe Abb. 3.1), wenn man von der Struktur der Eingabedaten abstrahiert. Alle diese Anwendungen haben die Gemeinsamkeit, daß sie überwacht lernen.

Eine eigene Gruppe bilden die Anwendungen aus dem Bereich Optimierung. Der wesentliche Unterschied liegt im verwendeten Verarbeitungsprinzip. Während die anderen Anwendungen anhand einer Trainingsmenge angelernet werden – die KNN also Strukturen, die für eine Problemlösung notwendig sind, selbst erlernen – werden die KNN zur Optimierung nicht trainiert. Bei diesen Netzen wird nur spezifiziert, welche Eigenschaften eine Lösung besitzen muß, das Generieren der Lösung übernimmt das Netz.

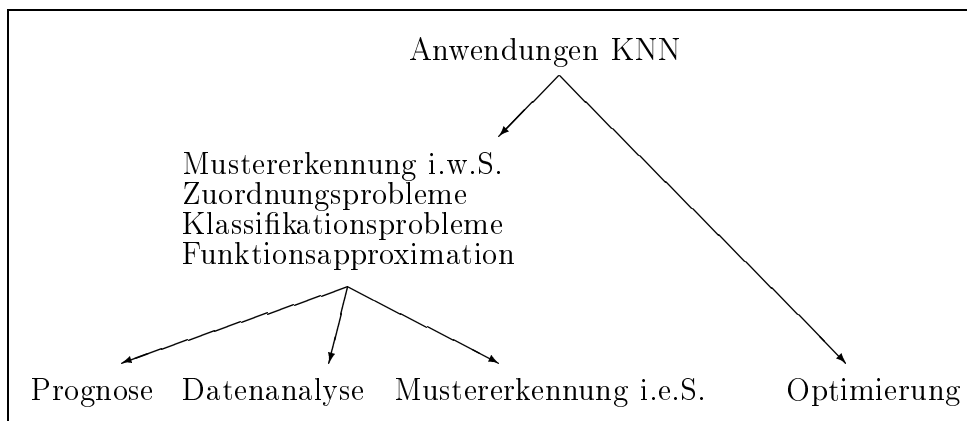


Abbildung 3.1: Hierarchisches Klassifikationsschema für wirtschaftliche Anwendungen KNN [Loh94]

Allgemein kann gesagt werden, daß sich die wirtschaftlichen Anwendungen KNN häufig in solchen Problembereichen etablieren, in denen eine algorithmische Formulierung der Problemlösungsfähigkeit schwierig oder sogar unmöglich ist. Die meisten Anwendungen kommen aus den Bereichen der Finanz-, Produktions- und Absatzwirtschaft.

3.1 Prognose

Bei wirtschaftlichen Prognosen besteht die Aufgabe darin, einen Ausschnitt aus der Wirklichkeit zu modellieren, um Vorhersagen (Prognosen) über zukünftige Entwicklungen in diesem Ausschnitt treffen zu können. Prognoseanwendungen von KNN basieren typischerweise auf Zeitreihenanalysen, d.h. einer Zahlenfolge

liegt ein funktionaler Zusammenhang zugrunde, der in einem mathematischen Modell abgebildet wird. Für nicht-deterministische Zeitreihen existieren allerdings keine exakten Modelle, mit denen eine sichere Vorhersage zukünftiger Werte möglich ist. Aus diesem Grund setzt man für Prognosen u.a. statistische Verfahren (Glättung, Extrapolation) ein. Einige dieser Verfahren unterstellen restriktive Annahmen über die zugrundeliegende Funktion (z.B. Linearität). KNN weisen gegenüber diesen Verfahren den Vorteil auf, keine expliziten Annahmen über die Zusammenhänge in den Eingabedaten zu machen. Allerdings beeinflusst die Wahl der Netzwerkkonfiguration implizit die Fähigkeiten eines KNN. Ein KNN zur Vorhersage zukünftiger Entwicklungen wird mit Gegenwarts- und Vergangenheitsdaten trainiert (ex-post Analyse). Es lernt also, wie sich Zustände der Vergangenheit entwickelt haben und überträgt diese Entwicklung auf die Gegenwart oder Zukunft. Prinzipiell sind alle KNN, die mittels Assoziationen lernen, für diese Aufgaben anwendbar. In der Praxis findet man jedoch (fast) ausschließlich zwei Netztypen: Backpropagation- und Kohonen-Netze.

Eine typische Prognose-Anwendung in der Finanzwirtschaft ist die Aktienkursprognose. Im nächsten Kapitel wird anhand von verschiedenen Beispielen diese Anwendungsform ausführlich besprochen. Statt Aktienkurse lassen sich z.B. auch Devisenkurse [RP90a, RP91, BS92, WdG92, WHR92, Pod93, Kna94, PW94, RP94, RK95, Mec95] und Zinsentwicklungen [Wil91b, Wil94, Rie92, Rie94, Mat94, Pod94, RK95, Füs95] oder volkswirtschaftliche Kenngrößen [All95] mit Hilfe von KNN vorhersagen.

Innerhalb der Produktionswirtschaft ist es möglich, Lagerbestände oder die Auftragsentwicklung vorherzusagen. Als Anwendungen der Absatzwirtschaft kommen die Prognose der Umsatzerwartungen [SHG90, HZ92], Preis- oder Kostenentwicklungen [Bec94] in Betracht.

3.2 Datenanalyse

Zu diesem Aufgabentyp zählen alle klassifikatorischen Anwendungen. Die Klassifikation ist eng mit der Prognose (vgl. Kap. 3.1) verknüpft. Das Ziel einer Klassifikation ist letztendlich ebenfalls eine Prognose (z.B. Klassifikation von Jahresabschlüssen in der Insolvenzprognose). Der wesentliche Unterschied zwischen Prognose- und Klassifikationsanwendungen liegt in der Art der Eingabe- und Ausgabedaten. Im Gegensatz zu Prognosen bestehen die Eingabesätze bei Klassifikationen aus statischen, zeitpunktbezogenen Daten (z.B. abgeschlossenen Kreditfällen). Der Prognosewert einer Klassifikation entspricht daher auch nicht einer Zeitreihenfortsetzung. Prognosen durch Klassifikation beruhen auf der Annahme, daß die vergangenheitsbezogenen Daten typische Muster enthalten, die erstens eine Klassifizierung mittels einer Trennfunktion erlauben und zweitens, daß gefundene Trennfunktionen auch für künftige, unbekannte Muster gültig sind.

Typische betriebswirtschaftliche Anwendungen sind in der Finanzwirtschaft (z.B. Risikomanagement) und in der Produktionswirtschaft (Qualitätskontrolle und -sicherung) zu finden. Einsatzgebiete im Risikomanagement sind u.a. Insolvenzprognosen¹⁰ [RP92a, SW93, Reh94, Reh95b], Bonitätsprüfungen¹¹ von individuellen Schuldern (Konsumentenkreditvergabe) [Wil91a, SLB92, RP92b, RvR93, MMM93, vRR94] oder von institutionellen Schuldern (Unternehmenskreditvergabe bzw. Unternehmensklassifikation) [OS90, RSR91, CGL91, E^{ea}92, CF93, B^{ea}94, KP94, WS94, MMM94, AMV94, Pod95, R^{ea}96], Bewilligung von Kreditkarten [Did95] oder die Vorhersage von Bankkonkursen [TK90, TK91, TK92, dBS95].

In der produktionswirtschaftlichen Qualitätskontrolle finden sich sehr spezielle Anwendungen [Hoh91]. Ein Beispiel für die akustische Qualitätskontrolle stellen die Überprüfungen von Verbrennungsmotoren dar [Lut93]. In der optischen Qualitätskontrolle werden KNN z.B. zur Überprüfung von Verschlusskappen auf Flaschen oder in der Stahlindustrie zur Überprüfung von Schweißnähten verwendet [MHP95]. Andere Datenanalysen mittels KNN in der Produktionswirtschaft sind in der Konstruktionsunterstützung [BP93], Angebotsplanung, Kosten- und Lieferterminabschätzung, Personalbedarfsplanung [Fai93] zu finden.

Im Bereich der Absatzwirtschaft werden KNN zur Marketingunterstützung durch Trenderkennung [RM93], Ermittlung von Marktreaktionsfunktionen [Hru91, HN93] und Modellierung des Kaufverhaltens [GDW94] eingesetzt.

3.3 Mustererkennung i.e.S.

Die Mustererkennung i.e.S. ist in die Bereiche der Schrift-, Sprach- und Bildverarbeitung untergliedert. Die Anwendungen in der Finanzwirtschaft sind i.d.R. optische Schrifterkennung (engl. optical character recognition (OCR)). Ein typisches Beispiel ist die Unterschriftenerkennung [Bar91]. Die Verarbeitung sensorischer (akustischer, optischer) Eingabedaten zur visuellen und auditiven Mustererkennung bildet die Grundlage in der Robotik und der Steuerungs- und Regelungstechnik, die in der Produktionswirtschaft im Bereich der Fertigung eingesetzt werden.

¹⁰Versuche, mit Hilfe ausgewählter Kennzahlen (z.B. Liquiditäts-Kennzahl, Kosten im Verhältnis zu Erlös, Umsatz je Kunde, usw.), die z.T. auch mehr oder weniger systematisch verknüpft werden, mittels mathematisch-statistischer Verfahren unter Einsatz von EDV-Anlagen Anhaltspunkte für die Wahrscheinlichkeit von erheblichen Kapitalverlusten vorauszusagen [Gab93].

¹¹Prüfung (vor Vertragsabschluß) der Fähigkeit des Schuldners, in Zukunft seinen Schuldendienstverpflichtungen nachzukommen. Seitens Banken spricht man auch von Kreditwürdigkeitsprüfungen [Gab93].

3.4 Optimierung

KNN stellen im Bereich der Optimierungsprobleme eine Alternative zu den bekannten Lösungsverfahren aus dem Operations Research und der Statistik dar. Die Funktionsweise eines KNN bei Optimierungen ist vergleichbar mit einer Heuristik, KNN finden u.U. ebenfalls nur suboptimale Lösungen. Allerdings können KNN abhängig von der fallspezifischen Aufgabenstellung z.T. schneller (bei Parallelisierung) oder sicherer eine Optimierung vornehmen. Im Gegensatz zu einigen Verfahren des Operations Research ist es in bestimmten Netzwerkmodellen möglich, ein lokales Minimum wieder zu verlassen.

Ein klassischer Anwendungsfall der Absatzwirtschaft findet sich in der Tourenplanung (im einfachsten Fall das Problem des Handlungsreisenden). Bei der Tourenplanung sollen von einem Standort ausgehend die kürzeste Route zwischen mehreren räumlich verteilten Kunden für eine bestimmte Anzahl von Fahrzeugen gefunden werden (Rundreisen einzelner Fahrzeuge). Dabei treten zwei Teilprobleme auf: die eindeutige Zuordnung von Kunden zu einem Fahrzeug und die Ermittlung der kürzesten Strecke für eine bestimmte Rundreise. KNN lösen beide Teilprobleme simultan [PT92, RS94].

Ein weiteres Beispiel für Optimierungsanwendungen besteht in der Maschinenbelegungsplanung der Produktionswirtschaft. Der Optimierungsansatz basiert auf einer zu minimierenden Kostenfunktion. Das KNN spiegelt dabei unzulässige und zulässige Lösungen des Problems wieder und lernt die optimale Lösung durch Minimierung einer Zielfunktion unter Restriktionen [RS92, Loh94, CM95, May96].

4 Aktienanalysen

Anlageentscheidungen basieren auf den Erwartungen der Marktteilnehmer über die zukünftige Entwicklung an den Kapitalmärkten¹². Die Aktienanalyse dient der Begründung der Anlageentscheidungen für Aktien¹³ (engl. stocks). Aufgabe der Aktienanalyse ist die Auswahl der Titel sowie des Zeitpunktes für den Kauf und Verkauf unter Berücksichtigung des jeweiligen Anlageziels, also der Erwirtschaftung einer möglichst hohen Rendite (Kurssteigerungen und Dividendenausschüttungen) bei vorgegebener Risikoneigung und Liquidität.

4.1 Die Random-Walk-Hypothese

Bereits 1827 fand der Botaniker Robert Brown unter dem Mikroskop, daß Schwebeteilchen in Pflanzenzellen eine sehr zittrige und ungeordnete Bewegung ausführen, die sogenannte Brownsche Bewegung. 1905 interpretierte der Physiker und Nobelpreisträger Albert Einstein die Brownsche Bewegung auf Grundlage der kinetischen Theorie der Materie. Die Bewegung eines Teilchens ist zu jedem Zeitpunkt unabhängig von der Bewegung zu einem anderen Zeitpunkt, also hat das Teilchen kein Gedächtnis. Die erwartete zeitliche Änderung in der Lage des Teilchens ist gleich Null, d.h. das Teilchen hat keine bevorzugte Bewegungsrichtung. Dagegen ist der mittlere quadratische Abstand, den das Teilchen bei seiner Bewegung zurücklegt, proportional zu der Zeitdauer der Bewegung. Die Proportionalitätskonstante ist der Diffusionskoeffizient. Deshalb wird die Brownsche Bewegung auch als Diffusionsprozeß verstanden. Den mathematischen Hintergrund der Diffusion liefert die Stochastik, also die Theorie zufälliger Prozesse.

Überträgt man dieses Konzept nun auf den Aktienmarkt, dann werden Aktien statt Schwebeteilchen betrachtet und die Teilchenbewegungen entsprechen den Aktienkursänderungen. Untersuchungen dazu hat Louis Bachelier bereits 1900 im Rahmen seiner Dissertation durchgeführt [Bac00] und das Ergebnis als Random-Walk-Hypothese formuliert. Im Mittelpunkt steht die Aussage, daß alle Kursänderungen reine Zufallsbewegungen sind. Bachelier geht davon aus, daß sich der Börsenkurs einer Aktie aus einem in Geldeinheiten bewerteten Mittelwert aller Nachrichten aus der Vergangenheit und Erwartungen für die Zukunft zusammensetzt. Jeder Marktteilnehmer bestimmt den inneren Wert einer Aktie und kommt so zu einer Schätzung des Börsenkurses. Die Differenz aus beiden

¹²Der Kapitalmarkt ist ein Markt für langfristige, durch Wertpapiere verbriefte Kredite. Er ist unterteilt in Aktien- und Rentenmarkt. Auf letzterem werden festverzinsliche Wertpapiere, auch Renten genannt, gehandelt. Zu diesen Papieren gehören Obligationen und Anleihen (engl. bonds). Die Börse ist die organisierte Form des Kapitalmarkts.

¹³Die Aktie ist ein Anteils- oder Teilhaberpapier, welches ein Mitgliedsschaftsrecht des Aktionärs an einer Aktiengesellschaft (AG) in einer Aktienurkunde verbrieft. Der Aktionär wird Teilhaber am Aktienkapital und damit Mitinhaber des Gesellschaftsvermögens.

ist aber rein zufällig. Schwankungen der Kurse werden durch neue Informationen hervorgerufen. Diese Informationen werden von den einzelnen Anlegern unterschiedlich bewertet, was zur Folge hat, daß sie in ihrer Summe wiederum als zufällige Größe in den Börsenkurs einfließen. Auch der Zeitpunkt, wann sich eine Information auf den Kurs einer Aktie auswirkt, ist rein zufällig. Zusammengefaßt sind somit alle Ereignisse am Aktienmarkt völlig unabhängig voneinander. Folglich kann auch nicht aus der Vergangenheit auf die zukünftige Entwicklung eines Kurses geschlossen werden, der Kurs hat kein Gedächtnis (s.o.), die Kursprognose verliert ihren Wert. Das bedeutet allerdings nicht, daß an der Börse niemand gewinnen oder verlieren kann, sondern daß sich in die möglichen Gewinne und Verluste ausgleichen (Nullsummenspiel).

Bacheliers Hypothese ist allerdings nicht ganz schlüssig. Die Existenz der fundamentalen Aktienanalyse (vgl. Kap. 4.2) wird bereits vorausgesetzt. Nur aufgrund der Tatsache, daß sich viele Anleger mit der Bestimmung des richtigen Börsenkurses beschäftigen, erscheinen die Aktienkurse einem Außenstehenden als rein zufällig. Würde sich niemand mehr mit der Aktienanalyse befassen, so wären sinkende Kurse die Folge. Aber gerade dann kann bei unveränderter Dividendenhöhe der Aktienerwerb aus Renditegesichtspunkten wieder interessant werden.

Die Random-Walk-Hypothese ist immer noch Forschungsgegenstand [Coo64, Fam65, ACW97], oft wird jedoch die Hypothese des effizienten Marktes benutzt. Diese stützt sich auf die Annahmen, daß der Markt vollkommen ist, die Teilnehmer rational handeln, die Preisänderungen unabhängig voneinander sind und der Markt einem Random-Walk folgt. Wenn der Aktienmarkt der Hypothese des effizienten Marktes genügt, ist es gerechtfertigt, ihn mit den Methoden der Stochastik zu analysieren. Im Mittelpunkt steht dabei die Untersuchung der zeitlichen Unabhängigkeit von Kursänderungen. Ein stochastischer Prozeß, der genau diese Eigenschaft besitzt, wird auch als Markov-Kette bezeichnet. Die ersten zwei Annahmen der Theorie effizienter Märkte treffen aber auf Kapitalmärkte nicht uneingeschränkt zu. Somit sind die Ergebnisse der Aktienmarktanalyse mit Hilfe stochastischer Methoden vor dem Hintergrund dieser Ineffizienz zu interpretieren.

4.2 Die fundamentale Aktienanalyse

Die fundamentale Aktienanalyse (FA) vertritt die Auffassung, daß die Entwicklung der Aktienkurse von der Entwicklung gesamtwirtschaftlicher, monetärer, branchenspezifischer und unternehmensbezogener Größen bestimmt wird. Als kursbestimmender Faktor für eine Aktie wird der *innere Wert* des jeweiligen Unternehmens angenommen, der aus internen und externen Unternehmensdaten zu bestimmen ist. Dabei wird unterstellt, daß bei rationalem Verhalten der Marktteilnehmer im wesentlichen das Ertragsstreben die Entscheidung über den Kauf und Verkauf einer Aktie bestimmt. Der innere Wert muß nicht mit dem Kurswert

der Aktie identisch sein, vielmehr schwankt der Kurs der Aktie um diesen. Ist der innere Wert größer als der Kurswert, dann wird die Aktie zum Kauf empfohlen.

Die gesamtwirtschaftliche Analyse Zwischen der Bewegung der Aktienkurse und den konjunkturellen Schwankungen der volkswirtschaftlichen Nachfrage und Produktion sowie der Kapitalakkumulation besteht ein systematischer Zusammenhang. Der Konjunkturzyklus ist dadurch gekennzeichnet, daß nahezu alle volkswirtschaftlich relevanten Größen zyklisch um ihre trendmäßige Entwicklung schwanken. Um Aussagen über die voraussichtliche Kursentwicklung machen zu können, muß man sowohl die konjunkturellen Einflußfaktoren kennen, als auch die Frühindikatoren der Konjunktur.

Einen der wichtigsten Frühindikatoren für die Konjunkturentwicklung stellt die Entwicklung der Auftragseingänge dar, die monatlich vom Statistischen Bundesamt veröffentlicht wird. Der Auftragseingang von heute bestimmt die Produktion von morgen. Somit liefern die Auftragseingänge einen wichtigen Hinweis auf die künftige Wirtschaftsentwicklung. In diesem Zusammenhang stellt die Auslastung der Kapazitäten einen weiteren Frühindikator dar. Denn bei voll ausgenutzten Kapazitäten ist eine Produktionssteigerung nur mit steigenden Durchschnittskosten verbunden (zusätzliche Investitionen).

Zu den Konjektуреinflußgrößen gehören u.a. Investitionen, die Entwicklung des Arbeitsmarktes, die Konsumneigung, Wechselkurse und Preise. Hohe Investitionen der Industrie signalisieren, daß die Wirtschaft die Wachstumschancen günstig einschätzt. Bei der Beurteilung der Arbeitsmarktlage darf nicht allein die Zahl der Arbeitslosen betrachtet, sondern es muß auch das Ausbildungsniveau berücksichtigt werden. Es gibt Phasen, in denen die stärksten Konjunkturimpulse vom Konsum ausgehen. Über die zukünftige Entwicklung des Konsums gibt z.B. der Konsumklimaindex der Nürnberger Gesellschaft für Konsumforschung (GfK) Hinweise. Gerade für die Bundesrepublik Deutschland mit einem hohen Exportanteil am Bruttosozialprodukt (BSP) ist die Entwicklung der Wechselkurse von großer Bedeutung. Ein hoher DM-Kurs verteuert deutsche Produkte im Ausland und erschwert die Absatzchancen oder drückt die Gewinnspanne. Er senkt aber auch die Kosten für importierte Rohstoffe oder Vorprodukte. Gegenüber den im Europäischen Währungssystem (EWS) zusammengeschlossenen Währungen sind Wechselkursschwankungen nur innerhalb gewisser Grenzen möglich. Dagegen kann der Kurs der DM aber gegenüber dem US-Dollar und gegenüber dem japanischen Yen frei schwanken. Stark steigende Preise können dazu führen, daß die Notenbank¹⁴ weniger Kredite vergibt, was die konjunkturelle Entwicklung stören

¹⁴Die Notenbank ist eine zur Notenausgabe berechnigte Bank. In den meisten Ländern ist es nur eine Bank, die Zentralbank (in der Bundesrepublik Deutschland ist es die Deutsche Bundesbank). Die Notenausgabe ist ein passives Kreditgeschäft der Notenbanken. Die ausgegebenen Banknoten werden zu Goldankäufen und zur Kreditgewährung verwendet.

würde. Außerdem hat beispielsweise die Entwicklung des Erdölpreises erheblichen Einfluß auf die weltweite Konjunktur. Höhere Erdölpreise dämpfen i.d.R. die Konjunkturentwicklung, niedrigere Preise regen die Wirtschaft an.

Die monetäre Analyse Die monetäre Analyse stellt eine wesentliche Erweiterung der Konjunkturanalyse dar. Geldpolitik kann als Summe aller Maßnahmen der Zentralbank zur Konjunktursteuerung verstanden werden. Geldpolitischer Handlungsbedarf besteht besonders in Zeiten der Rezession und der Hochkonjunktur. Ziel der Geldpolitik ist es, andere volkswirtschaftliche Größen, wie etwa die Inflationsrate¹⁵, die Investitionstätigkeit und die Arbeitslosigkeit, zu beeinflussen. Die Monetaristen sehen in der Geldmenge und den Zinsen die maßgeblichen Einflußfaktoren für die Kursentwicklung an den Aktienmärkten.

Die Instrumente der Geldpolitik der Notenbanken zur Beeinflussung des Kapitalmarktes und zur Steuerung der Geldmenge¹⁶ sind im wesentlichen der Diskont¹⁷- und Lombardsatz¹⁸, die Mindestreservesätze¹⁹ und die Offenmarktpolitik²⁰.

Niedrige oder sinkende Zinsen aufgrund einer reichlichen und zunehmenden Liquiditätsausstattung verstärken die konjunkturellen Aufwärtskräfte. Sie führen bei den Unternehmen zu niedrigeren Produktions-, Lager- und Investitionskosten und bringen steigende Gewinne. Zinsänderungen beeinflussen den Kapitalmarkt aber auch direkt. Steigende Zinsen machen festverzinsliche Wertpapiere im Vergleich zu Aktien attraktiver und dämpfen die Bereitschaft, auf Kredit zu kaufen. Ein wichtiger Indikator für die Liquiditätsentwicklung einer Volkswirtschaft ist das Wachstum der Geldmenge. Die Deutsche Bundesbank veröffentlicht regelmäßig

¹⁵Als Geldwertstabilität bezeichnet die Deutsche Bundesbank eine Inflationsrate, also die Geldentwertung, unter 2%.

¹⁶Die Deutsche Bundesbank definiert die Geldmengen M1, M2, M3 und die Zentralbankmenge. Der Geldmenge M3 wird aus Sicht der Deutschen Bundesbank die höchste Aussagekraft zugeschrieben. M3 setzt sich zusammen aus Bargeld- und Sichteinlagenbeständen inländischer Nichtbanken, Termineinlagen inländischer Nichtbanken mit einer Befristung unter 4 Jahren und Spareinlagen inländischer Nichtbanken mit gesetzlicher Kündigungsfrist.

¹⁷Von der Zentralbank festgesetzter Zinssatz, mit dem sie im Rahmen der Diskontpolitik einen in n Tagen fällig werdenden und ihr heute zum Rediskont von einer Bank eingereichten Wechsel abdiskontiert. Der Diskontsatz ist für die Geschäftsbanken ein Leitzinssatz für die Kreditvergabe an Unternehmen und Haushalte.

¹⁸Neben dem Diskontsatz ist der Lombardsatz eine weitere Leitzinsgröße. Der Lombardsatz ist der Zinssatz, zu dem die Banken zur Geldbeschaffung lombardfähige Wertpapiere bei der Notenbank verpfänden können. Der Lombardsatz liegt i.d.R. über dem Diskontsatz.

¹⁹Die Banken sind gezwungen, entsprechend der Höhe ihrer Kundeneinlage eine bestimmte Mindestreserve bei der Zentralbank zu hinterlegen. Da die Höhe der Kundeneinlagen täglich schwankt, wird in der Praxis so verfahren, daß die Banken mit zu hohen Mindestreserven den Banken mit zu geringen Mindestreserven ihre Überschüsse (Tagegeld) überlassen.

²⁰Im Rahmen der Offenmarktpolitik greift die Zentralbank direkt am Kapitalmarkt ein. Der Begriff Offenmarktpolitik beschreibt den Kauf und Verkauf von Schuldverschreibungen, Wechseln und festverzinslichen Wertpapieren mit der Absicht, die Geldversorgung zu beeinflussen und die Feinsteuerung des Zinsniveaus zu erreichen.

einen Zielkorridor für den Zuwachs der Geldmenge. Sie gibt damit zu erkennen, wie sie die Konjunktur einschätzt, wieviel an Liquidität die Wirtschaft aus Sicht der Währungshüter für ein weitgehend inflationsfreies Wachstum benötigt.

Die Branchenanalyse Die Branchenanalyse dient in erster Linie dazu, eine Auswahl von Gesellschaften für die sich anschließende Unternehmensanalyse zu treffen. Die wirtschaftliche Entwicklung einer Branche muß nicht mit der gesamtwirtschaftlichen Konjunkturentwicklung parallel verlaufen. Wie für die gesamtwirtschaftliche Entwicklung, sind es die Auftragseingänge und die Industrieproduktion, aus denen sich Hinweise auf die zu erwartende Umsatz- und Gewinnentwicklung einer Branche ergeben. Für die Börse und die Kursentwicklung innerhalb der einzelnen Branche ist die Zinsentwicklung eine weitere Bestimmungsgröße. Dies zeigt sich besonders bei Unternehmen der Versorgungswirtschaft, deren hohe Investitionssummen zum größten Teil fremdfinanziert sind, was entsprechend große Zinsbelastungen mit sich bringt. Außerdem ist die Währungsentwicklung für die Exportunternehmen von Bedeutung. Betroffen sind vor allem der Maschinenbau, die Elektro- und Automobilindustrie und die Chemie. Versorger und Bauindustrie sind i.a. so gut wie frei von Währungseinflüssen.

Die Unternehmensanalyse Die Bewertung einzelner Aktien orientiert sich üblicherweise an der Ertragskraft einer Gesellschaft. Grundlage stellt der Jahresabschluß dar. Aus der Bilanz, der Gewinn- und Verlustrechnung sowie dem Geschäftsbericht werden die Daten des Unternehmens im Rahmen der Bilanzanalyse aufbereitet und in betriebswirtschaftliche Kennzahlen verdichtet. Die Bilanzanalyse setzt sich aus Strukturanalyse (Vermögen-, Kapital- und Finanzstruktur), Finanzanalyse (Liquidität, Finanzkraft und Finanzverhalten) und Erfolgsanalyse (Umsatzerfolg, Ertragskraft, Kapitalrentabilität) zusammen.

Kennzahlen der Strukturanalyse sind beispielsweise die Fremd- und Eigenkapitalquote einer Unternehmung. Aus der Herkunft der Mittel zur Finanzierung lassen sich Rückschlüsse auf das Ergebnisrisiko eines Unternehmens ziehen. Je höher der Anteil des Fremdkapitals am Gesamtkapital, desto höher der relative Fixkostenanteil (Zinszahlungen) und desto größer das Ergebnisrisiko.

Kennzahlen der Finanzanalyse sind u.a. der Liquiditätsgrad und der Cash-Flow (Kassenzufluß). Die erste gibt an, wieviel Geld oder geldähnliche Mittel der Unternehmung zur Deckung von kurzfristigen Verbindlichkeiten zur Verfügung stehen. Der Cash-Flow bezeichnet die Fähigkeit einer Unternehmung zur Innenfinanzierung, d.h. er gibt an, in welchem Umfang Investitionen, Gewinnausschüttungen und Schuldentilgungen in der abgelaufenen Rechnungsperiode aus eigener Kraft finanziert werden konnten. Die Verwendung dieser Kennzahlen zur Analyse einzelner Aktien muß jedoch vor dem Hintergrund erfolgen, daß unterschiedliche Definitionen [GS91] dieser Kennzahlen existieren.

Kennzahlen der Erfolgsanalyse sind z.B. die Eigenkapitalrentabilität oder Umsatzrendite. In der Eigenkapitalrentabilität kommt die Verzinsung des Eigenkapitals durch seinen Einsatz im Unternehmen zum Ausdruck. Die Eigenkapitalrendite vor Steuern sollte deutlich über den marktüblichen Zins für längerfristige Kapitalanlagen liegen, da im Unternehmergewinn zusätzlich eine Vergütung für die eingegangenen Risiken enthalten sein sollte. Die Umsatzrendite gibt das Verhältnis Gewinn zu Umsatz an und eignet sich für Vergleiche zwischen Unternehmen derselben Branche.

Entscheidungen über den Kauf oder Verkauf einer Aktie beruhen häufig auf kursbezogene Kennzahlen. Die Dividendenrendite mißt die effektive Verzinsung der Aktie auf der Grundlage des jeweiligen Kurses. Je höher die Dividendenrendite ist, desto rentabler ist die Aktie. Das Kurs-Gewinn-Verhältnis (KGV) ist, wie der Name schon sagt, das Verhältnis Aktienkurs zu Gewinn pro Aktie. Das KGV einer Aktie kann als Preismaßstab angesehen werden: je niedriger das KGV einer Unternehmung im Vergleich zum KGV anderer Unternehmen innerhalb derselben Branche ist, desto preiswerter ist die Aktie. Problematisch ist allerdings die Ermittlung des Gewinns. Zumeist wird auf der Grundlage, daß die Summe aller abgezinster Dividendenzahlungen den Wert der Aktie ergibt, der zukünftige Gewinn der Aktie ermittelt. Dieser muß aber nicht notwendigerweise mit dem tatsächlich im Produktionsprozeß erwirtschafteten Gewinn übereinstimmen.

4.3 Die technische Aktienanalyse

Die technische Aktienanalyse (TA) betrachtet den Aktienkurs als Ergebnis des Zusammenwirkens von Angebot und Nachfrage, also des Marktprozesses, und unterstellt, daß sich aus den Kursbewegungen der Vergangenheit die zukünftige Kursentwicklung ersehen läßt. Die TA bildet keine geschlossene Theorie sondern basiert auf folgenden Axiomen:

- 1 alle kursbeeinflussenden Faktoren sind im Kurs bereits enthalten,
- 2 Aktienkurse verlaufen in Trends,
- 3 die Marktteilnehmer verhalten sich als Masse zumeist irrational,
- 4 die Vergangenheit wiederholt sich.

Die TA hat im wesentlichen zwei Aufgaben zu erfüllen: die Trendprognose und das Markt-Timing, d.h. die Beantwortung der Frage, wann der Markt in welche Richtung geht. Im Rahmen einer Zeitreihenanalyse lassen sich die Aktienkurse in die Komponenten Trend, Zyklus und Zufall aufspalten. Diese Unterscheidung des Marktverhaltens ist von entscheidender Bedeutung für die Wahl der Anlagestrategie und der technischen Analyseinstrumente. Während die traditionelle TA (Chartanalyse) versucht Trendkomponenten zu identifizieren – die Anlagestrategie wäre hier Buy-and-Hold –, versucht die moderne technische Analyse (Indikatoranalyse) sowohl Trend- als auch Zykluskomponenten zu bestimmen; hier stellt

die Umschichtung eine adäquate Anlagestrategie dar. Die zufälligen Komponenten lassen sich nicht mit den Instrumenten der TA vorhersagen, sie sind meist das Ergebnis von Gerüchten und Spekulationen. Als entsprechende Anlagestrategie kann in diesem Fall die Diversifikation (Vermögensstreuung) genannt werden.

Die Dow-Theorie Die von Charles H. Dow in den Jahren zwischen 1900 und 1902 in einer Serie von Leitartikeln im Wall Street Journal, deren erster Herausgeber er war, dargestellte und später nach ihm benannte Dow-Theorie ist eine der ältesten Theorien zur Erklärung des Geschehens an den Aktienmärkten und gilt als Ursprung der TA. Die Dow-Theorie bezieht sich immer auf die Indexkurve²¹. Nach Dow läßt sich das Marktgeschehen (der Aktienindex) in drei unterschiedliche Trendarten einteilen. Es existieren kurz- und mittelfristige Kursschwankungen um einen *Primärtrend*. Die Ermittlung dieses Primärtrends ist Gegenstand der Analyse von Dow.

Die Chartanalyse Charts sind graphische Darstellungen einer Datenreihe. Die Mehrzahl der Charts ist so aufgebaut, daß auf der Abszisse der Zeitablauf dargestellt wird, während auf der Ordinate die Kurse eingezeichnet sind. Im Rahmen der Chartanalyse sind Linien-, Balken- und Point&Figure-Charts sehr beliebt.

Ein wichtiges charttechnisches Konzept sind *Widerstands-* und *Unterstützungslinien*. An früheren Höchst- und Tiefstkursen bilden sich solche Linien. Viele Marktteilnehmer, die einmal zu solchen Höchstkursen eingestiegen sind, verkaufen, wenn die Kurse das alte Niveau erreicht haben. Deshalb entsteht an solchen Punkten immer wieder erhöhtes Angebot, und die Börse benötigt mehrere Anläufe, um dieses Niveau zu überschreiten. Man spricht von der Widerstandslinie, die einem Kursaufschwung zunächst entgegensteht. Wenn die Kurse auf einen alten Tiefpunkt gesunken sind, glauben andererseits viele Börsianer, daß nun der Boden erreicht sei. Es kommt an diesen Punkten zu nachlassender Verkaufsbereitschaft bzw. zu wieder einsetzender Nachfrage, so daß sich eine Unterstützungslinie bildet, die einen weiteren Kursrückgang zumindest zeitweise verhindert. Das Konzept von Widerstand und Unterstützung basiert demnach auf psychologischen Faktoren.

Trendlinien sind eine weitere Möglichkeit, um Unterstützungs- und Widerstandsniveaus in den Charts festzustellen. Eine Trendlinie entsteht, indem man die Tief- bzw. Hochpunkte der Kursbewegungen mit einer Gerade verbindet. Eine

²¹Ein Index ist eine Kennziffer, die Veränderungen bestimmter Größen zum Ausdruck bringt und Vergleiche insbesondere von Wert- oder Preisveränderungen ermöglicht. Ein Aktienindex spiegelt den Kursverlauf eines Wirtschaftszweiges oder eines nationalen Marktes wieder. Bereits 1884 hat Charles H. Dow einen Aktienindex veröffentlicht, aus dem später die Dow-Jones-Averages wurden.

Up-Trendlinie erhält man aus der Verbindung steigender Tiefpunkte, eine *Down-Trendlinie* aus der Verbindung fallender Hochpunkte. Je länger eine Trendlinie intakt geblieben ist, und je öfter die Trendlinie auf ihren Bestand getestet wurde, umso mehr Vertrauen kann in ihrem weiteren Bestand gesetzt werden, und um so bedeutungsvoller ist ihr Durchbruch.

Lassen sich die Kursbewegungen einer Aktie durch eine Trendlinie und eine Parallele zu dieser Trendlinie einfangen, erhält man einen *Trendkanal*. Dieser definiert die Schwankungsbreite der Kurse und gibt bei mittelfristigen Bewegungen gute Anhaltspunkte für Kauf- (untere Trendlinie) und Verkaufszeitpunkte (obere Trendlinie).

Zur Interpretation der Charts werden geometrische Figuren wie Rechtecke, Dreiecke und andere graphische Figuren (u.a. Wimpel, Keile, Flaggen, Untertassen, Kopf-Schulter-Formationen) verwendet. Bei diesen *Formationen* unterscheidet man solche, die den bisherigen Trend bestätigen sollen (z.B. Wimpel), von Trendumkehrformationen (z.B. Kopf-Schulter-Formation oder Untertasse), die eine Änderung des bisher vorherrschenden Trends signalisieren.

4.4 Die moderne technische Aktienanalyse

Die moderne TA ist nichts anderes, als die statistische Auswertung von Kurszeitreihen. Sie wird auch als Indikatoranalyse bezeichnet, weil sie sich mit der Bestimmung und Bewertung von technischen Indikatoren beschäftigt. Technische Indikatoren können sowohl der Interpretation einzelner Aktien wie der Einschätzung des Gesamtmarktes dienen und lassen sich in Indikatoren der Trendanalyse und Indikatoren der Zyklusanalyse einteilen.

Die Trendanalyse Trendindikatoren haben die Aufgabe, die kurzfristigen und zufälligen Schwankungen der Aktienkurse zu glätten und den Kurstrend sichtbar zu machen. Eine Methode, um diese nicht aussagekräftigen kurzfristigen Schwankungen auszuschalten, ist die Bildung *gleitender Durchschnitte* (GD). Verwendet werden i.d.R. gleitende Durchschnitte für 200, 100 und 38 Börsentage²². Beim GD wird für jeden Börsentag ein Durchschnittskurs der letzten 200 (oder entsprechend weniger) Börsentage gebildet. Am nächsten Börsentag kommt ein neuer Wert hinzu, während der älteste Wert weggelassen wird. Schneidet die Kurslinie den GD von unten nach oben bei einem horizontal verlaufenden GD, wird dies als Kaufsignal gedeutet. Sinkt dagegen der Kurs unter die Durchschnittsline, ist dies häufig ein Verkaufssignal. Wenn bei einem stark steigenden GD der Kurs den GD von oben berührt oder leicht durchbricht, kann der Kurs schon bald wieder in seinen alten Trend einschwenken.

²²200, 100 und 38 Börsentage entsprechen in etwa den Zeiträumen von einem Jahr, 6 Monaten und 2 Monaten.

Ergänzend zum GD wird auch häufig die *Advance-Dcline-Linie* (ADL), die auch Gewinner-Verlierer-Verhältnis genannt wird, zur Trendbestimmung herangezogen [Gra76]. Die ADL ist ein Frühindikator, der die quantitative Bewegung des Marktes widerspiegeln soll. Dieser Indikator ist die kumulierte Differenz zwischen der Zahl von Aktien, die an einem Börsentag im Kurs gestiegen sind und der Zahl derer, die an diesem Tag im Kurs gesunken sind. Eine steigende ADL signalisiert, daß die Zahl der Aktien mit Kursgewinnen zunimmt.

Die Zyklusanalyse Im Mittelpunkt der zyklischen Aktienanalyse steht das *Momentum*, d.h. die relative Veränderung des Aktienkurses innerhalb einer bestimmten Zeit. Der konstruierte Momentum-Oszillator geht der Kursentwicklung voran und kann somit als Frühindikator für bevorstehende Trendwenden verwendet werden (nachlassendes Momentum). Eine wichtige Eigenschaft des Momentums ist es, den Trend aus der Zeitfunktion herauszufiltern.

Ein häufig verwendeter Indikator der Zyklusanalyse ist der Relative Stärke Index (RSI) nach J. Welles Wilder [Wel78]. Er mißt das Verhältnis aus den durchschnittlichen Kursveränderungen einer Aktie an den Tagen mit steigenden Kursen zu den durchschnittlichen Kursveränderungen an den Tagen mit fallenden Kursen. Zur Berechnung des RSI werden i.d.R. die Schlußkurse der letzten 14 Tage herangezogen. Nach einer Trendwende sollte ein RSI von über 70 % als Verkaufssignal und ein RSI von unter 30 % als Kaufsignal gedeutet werden.

Der Relative Stärke Index sollte nicht mit dem Konzept der Relativen Stärke (RS) von R.A. Levy verwechselt werden [Lev68]. Die RS mißt die Beziehung zweier Kursreihen. Üblicherweise wird das Konzept der RS dazu verwendet, die Performance einer Aktie mit der Performance einer Branche oder eines Aktienmarktindex zu vergleichen. Dazu wird der Aktienkurs durch den Aktienindex dividiert und das Ergebnis als fortlaufende Linie im Chart eingezeichnet. Auf diese Linie lassen sich dann technische Indikatoren wie GD, RSI oder Trendlinien anwenden. Das Ziel ist es, Aktien zu finden, die stärkere Kursgewinne erzielen als der Durchschnitt (Index) [WT81].

TA versus FA Die Ansätze der technischen und fundamentalen Aktienanalyse entspringen aus unterschiedlichen Überzeugungen. Die FA glaubt, den Wert einer Aktie zu kennen, beschäftigt sich aber nicht mit deren Preis. Die TA kennt den Preis, beschäftigt sich aber nicht mit dem Wert einer Aktie. Es existiert aber keine Mischform oder Kombination beider Analysen²³ und die Anhänger der jeweiligen Methode beschränken sich allein auf diese. Der Vorteil der TA ist die Aktualität und Verfügbarkeit des Datenmaterials und der daraus resultierenden Geschwindigkeit bei der Durchführung solcher Analysen. Demgegenüber steht die Interpretierbarkeit der Ergebnisse der FA.

²³Mit KNN ist allerdings eine Verknüpfung von TA und FA zur Aktienkursprognose möglich.

4.5 Intermarket Analyse

Durch die Fortschritte in der Telekommunikations- und Computerindustrie, die Liberalisierung der Kapitalmärkte (Einführung von freien Wechselkursen, weitgehende Abschaffung der Kapitalverkehrskontrollen) und die Entstehung derivativer Instrumente (Optionen, Futures)²⁴ sind die internationalen Finanzmärkte weit mehr miteinander verzahnt als noch vor 10 Jahren. Die zunehmende Globalisierung hat die Verhaltensweisen und die wechselseitigen Einflüsse der Finanzmärkte zwar grundlegend verändert, aber weder in der FA noch in der TA hat diese Tatsache bislang eine ausreichende Würdigung gefunden.

Ausgangspunkt in der Intermarket Analyse ist die Einsicht, daß sich ökonomische Größen ebenso in Trends bewegen wie Aktienkurse. Somit kann die Intermarket Analysis als Anwendung der TA auf die ökonomischen Größen, welche die Wechselwirkungen zwischen den einzelnen Finanzmärkten bewirken, verstanden werden. Der Finanzmärkte werden in vier Sektoren unterteilt [Mur92]: Aktien, Renten, Devisen und Rohstoffe. Die Basis bildet meistens die Analyse der Wechselkursbewegungen des US-Dollars zu anderen Währungen. Diese Schwankungen beeinflussen die Rohstoffpreise, insbesondere den Rohölpreis, was zu einer Inflationsänderung und damit zu einer Zinsänderung führt. Zwischen den Zinsen, festverzinslichen Wertpapieren und den Aktienkursen gibt es weitere Abhängigkeiten (siehe Kap. 4.2). Eine wichtige Bedeutung kommt auch dem Goldpreis zu. Diese Größe bewegt sich langfristig entgegen der Entwicklung des US-Dollars. Die US-amerikanische Notenbank richtet ihre Geldmarktpolitik u.a. an der Entwicklung des Goldpreises aus. Zwar sind alle diese ökonomischen Größen miteinander verflochten, in praktischen Anwendungen werden wegen des hohen Analyseaufwands jedoch nur die Wechselwirkungen zwischen wenigen (meistens nur zwei) Größen betrachtet.

²⁴Neben dem Kassahandel der Börse, bei dem die Abschlüsse sofort oder ganz kurzfristig erfüllt werden müssen, existieren die sogenannten Termingeschäfte. Finanztermingeschäfte (engl. financial futures) beschreiben den Kauf bzw. Verkauf einer dem Finanzmarkt (Geld- und Kapitalmarkt) zugeordneten, hinsichtlich Qualität und Quantität standardisierten Basisgröße zu einem bestimmten Preis bzw. Kurs zu einem bestimmten Zeitpunkt. Als Basisgröße kommen Aktien, Anleihen, Währungen, Rohstoffe (engl. commodities) und Indizes in Frage. Die wichtigsten Börsen für Termingeschäfte befinden sich in London, Chicago und New York. Das Optionsgeschäft ist ein bedingtes Termingeschäft in Wertpapieren, das dem Käufer von Optionscheinen (engl. warrants) die Wahlmöglichkeit gibt, innerhalb einer bestimmten Frist (Optionslaufzeit) oder zu einem bestimmten zukünftigen Zeitpunkt zu einem vorab vereinbarten Kurs (Basiskurs oder -preis) gegen sofortige Zahlung einer Optionsprämie (Optionspreis) zu kaufen (Kaufoption, put) oder zu verkaufen (Verkaufsoption, call). I.d.R. sind Käufer von Optionscheinen weniger daran interessiert, ihr Optionsrecht auszuüben, vielmehr kommt es ihnen darauf an, den Optionsschein noch während seiner Laufzeit zu verkaufen. Während der Käufer von Call-Optionsscheinen steigende Kurse erwartet, hofft der Käufer von Put-Optionsscheinen auf sinkende Kurse. Der Gewinn oder Verlust ergibt sich aus der Differenz zwischen dem vereinbarten Basiskurs und dem aktuellen Marktwert.

4.6 Nichtlineare dynamische Theorie

Die traditionellen Techniken und Modelle der Aktienanalyse basieren auf der Annahme, daß es sich bei dem Aktienmarkt um ein lineares System handelt. In den letzten Jahren kamen zahlreiche Untersuchungen zu dem Ergebnis, daß Kapitalmärkte und Volkswirtschaften von nichtlinearen dynamischen Prozessen bestimmt werden. Im Gegensatz zu den linearen Systemen gibt es bei den nichtlinearen Systemen als Reaktion auf die Veränderung einer Variablen nicht nur eine, sondern eine ganze Reihe von denkbaren Veränderungen des Gesamtsystems. In einem nichtlinearen System können relativ kleine Veränderungen in der einen Variablen zu überproportional hohen Veränderungen in den anderen Variablen und dem Verhalten des Gesamtmarktes bzw. Aktienindex führen. Die zunehmende Globalisierung der Finanzmärkte ist wohl das wichtigste Argument gegen die Charakterisierung des Aktienmarktes als ein lineares System.

Die Chaostheorie Der Aktienmarkt ist ein deterministisches System, da zwischen den kursbestimmenden Variablen und dem Ergebnis – dem Aktienkurs bzw. Aktienindex – eine kausale Beziehung besteht. Mit anderen Worten heißt dies, daß der Aktienmarkt kein zufälliges System ist (vgl. Kap. 4.1), weil Ursache-Wirkungs-Verknüpfungen existieren. Ein deterministisches System, in dem nichtlineare dynamische Prozesse ablaufen, läßt Ergebnisse innerhalb einer bestimmten Streuung zu. Innerhalb dieser Streuung erscheint das Verhalten des Systems chaotisch.

Die Grundaussage der Chaostheorie besagt, daß dynamische Systeme einen quasi-stabilen Zustand annehmen können, d.h. geringfügige Störungen lösen dann keine Veränderungen des Systemzustandes aus. Nur wenn gleichzeitig mehrere Störungen auftreten, verliert das System seine Fähigkeit, sich selbst zu stabilisieren. Ein Beispiel für das chaotische Verhalten von Kapitalmärkten ist der Kurssturz an den internationalen Aktienmärkten im Oktober 1987. Eine wichtige Eigenschaft eines nichtlinearen dynamischen Systems ist, daß kleine Fehler in den Anfangsbedingungen im Zeitablauf exponentiell wachsen und damit langfristige Prognosen bedeutungslos machen.

Fraktale Die fraktale Geometrie geht auf den Mathematiker Benoit Mandelbrot zurück [Man77]. Er entdeckte, daß sich zahlreiche mathematische Probleme auf die graphische Darstellung einer mehr oder weniger gewellten Linie zurückführen lassen, deren Dimension ein Bruchteil (Fraktal) einer ganzen Zahl zwischen 1 und 2 ist, und daß dieses Fraktal für viele chaotische Systeme konstant ist. Übertragen auf die traditionelle TA bedeutet dies, daß Trendlinien nun nicht mehr als Geraden (Dimension 1) sondern als gekrümmte Kurven in den Charts dargestellt werden müssen.

Die Spektralanalyse Als Spektrum wird die Gesamtheit der zyklischen Bewegungen einer Zeitreihe bezeichnet. Die Spektralanalyse ist ein Verfahren zur Berechnung dieser zyklischen Schwankungen. Selbst sich überlagernde Schwingungskomponenten lassen sich mit Hilfe der Spektralanalyse identifizieren. Die Schwingungskomponenten geben ihrerseits Aufschluß über die Bedeutung und Wechselwirkung einzelner Größen für die Zeitreihe.

4.7 Aktienanalysen mit Hilfe von KNN

KNN können im Rahmen der Aktienanalyse als statistische Verfahren der nicht-linearen Regressions- und Zeitreihenanalyse charakterisiert werden. Ausgehend von einer Blackbox werden an die Eingabeschicht des Netzes Daten angelegt, über die innere, während des Lernprozesses adaptierte Gewichtung verknüpft und als Prognoseinformation an der Ausgabeschicht zur Verfügung gestellt.

Hinsichtlich der Eingabedaten werden drei wesentliche Vorgehensweisen unterschieden: Berücksichtigung der eigenen vergangenen Zeitreihe sowie die von diversen technischen Indikatoren (TA) oder Benutzung von Unternehmensdaten, branchenspezifische und gesamtwirtschaftliche Faktoren in Form von Zeitreihen (FA). Außerdem ist eine Kombination aus beiden Möglichkeiten denkbar.

Bei der Netzauswahl muß prinzipiell zwischen vorwärtsgekoppelten und rekurrenten Netzen unterschieden werden. In vorwärtsgekoppelten Netzen wird der Faktor Zeit entweder direkt durch die Bildung von zeitverzögerten Datenreihen oder indirekt mittels Durchschnittsbildungen, absoluter oder relativer Abweichungen zur Vorperiode modelliert. Rekurrente Netze sind in der Lage die relative zeitliche Position eines Eingabemusters von seiner absoluten zeitlichen Position zu unterscheiden. Die Rückkopplungen erlauben dem Netz, bedingt durch den zyklischen Informationsfluß, vorangegangene Systemzustände in das Netzwerkverhalten dynamisch zu integrieren. Dies könnte der Kapitalmarktdynamik eher angemessen sein. Ein weiterer Vorteil rekurrenter Netze ist die sparsame Dimensionierung der Neuronen. Das Problem dieser Netze ist es aber, einen geeigneten Lernalgorithmus zu finden. Aus diesem Grund beschränken sich die meisten Anwendungen auf vorwärtsgekoppelte KNN.

Im folgenden werden nun anhand von einigen Beispielen die Aktienkurs- bzw. Aktienindexprognose mittels KNN kurz vorgestellt.

Modell von Schöneburg In den Jahren 1989/90 veröffentlichte E. Schöneburg erste Aufsätze bzw. Artikel zum Problembereich der Vorhersage von Aktienkursen [SGR89, Sch90, SHG90]. Sie beschäftigen sich primär mit den Einsatzmöglichkeiten von Adaline-, Madaline- und Backpropagation-Netzwerken zur tageweisen Kursprognose. Es standen ausschließlich die Tageskurse der Aktiengesellschaften BASF, Mercedes und Commerzbank für Trainingszwecke über den

Zeitraum vom 9.2.1989 bis 18.4.1989 und zum anschließenden Test vom 19.4.1989 bis 19.5.1989 (19 Tage) zu Verfügung. Die Prognosen mit Hilfe der Adaline- und Madaline-Netzwerke verliefen zufriedenstellend, wobei allerdings die Trefferquote der Mercedes-Aktie nicht überzeugen konnte. Sie lag i.d.R. nur geringfügig über der Signifikanzschwelle von 50 %. Deutlich besser verhielten sich die Resultate für die BASF- und Commerzbank-Aktie, sie lagen beide bei ca. 63 %. Begründet wird dieses nahezu gleiche Resultat damit, daß die Kursverläufe der beiden Aktien im Untersuchungs- und Testzeitraum ähnlich waren, gemessen an der maximalen Kursänderung pro Tag und der Differenz zwischen Minimal- und Maximalkurs. Die Untersuchungen von Schöneburg zum Backpropagation-Netzwerk sind durch Alternativenbetrachtungen gekennzeichnet und umfassen sowohl die Topologie als auch die Netzwerkparameter Lernrate η und Momentum α . Nach einer Reihe von aufgeführten Varianten konstatierte Schöneburg, daß das beste Resultat eines Backpropagation-Netzwerkes für die Mercedes-Aktie erzielt wurde und eine Trefferquote von 64 % entspricht. Hervorzuheben ist, daß mit dem Backpropagation-Verfahren prinzipiell nicht nur eine Steigt/Fällt-Prognose, sondern auch die explizite Vorhersage von Kurswerten möglich ist.

Modell von Kimoto, Asakawa, Yoda und Takeoda Das Modell von Kimoto, Asakawa, Yoda und Takeoda [K^{ea}90] wurde entwickelt, um die zeitliche Entwicklung des Tokyo Stock Exchange Price Index (TOPIX) zu prognostizieren. Hinter den Autoren stehen der japanische Fujitsu-Konzern und die Investment Technologie & Research Division der Nikko Securities Corporation. Gemeinsam wurde das Aktienhandelssystem SMPS (Stock Market Prediction System) entwickelt, ein vollautomatisiertes Kauf- und Verkauf-Vorhersagesystem, das aus mehreren kooperierenden Netzwerken besteht. Es wurde mit dem Ziel trainiert, die Zusammenhänge zwischen verschiedenen technischen und ökonomischen Indizes in Verbindung mit optimalen Kauf- und Verkaufszeitpunkten für ein von ihnen verwaltetes Portfolio spezifischer Aktien anhand der Entwicklung des TOPIX zu erkennen. Die Netze wurden nach einem modifizierten Backpropagation-Algorithmus, der sogenannten Supplementary-Learning-Methode, trainiert. Als Eingabedaten dienten u.a. die Dow-Jones-Averages, Zinssätze und das Umsatzvolumen der gehandelten Aktien. Die Güte des Systems wurde über einen Zeitraum von insgesamt 33 Monaten zwischen Januar 1987 und September 1989 getestet. Vier KNN wurden unabhängig voneinander trainiert und haben unabhängig voneinander Vorhersagen erstellt über die ein Mittelwert gebildet wurde. Das zur Anlage zu Verfügung stehende Kapital konnte im Untersuchungszeitraum mit einer einfachen Buy-and-Hold-Strategie um 67 % gesteigert werden, während das SMPS eine Steigerung von 98 % erzielte. Ergänzend dazu hält M. Yoda fest, daß das System seit mehr als drei Jahren im Einsatz ist und 62,1 % der 145 erstellten Steigt- bzw. Fällt-Prognosen korrekt erkannt hat [Yod94].

Modell von Rehkugler und Podding H. Rehkugler und T. Podding haben seit 1990 eine Reihe von Beiträgen zur Aktienkursprognose mit Hilfe KNN beigesteuert [RP90b, Reh91, RP91, RK95, Reh95a]. In einem dieser Modelle [RP90b] untersuchten sie die Entwicklung des Aktienindex des statistischen Bundesamtes anhand zweier KNN: dem Mehrschichten-Perzeptron (MLP) und der Boltzmann-Maschine. Sie gingen von einem fiktiven Startkapital und Wiederanlage von Gewinnen unter Berücksichtigung von Transaktionskosten (1,35 % des Kurswertes) aus. Als Referenzmaßstab diente eine Buy-and-Hold-Strategie. Der ökonomische Erfolg des Modells wurde durch den Mehrgewinn ausgedrückt, d.h. der prozentuale Unterschied des Gewinns beider Anlagestrategien. Die Eingabedaten des Modells sind die Umlaufrendite der Anleihen der öffentlichen Hand, die Relation Geldvolumen M1 zu nominalen BSP und das Geschäftsklima des Investitionsgütergewerbes. Es wurden fünf Ausgabeneuronen benutzt, vier bildeten Bereiche für prozentuale Indexveränderungen ab, das fünfte wurde als Steigt/Fällt-Indikator genutzt. Die Ergebnisse wurden außerdem mit einem statistischen Modell (multivariate Regressionsanalyse) verglichen. Während das statistische Modell eine Trefferquote von ca. 63 % und einen Mehrgewinn in Höhe von ca. 43,3 % erzielte, hatte das MLP eine Trefferquote von ca. 72 % und einen Mehrgewinn von 93 % zu verbuchen. Die Resultate der Boltzmann-Maschine liegen allerdings (nur) im Bereich des statistischen Modells.

Modell von Siemens-Nixdorf Informationssysteme AG Die Firma Siemens-Nixdorf Informationssysteme (SNI) AG hat unter der Mitarbeit von H.-G. Zimmermann, H.-O. Isbert, M. Miller, S. Baun und A. Storm eine Software-Entwicklungsumgebung für Neuronale Netze (SENN) entwickelt, die im Laufe dieser Entwicklung und durch größere Kooperationsprojekte immer mehr auf den Anwendungsbereich der Finanzanalyse und Finanzprognose zugeschnitten wurde. Es wurde ein Modell entwickelt, um die Ein-Tages-Rendite des Deutschen Aktienindex (DAX) zu prognostizieren [BS92, Bau94]. Als KNN diente das MLP mit einer verborgenen Schicht. Die Ausgabe ist die Rendite des nächsten Tages. Als Rohdaten liegen ihren Untersuchungen anfänglich 34 technische und fundamentale Variablen zugrunde, deren Zeitreihen ein Überlappungsbereich von 1984 bis 1993, also ca. 10 Jahre, aufweisen. Sie werden insgesamt 7 Eingabe-Clustern zugeordnet:

- 1 Zeitvariablen, DAX und angelehnte Variablen
- 2 Kapital- und Geldmarktzinsen Inland
- 3 Geldmarkt monetäre Aggregate Inland
- 4 Ausländische Aktienindizes
- 5 Währungen und Gold
- 6 Kapital- und Geldmarktzinsen Ausland
- 7 Güter- und Arbeitsmarkt

Je nach Anforderung wurden diese Rohdaten aufbereitet (skaliert, logarithmiert oder differenziert). Im Rahmen von Voruntersuchungen wurden anschließend die transformierten Zeitreihen paarweise mittels Korrelationsanalysen auf Redundanzen untersucht. Sieben der ursprünglich 34 Einflußfaktoren konnten auf diese Weise eliminiert werden. Anschließend erfolgte die Untersuchung von Time-lag-Strukturen zwischen den verbliebenen Eingabezeitreihen und der DAX-Zeitreihe. Insgesamt ergaben sich so 190 transformierte Zeitreihen. Während der Lernphase wurde versucht, das Netzwerk sukzessive zu optimieren. Es kamen die Techniken Weight Pruning gemäß statistischer Signifikanz, Input-Pruning und Hidden-Merging zum Einsatz. Die Anzahl der Gewichte wurde dabei von 1520 auf 195, die Eingabeneuronen von 190 auf 90, die inneren Neuronen von 8 auf 3 und die Ausgabeneuronen von 27 auf 23 reduziert. Darüber hinaus wurde anhand von Sensitivitätsanalysen festgestellt, daß die aus dem inländischen Aktienmarkt selbst stammenden Zeitreihen, die ausländischen Aktienindizes, das inländische kurz- und langfristige Zinsniveau und das ausländische langfristige Zinsniveau den DAX am stärksten in seiner Entwicklung beeinflussen. Über den Testzeitraum vom 5.1.1993 bis 1.12.1993 (237 Tage) erbrachte das Modell eine Rendite von 41 %, eine einfache Buy-and-Hold-Strategie kam auf 34 %.

Modell der SGZ-Bank AG J. Graf, Mitarbeiter der Südwestdeutsche Genossenschafts-Zentralbank AG (SGZ-Bank AG), stellte 1991 ein System zur Vorhersage des DAX vor [Gra91]. Dieses Netzwerk sollte auf Monatsfrist optimale Kauf- und Verkaufszeitpunkte anhand von technischen und ökonomischen Einflußfaktoren vorhersagen. Die bei diesem neuronalen System zum tragen kommende Netzwerkarchitektur bestand aus vier Ebenen: einer Eingabe-, zwei verborgene und eine Ausgabeschicht. Zwischen je zwei Schichten sind die Neuronen vollständig miteinander vernetzt. Die dem Netzwerk angebotenen Trainingsdaten entstammen dem Zeitraum zwischen dem 1.1.1988 und 8.10.1990, die nachfolgenden 40 Wochen wurden zum Test herangezogen. Die Eingabedaten wurden in 5 Klassen unterteilt: technische Indikatoren, ausländische Aktienindizes, Wechselkurse, Zinssätze und Umsatzvolumen. Während eine Buy-and-Hold-Strategie im Testzeitraum einen Profit von 11,26 % erzielte, übertraf das KNN diesen Wert und erreichte 17,70 %.

Modell von Schuhmann und Lohrbach M. Schuhmann und T. Lohrbach versuchen mit einem technisch und einem fundamental orientierten Modell sowie einer Kombination daraus kurzfristige Steigt/Fällt-Prognosen von Aktienkursentwicklungen vorzunehmen [SL93,SL94]. Die Vorhersage bezog sich auf die Entwicklung des Kurses am nächsten Börsentage bzw. die Tendenz der nächsten fünf oder zehn Börsentage. Als Datenmaterial standen die täglichen Notierungen des DAX und der Hochtief-Aktie für den Zeitraum 31.12.1982 bis zum 31.12.1991 zur Verfügung. Außerdem wurden weitere Zeitreihen benutzt, die den Aufbau

eines fundamentalen Modells ermöglichen: Anzahl aller gehandelter Aktien, Anzahlen der gestiegenen, gefallen und unveränderten Aktien, Dow-Jones-Index, Nikkei-Index, Financial-Times-Index, Westbau-Index, Wechselkurse (US-Dollar, Englisches Pfund, Französischer Franc, Schweizer Franken und europäischer ECU), Geldmarkttagesätze (Tages-, Monats- und Dreimonatsgeld), Goldpreis und Rohölpreis. Aus diesen Zeitreihen wurden zusätzliche Indikatoren als technische Eingabeinformationen gebildet: der Trendoszillator (Verhältnis zweier gleitender Durchschnitte), die relative Stärke, das Momentum, der Relative-Stärke-Indikator und der Überkauft/Überverkauft-Indikator (Differenz des aktuellen Kurses zum tiefsten Kurs im Verhältnis zu der Differenz des höchsten Kurses zum tiefsten Kurs einer Periode). Es wurde ein Counterpropagation-Netzwerk und die Software Nwork von Neural Ware benutzt. Innerhalb der Gossberg-Schicht finden sich bei ihnen 4 Neuronen, die angeben, ob der Kurs am nächsten Tag steigt oder fällt und ob diese Änderung signifikant ist, d.h. nach Schuhmann und Lohrbach mehr als 0,5 % beträgt. Innerhalb der Kohonen-Schicht des Netzwerks wurde zunächst die Anzahl der Neuronen gleich der Anzahl der Lerndatensätze gesetzt, da jedes Neuron der Kohonen-Schicht eine Klasse repräsentieren kann. Die Anzahl der Neuronen wurde dann in nachfolgenden Schritten sukzessive halbiert, um einem Auswendiglernen des Netzes entgegenzuwirken. Die Trainingsphasen der Lernalgorithmen wurden abgebrochen, wenn eine vorgegebene Toleranzgrenze für den Netzwerkfehler unterschritten wurde. Mit Hilfe von Kreuz-Korrelations-Untersuchungen bestimmten Schuhmann und Lohrbach die endgültige Zusammensetzung der Eingabeinformationen. Die besten Ergebnisse wurden mit den technisch orientierten Eingabedaten erzielt, die Trefferquote lag über 60 %. Als schlechter erwiesen sich die Ergebnisse, die mit Hilfe der fundamental orientierten Eingabedaten oder der Kombination aus technischen und fundamentalen erzielt wurden.

Modell von Malliaris M.E. Malliaris veröffentlichte 1994 ein neuronales Modell zur kurzfristigen Vorhersage des Standard-and-Poor's 500-Index (S&P 500) [Mal94]. Motivation war der Börsencrash im Oktober 1987, der ihrer Meinung nach alles andere als zufällig war. Für die Untersuchungen, in denen die Frage, ob Kursentwicklungen eher durch Zufall (Random-Walk-Hypothese) oder chaotisch-dynamisch Strukturen zustanden kommen (vgl. Kap. 4.1 und 4.6), eine wichtige Rolle spielte, wurde ein Backpropagation-Netzwerk benutzt. Zum Training des Netzes dienten Wochenschlußnotierungen in den Jahren 1989 und 1990: S&P 500 Schlußindex, 3-Month Treasury Bill Interest Rate, 30-Year Treasury Bond Interest Rate (öffentliche Anleihen mit einer Laufzeit von 3 Monaten bzw. 30 Jahren), wöchentliches Umsatzvolumen der New Yorker Börse, Geldmengen M1 und M2, Preis/Lohn-Verhältnis, Goldpreis, Rohölpreis, CBOE put/call-Verhältnis²⁵. Mit

²⁵Das Verhältnis des Umsatzes an put-Optionen dividiert durch den Umsatz bei call-Optionen und der Terminbörse von Chicago (Chicago Board Options Exchange).

Hilfe dieser zehn Faktoren sollten in dem hier diskutierten Ansatz die Schlußnotierung des S&P 500-Index für den nächsten Freitag abgeleitet werden. In das Netz flossen hierzu als Eingabe neben der aktuellen Ausprägung der Einflußfaktoren auch zwei zu früheren Zeitpunkte festgehaltenen Notierungen ein. Mit Hilfe eines Optimierungsverfahrens auf Basis Genetischer Algorithmen wurde die Anzahl der verborgenen Schichten und deren Neuronen bestimmt. Das fertig trainierte Netzwerk bestand dann aus zwei verborgenen Schichten mit 24 bzw. 8 Neuronen. Zur Ergebnisbeurteilung, insbesondere zum Vergleich mit der Random-Walk-Hypothese, wurde die mittlere Standardabweichung, der mittlere quadratische Fehler und als weiteres Kriterium die Korrelation zwischen prognostizierter und historischer Ausprägung des S&P 500-Index berechnet. In jeder der zweijährigen Testphase schlug das KNN das Random-Walk-Modell.

Modell von Ruggiero M.A. Ruggiero baute basierend auf vermuteten marktübergreifenden Interdependenzen ein KNN zur Vorhersage des S&P 500-Index über ein Zeitraum von fünf Wochen auf [Rug94]. Als Eingabedaten wurden der 30-Year Treasury Bond Index und der Commodity Research Bureau Index (Preisindex für Rohstoffe) neben der eigenen Historie der S&P 500-Indexentwicklung herangezogen. Als Datenmaterial zum Training standen ihm die genannten Zeitreihen (Wochenschlußnotierungen) zwischen dem 2.1.1988 und 20.5.1994 zur Verfügung. Aufgeteilt wurde diese Datenbank in eine Trainingsdatenbank (1.6.1988 - 31.12.1992), eine Validierungsmenge (1.1.1993 - 28.2.1994) und Testdaten (1.3.1994 - 20.5.1994). Zum Training wurde ein dem Backpropagation-Algorithmus verwandtes Lernverfahren (Back-Percolation-Algorithmus) benutzt. Die Rohdaten wurden vorverarbeitet und 63 potentielle Zeitreihen abgeleitet, die allerdings nach einer Optimierung auf 18 reduziert wurden. Das Netzwerk bestand weiterhin aus 5 verborgenen Neuronen und einem Ausgabeneuron. Die Ergebnisse der besten Netzwerke lagen zwischen 68,3 % und 71,6 % Trefferquote.

5 Prognose des Deutschen Aktienindex

Deutscher Aktienindex Der Deutsche Aktienindex (DAX) (siehe Abb. 5.1) wurde am 1. Juli 1988 eingeführt. Der Basiswert von 1000 am 31.12.1987 wurde rückberechnet. Dieser Aktienindex basiert auf der Kursentwicklung von 30 an der Frankfurter Wertpapierbörse (FWB) notierten Aktienwerten mit hoher Umsatzaktivität, großer Börsenkapitalisierung und frühen Eröffnungskursen. Er wird während der amtlichen Handelszeit (10:30 bis 13:30 Uhr MEZ) jede Minute neu berechnet. Die Gewichtung der im Index enthaltenen Gesellschaften erfolgt nach der Höhe ihres Grundkapitals. Aufgrund von Ausgaben von Gratisaktien, Barkapitalerhöhungen, Nennwertumstellungen und Bardividendenzahlungen muß Indexpflege betrieben werden. Der Kurs der einzelnen AG wird mit einem Korrekturfaktor bereinigt. Der Austausch von Gesellschaften im DAX wird mittels eines sogenannten Verkettungsfaktors berücksichtigt.

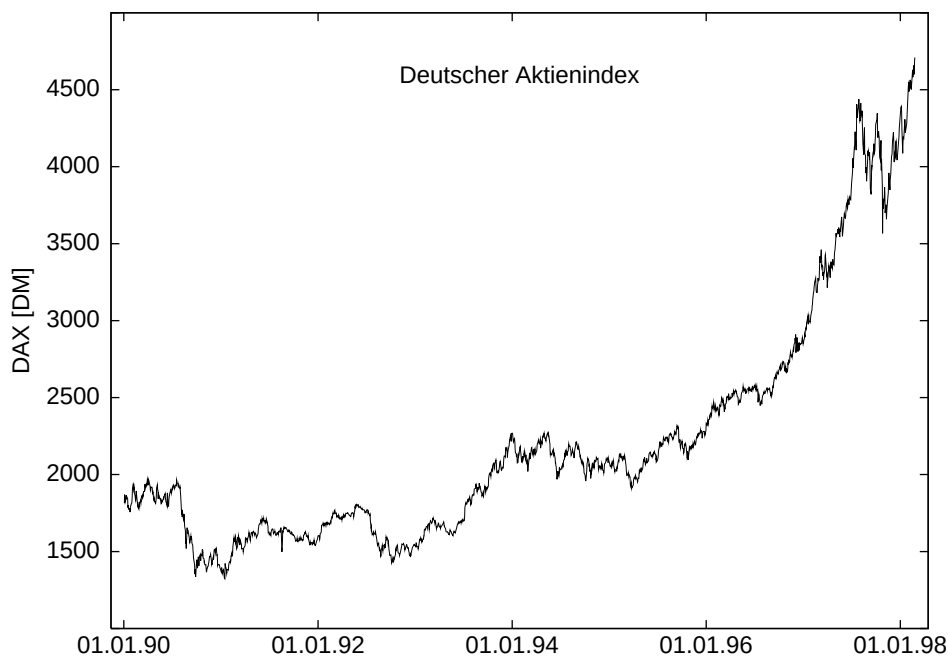


Abbildung 5.1: Zeitlicher Verlauf des Deutschen Aktienindex

5.1 Ziel der DAX-Prognose

Der DAX soll vom 1.10.1997 bis zum 28.2.1998, also über einem Zeitraum von 5 Monaten, vorhergesagt werden. Die Prognose mit Hilfe KNN beschränkt sich auf die DAX-Vorhersage des nächsten Tages. Das Zeitfenster des Prognosezeitraumes wird während der Simulation sukzessive um einen Tag verschoben. In

Abb. 5.1 erkennt man, daß es Ende 1997, genauer am 28. Oktober, zu heftigen Kurseinbrüchen an der FWB kam. Der DAX hat etwa 8 % zum Vortag verloren. Ursache war ein Crash am 23. Oktober an der Börse in Hongkong. Ziel der DAX-Prognose ist somit einerseits die Vorhersage der Wirkung des Asien-Crashes auf den deutschen Aktienmarkt und andererseits der Test dieser Prognosen auf Anlageentscheidungen (Markt-Timing).

Der Asien-Crash 1997 Im Herbst 1997 haben sogenannte Tigerstaaten wie Thailand und Malaysia nach und nach ihre Bindung an die US-amerikanische Währung aufgeben müssen, weil international finanzkräftige Devisenspekulanten in diese Währungen investierten. Dies führte zu einer starken Kapitalflucht, einbrechenden Aktienkursen (um 30 % bis 50 %) und zu kräftigen Abwertungen ihrer nun frei schwebenden Landeswährung, während die Devisenspekulanten Milliarden Gewinne erzielten. Der Hongkong-Dollar ist allerdings seit 14 Jahren an den US-Dollar gekoppelt und die Hongkonger Währungsbehörde (HKMA) verfügt über Devisenreserven von 80 Mrd. Dollar. Damit ist die Position Hongkongs wesentlich stärker als die Thailands oder Malaysias. Doch um den Kurs des Hongkong-Dollars zu stabilisieren, sah sich die HKMA gezwungen, zusätzlich zu milliardenschweren Investitionen am Devisenmarkt auch die kurzfristigen Zinssätze radikal hinaufzusetzen. Die Zinssätze für Tagesgelder stiegen daraufhin von 6 % auf zunächst 150 %, um dann am 23. Oktober sogar auf 250 % bis 300 % zu klettern. Da 80 % des Einkommens in Hongkong auf Dienstleistungen entfällt, und hier wiederum hauptsächlich auf die besonders zinsempfindlichen Branchen Immobilien und Banken, kam es zu regelrechten Panikverkäufen an der Börse. Nach Tagesverlusten von jeweils 4,5 % am Montag und Dienstag sowie einem weiteren Absturz um 6,2 % am Mittwoch, folgte schließlich am Donnerstag (23.10.) der nach 1987 größte Crash in der Börsengeschichte Hongkongs mit einem Rückgang von 10,4 %, wobei der Hang Seng Index zeitweise sogar 16 % im Minus stand. Aus ganz Ostasien zogen die großen Investmentfonds ihre Gelder ab, was zu Kurseinbrüchen an allen asiatischen Börsen führte. Schließlich kam es auch zu Unsicherheiten an den Börsen in Nordamerika und Europa, so daß dort ebenfalls die Kurse einbrachen.

5.2 Eingabedaten

Grundlage der DAX-Prognose dieser Arbeit bildet die Intermarket-Analyse (vgl. Kap. 4.5). Die Eingabedaten liegen auf Tagesbasis vom 1.1.1990 bis zum 28.2.1998 vor und werden in vier Sektoren unterteilt: Aktienindizes, Devisen, Renten und Rohstoffe. Hinsichtlich der Weltmärkte (Nordamerika, Asien/Pazifik und Europa) lassen sich diese Sektoren weiter differenzieren. Bei den Aktienindizes werden die Schlußkurse des amtlichen Präsenz-Handels an den jeweiligen Börsen und bei

den Wechselkursen die amtlichen Kassa-Kurse zum Mittags-Fixing an der Devisenbörse Frankfurt verwendet. Als Eingabedaten dienen der DAX und weitere 40 Zeitreihen.

Aktienindizes (siehe Abb. 5.2):

- 1** Dow Jones Industrial Average Index
- 2** New York Stock Exchange Composite Index
- 3** National Association of Securities Dealers Automated Quotation Composite Index
- 4** Standard & Poor's Composite 500 Index
- 5** Toronto Stock Exchange 300 Index
- 6** Australien Share Price Index (All Ordinaries)
- 7** Nikkei 225 Index
- 8** Singapore Straits Times Index
- 9** Hang Seng Index
- 10** Korea Composite Stock Price Index
- 11** Financial Times Stock Exchange 100 Index
- 12** Affärsvärlden General Index
- 13** Compagnie des Agents de Change 40 Index
- 14** Milano Italia Borsa Index

Devisen (Basis: Deutsche Mark):

- 15** US-amerikanische Dollar
- 16** Kanadische Dollar
- 17** Australische Dollar
- 18** Singapur Dollar
- 19** Hongkong Dollar
- 20** Koreanische Won
- 21** Japanische Yen
- 22** Britische Pfund
- 23** Schwedische Kronen
- 24** Französische Francs
- 25** Italienische Lire
- 26** Schweizer Franken
- 27** Spanische Peseten

Renten:

- 28** 30 Years US Treasury Bond Rate
- 29** 10 Years US Treasury Bill Rate
- 30** 3 Month US Treasury Note Rate
- 31** Umlaufrendite BRD²⁶

²⁶Rendite festverzinslicher, im Umlauf befindlicher Wertpapiere. Tägliche Feststellung mittags ca. 13:30 Uhr MEZ von der Bundesbank Frankfurt.

Rohstoffe:

- 32** American Stock Exchange Oil and Gas Index
- 33** American Stock Exchange Gold and Silver Index

Sonstiges:

- 34** Diskontsatz USA
- 35** Diskontsatz BRD
- 36** Geldmenge M3 USA
- 37** Geldmenge M3 BRD
- 38** Volumen NYSE (New York)
- 39** Volumen KSE (Seoul)
- 40** Wochentag

Index	Börse	Aktien ²⁷	Gewicht ²⁸	Seit ²⁹	Basis	Datum
Dow Jones	New York	30	Preis	01.10.28	100	(1896)
NYSE-C	New York	≈ 1000	Kapital	01.06.64	50	(31.12.65)
NASDAQ-C	USA	> 5000	Kapital	05.02.71	200	(29.10.93)
S&P 500	New York	500	Kapital	03.01.28	100	(01.04.82)
TSE-300	Toronto	300	Kapital	01.01.75	1000	(01.01.75)
All Ord.	Sydney	≈ 300	Kapital	01.01.80	500	(31.12.79)
Nikkei-225	Tokyo	225	Preis	05.01.53	176,21	(16.05.49)
STI	Singapur	30	Kapital	03.01.66	100	(30.12.66)
Hang Seng	Hongkong	33	Kapital	24.11.69	100	(31.07.64)
KOSPI	Seoul	200	Kapital	04.01.75	100	(04.01.80)
FTSE-100	London	100	Kapital	02.01.84	1000	(31.12.83)
AFGX	Stockholm	≈ 262	Kapital	03.01.86	100	(31.12.79)
CAC-40	Paris	40	Kapital	01.03.89	1000	(31.12.87)
MIB	Mailand	≈ 250	Kapital	02.01.75	1000	(02.01.75)

Abbildung 5.2: Aktienindizes

Die Volumen sind in Anzahl der täglich gehandelten Aktien angegeben und nicht deren Wert in der jeweiligen Währung. Der Wochentag ist als Zahl codiert (1=Montag, 2=Dienstag, usw.) und wird ebenfalls zur Analyse herangezogen,

²⁷Einige Aktienindizes (z.B. NYSE-C) enthalten alle inländischen Aktien, die an dieser Börse gehandelt werden. Da sich diese Anzahl im Laufe der Zeit ändert, kann kein genauer Wert angegeben werden.

²⁸Die Aktien des Dow Jones und Nikkei-Index werden im Prinzip nicht gewichtet, sondern einfach der arithmetische Mittelwert der Kurse (Preise) gebildet. Die Gewichtung ist für alle Aktien im Index gleich und ist durch den Kehrwert der Anzahl der im Index enthaltenen Aktien gegeben.

²⁹Der Beginn täglicher Notierungen der Indizes muß nicht mit dem Beginn der Indexnotierung übereinstimmen, da diese zuvor oft nur wöchentlich festgestellt wurden.

weil viele Börsianer von einem Montag-Phänomen sprechen, d.h. zu Wochenbeginn steigen die Kurse oftmals stark an, da die Börsengeschäfte zwei Tage ruhten.

Desweiteren werden auch monetäre Daten benutzt, die allerdings nicht auf täglicher Basis vorliegen. Die Diskontsatzänderungen beziehen sich auf einzelne Tage und die Geldmenge M3 der USA liegt wöchentlich und die der BRD monatlich vor. Damit diese Daten zeitlich zu den anderen Daten passen, wird der Wert dieser Größen so lange beibehalten, bis ein neuer Wert vorliegt. Beispielsweise sind dann die Werte von M3 der USA wegen der wöchentlichen Notierung von montags bis freitags identisch.

Problematisch in der Simulation ist die zeitliche Abgrenzung eines Datums, da die drei betrachteten Weltmärkte räumlich und damit auch zeitlich durch unterschiedliche Zeitzonen voneinander getrennt sind. Als Zeitpunkt der Simulation wird 15:00 Uhr MEZ gewählt. Die asiatischen und europäischen Börsen haben dann bereits geschlossen, die nordamerikanischen Börsen noch nicht eröffnet³⁰. Somit müssen alle nordamerikanischen Daten in der Simulation um einen Tag vor verschoben werden. Denn dann gehören zu einem Eingabedatensatz die europäischen und asiatischen Werte von heute und die nordamerikanischen von gestern, wenn der DAX von morgen prognostiziert werden soll.

Insgesamt ergeben sich aus den 2981 Tagen des Untersuchungszeitraums 1772 komplette Datensätze, worauf 70 auf den Vorhersagezeitraum entfallen. Leider sind drei Lücken in den Vorhersagezeitreihen. Zwischen dem 22.12.1997 und 5.1.1998 liegen keine vollständigen Daten aufgrund der Weihnachtsfeiertage, Neujahr und Wochenenden vor, vom 26.1. bis 1.2.1998 fehlen die Werte der Börsen in Singapur und Hongkong und vom 16.2. bis 26.2.1998 die Daten aus Kanada. In Anhang A sind Abbildungen mit den zeitlichen Entwicklungen der Eingabedaten dargestellt und Stellen angegeben, wo sich diese Daten finden lassen (Anhang B).

5.3 Datenpräsentation

Zur Datenpräsentation gehört eine Korrelationsanalyse, die bereits im Vorfeld der Simulation die Komplexität des Problems reduziert. Anschließend werden die übriggebliebenen Daten vorverarbeitet und konvertiert. Es werden also transformierte Zeitreihen gebildet, die das KNN als Eingabe verwenden kann.

³⁰Mittlerweile sind an den Weltbörsen computergestützte Handelssysteme im Einsatz (z.B. seit dem 28.11.1997 das elektronische Handelssystem Xetra an der FWB, das von 8:30 bis 17:04 Uhr MEZ den Xetra-DAX ermittelt), welche die Börsenzeiten verlängern und wochentags einen Handel rund um die Uhr ermöglichen. Diese sehr jungen Informationen bleiben aber unberücksichtigt, weil zum erfolgreichen Trainieren des KNN viele Datensätze benötigt werden und auch nur die Tage in der Simulation verwendet werden können, an denen Daten von allen betrachteten Börsen vorliegen.

Korrelationsanalyse Mit Hilfe einer Korrelationsanalyse lassen sich Abhängigkeiten zwischen den Zeitreihen feststellen und damit die Zahl der Eingabedaten reduzieren. Dazu werden paarweise die linearen Korrelationskoeffizienten der 41 Eingabezeitreihen berechnet. Der Einfachheit halber wird Linearität unterstellt, obwohl diese Annahme auf Finanzmärkten i.a. nicht erfüllt ist (vgl. Kap. 4.6).

Der lineare Korrelationskoeffizient

$$\rho(X, Y) = \frac{\text{COV}(X, Y)}{\sqrt{V(X) \cdot V(Y)}} \quad (5.1)$$

brechnet sich aus Kovarianz

$$\text{COV}(X, Y) = E(X \cdot Y) - E(X) \cdot E(Y) \quad (5.2)$$

und der Varianz

$$V(X) = E(X^2) - E(X)^2 \quad , \quad (5.3)$$

wobei der Erwartungswert

$$E(X) = \frac{1}{n} \sum_{t=1}^n X_t \quad (5.4)$$

als arithmetischer Mittelwert über alle n Datenpunkte X_t der Zeitreihe definiert ist. Es werden Korrelationsanalysen mit $n = \{10, 38, 70, 400\}$ gebildet, was in etwa den realen Zeiträumen von zwei Wochen, zwei Monaten, dem Vorhersagezeitraum und zwei Jahren entspricht. Der Quellcode des benutzten Programms und die berechneten Korrelationskoeffizienten sind im Anhang C angegeben.

Da die Prognose des DAX eine kurzfristige ist, werden den Korrelationskoeffizienten, die über den Zeithorizont von 10 Börsentagen berechnet werden, die größte Bedeutung beigemessen. Gemäß der Arbeit der SNI AG (vgl. Kap. 4.7) wird bei Korrelationskoeffizienten, deren Absolutbeträge größer als 0,85³¹ sind, eine der beiden Zeitreihen in der Eingabe des KNN eliminiert. Es werden nur die Daten zur Korrelationsanalyse herangezogen, die tatsächlich vor der ersten Prognose vorliegen, in diesem Fall also 1702 Datensätze.

Allgemein läßt sich feststellen, daß hohe Korrelationen nur zwischen Zeitreihen innerhalb der Sektoren der Intermarket-Analyse auftreten und daß i.d.R. die Korrelationskoeffizienten mit zunehmender Betrachtungsdauer n größer werden.

³¹In dem Modell der SNI AG lag der Wert bei 0,9. Allerdings ist in deren Arbeit nicht angegeben, welcher Zeitraum der Korrelationsanalyse zugrunde liegt. Außerdem wurde dort die Korrelationsanalyse auf die transformierten Zeitreihen angewendet. In dieser Arbeit werden aber nur monotone Transformationen auf die Zeitreihen durchgeführt, so daß die Anwendung der Korrelationsanalyse vor der Transformation gerechtfertigt ist.

Bei den Aktienindizes zeigen sich hohe Korrelationen zwischen dem Dow Jones Industrial Index, dem NYSE Composite Index und dem S&P Composite 500 Index. Der andere US-Aktienindex NASDAQ Composite korreliert kurzfristig nicht so stark mit diesen, weil er immerhin über 5000 Aktien enthält und außerdem junge Branchen (insbesondere Technologiewerte) in ihm stark gewichtet werden. Außerdem fällt auf, daß DAX und KOSPI sich völlig unkorreliert zueinander verhalten.

Im Devisen-Sektor ergeben sich intensive Korrelationen zwischen den Dollar-Währungen untereinander (mit Ausnahme des Australischen Dollars) und dem südkoreanischen Won. Dies liegt an der starken Kopplung dieser Währungen an den US-amerikanischen Dollar. Ansonsten ergibt sich noch eine Korrelation auf dem Rentenmarkt und zwar zwischen den US-Anleihen mit 30-jähriger und mit 10-jähriger Laufzeit.

Insgesamt können so also 7 der 41 Zeitreihen eliminiert werden: NYSE Composite, S&P 500, Kanadische Dollar, Singapur Dollar, Hongkong Dollar, Koreanische Won und 10 Years US Treasury Bill Rate.

Transformation und Konvertierung Alle Eingabedaten, die in absoluten Geldeinheiten gemessen werden (z.B. Aktienindizes und Devisen) werden zunächst logarithmiert, um einen eventuellen exponentiellen Wachstumstrend (inflationäre Einflüsse) zu beseitigen. Anschließend werden alle Daten, die in täglicher Notierung vorliegen, hinsichtlich der Zeit differenziert

$$\begin{aligned} Y_t^0 &= \ln(X_t) \\ Y_t^i &= \frac{1}{i} (\ln(X_t) - \ln(X_{t-i})) \quad i = 1, \dots, m-1 \end{aligned} \quad (5.5)$$

wobei m den Differentiationsgrad angibt. Die Größe Y_t^i kann auch als stetige Wachstumsrate interpretiert werden. Der Differentiationsgrad gibt die Anzahl der vergangenen Tage einschließlich des heutigen Tages an, die als Eingabedaten für die Prognose verwendet werden sollen.

Schließlich werden alle Zeitreihen skaliert, d.h. auf das Intervall $[a, b]$ abgebildet.

$$Z_t^i = (b - a) \cdot \frac{Y_t^i - \text{MIN}(Y_t^i)}{\text{MAX}(Y_t^i) - \text{MIN}(Y_t^i)} + a \quad (5.6)$$

MAX und MIN bezeichnen jeweils den größten und kleinsten Wert der Zeitreihe Y_t^i . Für die Simulation werden die Parameter $a = 0,1$ und $b = 0,9$ gewählt. Der Quellcode des Programms, das diese Vorverarbeitung durchführt und anschließend die Daten in Dateien schreibt, die zur Simulation benutzt werden können, ist in Anhang D angegeben.

5.4 Durchführung der Simulationen

Stuttgarter Neuronale Netze Simulator Der Stuttgarter Neuronale Netze Simulator (SNNS) wird benutzt, um die Prognose des DAX durchzuführen. Dieses Software-Werkzeug entstand 1989 an der Universität Stuttgart und wird seitdem ständig weiterentwickelt [Z^{ea}95]. Es ist kostenfrei unter Beachtung der Lizenzbedingungen erhältlich. SNNS bietet eine effiziente und flexible Entwicklungsumgebung für Künstliche Neuronale Netze, wird für Forschung und Anwendung eingesetzt und besteht hauptsächlich aus zwei Komponenten: nämlich dem Simulatorkern und einer graphischen Benutzeroberfläche. Der Kern hält interne Datenstrukturen zur Beschreibung des KNN vor sowie diverse Lernfunktionen, Optimierungsverfahren und Analysemöglichkeiten, die von der Benutzeroberfläche aus kontrolliert werden können. Der Simulator SNNS ist komplett in ANSI-C geschrieben und läuft auf Unix-Plattformen unter X-Windows³². Zur Simulation wird eine Workstation Silicon Graphics O2 (180 MHz Prozessor, 192 kB RAM) benutzt.

Aufteilung der Datenmengen Zunächst werden die Daten in zwei disjunkte Mengen unterteilt. Der Trainingszeitraum vom 1.1.1990 bis zum 30.9.1997 enthält 1702 und der Testzeitraum vom 1.10.1997 bis zum 28.2.1998 70 Datensätze (Generalisierungsmenge). Die Daten des Trainingszeitraumes werden noch einmal in zwei disjunkte Mengen aufgeteilt, wobei in chronologischer Reihenfolge jeder vierte Datensatz der Validierungsmenge zugeordnet wird, die restlichen Datensätze bilden dann entsprechend die Trainingsmenge. Es wird ein Differentiationsgrad von 5 verwendet. Somit besteht die Trainingsmenge aus 1273 und die Validierungsmenge aus 424 Mustern.

Netztopologie Als KNN wird ein vollkommenes vernetztes MLP mit einer bzw. zwei inneren Schichten benutzt (vgl. Kap. 2.5). Die Anzahl der Eingabeneuronen ist von der Anzahl der benutzten Eingabegrößen (34), von denen, die differenziert werden (29) und vom Differentiationsgrad (5) abhängig, dann aber eindeutig festgelegt. In diesem Fall werden 150 Eingabeneuronen benötigt. Durch den Vorhersagezeitraum von einem Tag ist die Anzahl der Ausgabeneuronen ebenfalls festgelegt, nämlich 1. Die Anzahl der Neuronen in den inneren Schichten kann beliebig variiert werden. Grob richtet sie sich aber nach der Anzahl der Eingabeneuronen und der Anzahl von inneren Schichten. Als Ausgangspunkt der Netzdimensionierung wird ein festes Übersetzungsverhältnis³³ zwischen den Schichten gewählt, und zwar den Wert 10 beim dreischichtigen MLP und 5 beim

³²Mittlerweile ist auch eine Version für Windows 95 auf dem PC erhältlich.

³³Mit dem Übersetzungsverhältnis ist der Multiplikationsfaktor gemeint, mit dem man, ausgehend von der Ausgabeschicht, die Zahl der Neuronen multiplizieren muß, um auf die Zahl der Neuronen in der nächsten Schicht zu kommen.

vierschichtigen. Während der Simulation ohne Optimierungsverfahren werden die Anzahlen der Neuronen in diesen inneren Schichten variiert. Bei einer Ergebnisverschlechterung (s.u.) wird das neue Netz verworfen und das alte behalten.

Der Einfachheit halber werden die Netze im folgenden hinsichtlich ihrer Topologie mit (NI-NH1-NH2-NO)-MLP bzw. (NI-NH1-NO)-MLP bezeichnet, wobei für die Platzhalter NI die Anzahl der Eingabeneuronen, für NH1 die Anzahl der Neuronen der ersten inneren Schicht, usw. eingesetzt werden.

Lernverfahren Die Gewichte der Verbindungen werden mittels einer Zufallsfunktion initialisiert. Der Wertebereich erstreckt sich von -1 bis +1. Als Aktivierungsfunktion wird von allen Neuronen des KNN die logistische Sigmoid-Funktion (vgl. Kap. 2.5) verwendet. Normalerweise würde man den Wert 0,5 als Schwellenwert einstellen, da der Funktionsbereich der Ein- und Ausgaben auf das Intervall $[0,1 ; 0,9]$ skaliert ist, und weil die Aktivierungsfunktion bei Null am empfindlichsten auf die Eingaben reagiert. Da die Verteilung der skalierten Eingaben aber nicht notwendigerweise symmetrisch um den Intervall-Mittelpunkt verteilt ist, wird der Schwellwert wie ein Gewicht behandelt und während des Trainings mitangepaßt.

Als stabiles Lernverfahren bei den Simulationen hat sich der Backpropagation-Algorithmus mit Momentum-Term und den Parametern $\eta = \{0,1 ; 0,2 ; 0,8\}$ (Lernrate) und $\alpha = 0,8$ (Momentum) erwiesen. Die Trainingsmuster werden dem KNN in zufälliger Reihenfolge präsentiert. Nach 10 Durchläufen (Zyklen oder Epochen) der gesamten Trainingsmenge, wird einmal der Fehler³⁴ des Netzes auf der Validierungsmenge gemessen. Wenn dieser Fehler minimal wird, ist das Training abzubrechen (vgl. Kap. 2.4).

Die Festlegung des Zeitpunktes des Trainingsabbruchs ist nicht eindeutig. Die Fehlerfunktion nimmt in allen Simulationen einen sehr flachen Verlauf an und ist starken Schwankungen ausgesetzt (siehe Abb. 5.3), so daß man nicht von einem echten Minimum sprechen kann. Deshalb wird das Netz zunächst übertrainiert, um einen Anhaltspunkt zu bekommen, wann das Training in etwa abzubrechen ist³⁵. In den nachfolgenden Simulationen wird dann stichprobenartig (d.h. in Abständen von etwa 100 bis 1000 Zyklen je nach Lernrate) um diesen Zeitpunkt,

³⁴Es werden die Summe der Fehlerquadrate (engl. sum of squared errors (SSE)), der mittlere quadratische Fehler (engl. mean squared error (MSE)) und der SSE pro Ausgabeeinheit berechnet. Die Verläufe der drei Fehler sind sehr ähnlich. Da der MSE oft sehr kleine Werte annimmt und dieser Bereich nicht mehr sinnvoll in dem Grafik-Fenster angezeigt werden kann, wird in dieser Arbeit der SSE als Fehlervergleichsmaßstab beim Trainingsstopp herangezogen.

³⁵Ein generelles Problem der Anwendung KNN ist die Reproduzierbarkeit der Ergebnisse. Schon allein die zufällige Initialisierung der Gewichte und die zufällige Reihenfolge der Trainingsmuster beeinflussen das Lernen und damit die Trainingsdauer und kann zu verschiedenen austrainierten Netzen führen und unterschiedliche Ergebnisse liefern.

der Fehler auch auf der Generalisierungsmenge gemessen³⁶. Obwohl der mittlere Fehler der Validierungsmenge sehr klein ist, führen die starken Schwankungen dazu, daß auch der Fehler der Generalisierungsmenge stark schwankt. Es kann außerdem nicht automatisch von einem kleinen absoluten Fehler der Validierungsmenge auf einen ebenfalls kleinen Fehler der Generalisierungsmenge geschlossen werden.

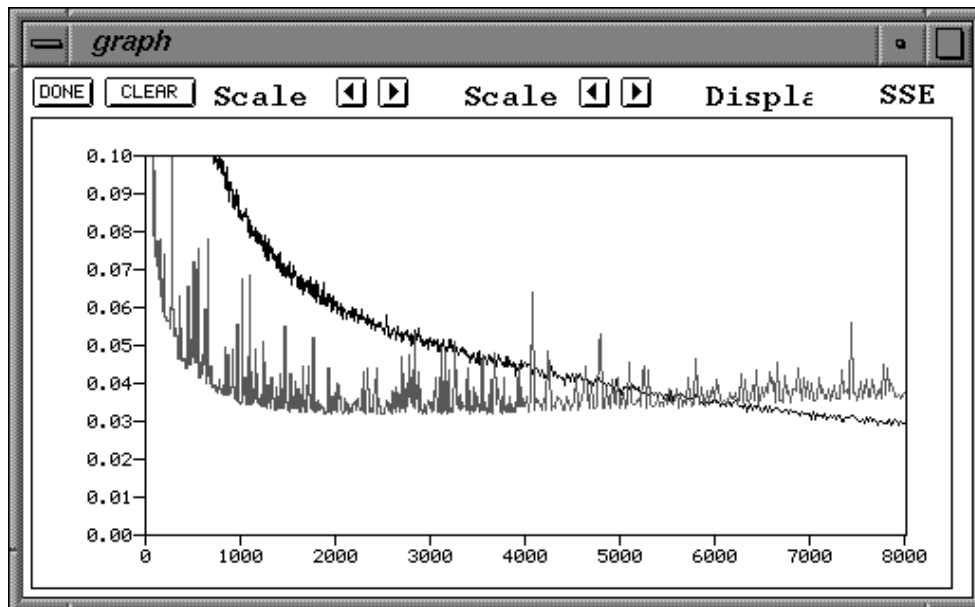


Abbildung 5.3: Typischer Verlauf der Fehlerfunktion von Trainings- und Validierungsmenge während der Simulation. Auf der Abszisse ist die Anzahl der durchlaufenen Zyklen und auf der Ordinate der SSE aufgetragen. Der auf der Validierungsmenge gemessene Fehler ist durch starkes Rauschen gekennzeichnet.

Optimierungsverfahren Die Optimierungswerkzeuge werden auf das KNN angewendet, das zuvor die besten Ergebnisse erzielt hat. Als Optimierungsmethode wird die Ausdünnungstechnik Optimal-Brain-Damage (vgl. Kap. 2.6) verwendet. Diese Art des Weight-Pruning wird mit den Techniken Input-Pruning und Hidden-Merging kombiniert angewendet. Bei einem Fehler von über 10 % (Absolutwert 5) wird das Ausdünnen abgebrochen. Als Lernalgorithmus wird hier ebenfalls Backpropagation mit Momentum-Term verwendet. Es werden 1000 Zyklen trainiert, 100 werden zum Nachtraining benutzt. Bei einem Fehler unter 1 % wird das Training abgebrochen. Nach dem Ausdünnen wird das Netz noch

³⁶Damit wird natürlich schon das Ergebnis verwendet, das prognostiziert werden soll. Zwischen Test und Anwendung muß jedoch unterschieden werden, d.h. erst das austrainierte KNN, das auch den Test bestanden hat, kann in die Anwendung gehen (vgl. Kap. 2.6).

mit dem Lernalgorithmus weitertrainiert, bis das Netz mit dem kleinsten Fehler gefunden ist (s.o.).

Es werden viele Simulationen mit Optimierung gestartet. In den meisten Fällen (ca. 60 %) ist das Optimierungsverfahren allerdings stecken geblieben. Von den übrigen Netzen haben etwa die Hälfte ähnliche Ergebnisse wie ohne Optimierung erzielt. Die restlichen Topologie-reduzierten Netze haben eine bessere Generalisierungsleistung erzielt. Dies steht im Einklang mit den Ergebnissen von Reed [Ree93] und Krogh und Hertz [KH92] und zu der allgemein herrschenden Meinung, daß Netze mit weniger freien Parametern (Gewichten) besser generalisieren können, aber auch im Widerspruch zu den Ergebnissen von Sietsma und Dow [SD91].

Die besten 10 Netze werden schließlich ausgewählt und hinsichtlich ihrer reduzierten Topologie analysiert. Es wird untersucht, welche Eingabegrößen mit Eingabeneuronen, aufgrund geringer Sensitivität nach der Optimierung entfernt werden und welche besonders stark die Ausgabe beeinflussen.

6 Ergebnisse und Diskussion

Es werden Simulationen sowohl mit als auch ohne Optimierungsverfahren durchgeführt und die Ergebnisse anschließend mit denen einer naiven Prognose verglichen. Desweiteren wird ein fiktives Depot, das den nachgebildeten DAX enthält, über den Untersuchungszeitraum verwaltet. Die Werte der Prognosen und die sich aus den Anlageentscheidungen ergebende Depotwertentwicklung ist im Anhang E dargestellt. Außerdem ist dort die Topologie des erfolgreichsten Netzes angegeben.

6.1 MLP ohne Optimierung

Die besten Ergebnisse hinsichtlich der Generalisierungsfähigkeit werden mit einem (150-30-5-1)-MLP mit einer Lernrate $\eta = 0,2$ und mit einem (150-15-1)-MLP mit $\eta = 0,8$ erzielt. Beide Ergebnisse sind sehr ähnlich, die Trainingsdauer ist in beiden Simulationen allerdings unterschiedlich. Das vierschichtige MLP wird ca. 2000 Epochen (Zyklen) trainiert, was einer Rechenzeit von etwa einer Stunde entspricht. Das kleinere Netz mit der höheren Lernrate benötigt nur etwa 300 Zyklen (ca. 10 Minuten Trainingszeit). Das Ergebnis für das (150-15-1)-MLP ist in Abb. 6.1 dargestellt.

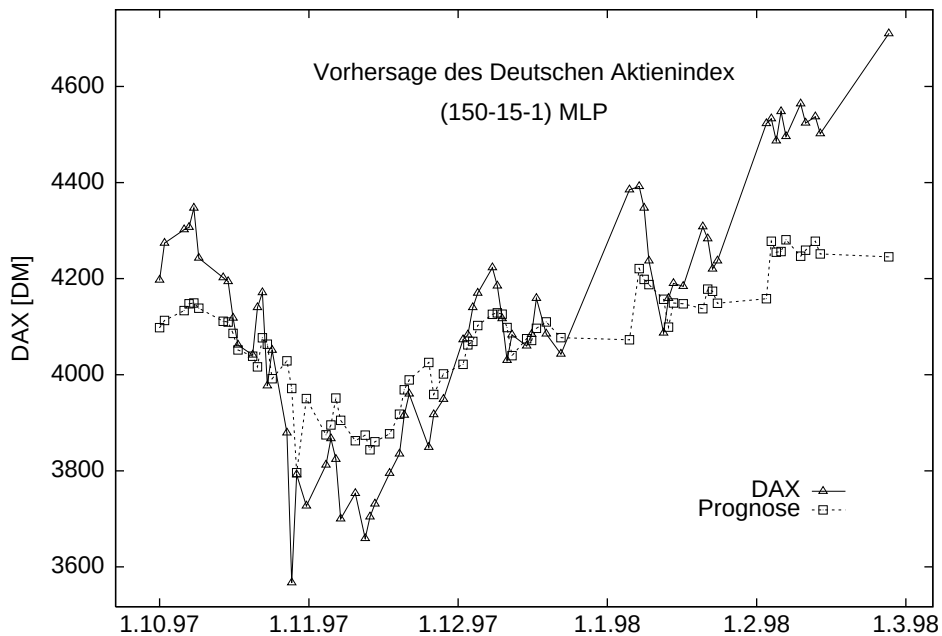


Abbildung 6.1: DAX-Prognose mit dem (150-15-1)-MLP

Die SSE-Fehler der Trainings- und Validierungsmenge sind, bezogen auf die logarithmierten und skalierten DAX-Werte, 0,5348 bzw. 0,1945. Die mittlere Abweichung von dem tatsächlichen DAX-Verlauf beträgt 159,82.

Im Prognosezeitraum erfährt der DAX eine Trendwende. Bis zum 13.11.97 ist ein abnehmender Verlauf erkennbar, anschließend steigen die DAX-Werte. Auffällig sind die extremen Kursschwankungen. Der DAX nimmt sein Minimum von 3567 am 28.10.1997 und sein Maximum von 4710 am 27.2.1998 an. Dies entspricht einer Kursänderung von 1143 oder einer Steigerung von 32 %. Hervorzuheben sind auch die drei Lücken im Vorhersagezeitraum (vgl. Kap. 5.2).

Die mittels KNN prognostizierten absoluten DAX-Werte weichen stark von den tatsächlichen Werten ab. Es werden lediglich die Trends erkannt. Diese sind aber nicht so ausgeprägt wie im tatsächlichen Verlauf. In der Trainings- und Validierungsmenge befinden sich keine so starken und kurzzeitigen Kursänderungen, wie sie im Prognosezeitraum auftreten. Es ist anzunehmen, daß dem KNN Kursschwankungen in ähnlicher Intensität und Kurzfristigkeit als Trainingsmuster präsentiert werden müssen, damit sich die Generalisierungsfähigkeit verbessert. Am 19. Oktober 1987, also fast genau 10 Jahre zuvor, gab es einen noch stärkeren weltweiten Börsen-Crash. Die Trainingsdaten müßten demnach bis ins Jahr 1987 zurückreichen. Es ist allerdings fraglich, ob sich durch das Training des KNN mit diesen zeitlich weit zurückliegenden Daten die Generalisierungsfähigkeit verbessern würde. Denn verglichen mit der Gesamtbetrachtungsdauer, sind die Zeitenräume starker Schwankungen nur sehr kurz.

Ab dem 22.12.1997 wird die Prognose deutlich schlechter. Die prognostizierten DAX-Werte sind zu klein. Für dieses Verhalten gibt es eine plausible Erklärung. Bei der Zeitdifferenzierung der Daten, deren Ergebnis auch als stetige Wachstumsrate interpretiert werden kann, wenn die Daten zuvor logarithmiert worden sind, wird durch die Anzahl der Tage, die zwischen den zwei betrachteten Werten liegen, dividiert (siehe Gl. 5.6). Es wird dabei die Differenz zwischen zwei Börsentagen und nicht die, zwischen den realen Tagen benutzt. Wochenenden und einzelne Feiertage verfälschen zwar diese Wachstumsraten etwas, das KNN erkennt und korrigiert diese kleinen Fehler aber, da sie sehr häufig und regelmäßig vorkommen. Nach dem 22.12.1997 ist aber der nächste Prognosetag der 5.1.1998. Es liegen also 14 Tage zwischen beiden. In die Berechnung der Wachstumsrate zwischen diesen Daten wird aber nur eine 1 als Divisor benutzt. Berücksichtigt man die 4 Tage der Wochenenden in diesem Zeitraum und korrigiert (multipliziert) die Kurssteigerung in der Prognose ($4073 - 4043 = 30$) mit dem Faktor 10, dann ergibt sich ein prognostizierter Wert von 4343 (statt 4073), der dem tatsächlichen Wert von 4385 schon sehr nahe kommt. Analog lassen sich die Sprünge zwischen dem 26.1. und 5.2. bzw. zwischen dem 13.2. und 27.2.1998 erklären.

Im allgemeinen nimmt die Prognosegüte ab, je länger der Trainingszeitraum zurückliegt. Neue Größen und Zusammenhänge, die das KNN bis dahin noch nicht gelernt hat, können inzwischen die Kursbewegungen beeinflussen. Deshalb wird man ein trainiertes KNN in der Anwendung nur für eine kurze Zeitdauer verwenden. Prinzipiell besteht die Möglichkeit, an jedem Tag des Prognosezeitraumes ein neues KNN zu trainieren, wobei die Trainings- oder Validierungsmenge um einen neuen Wert, nämlich den heutigen, ergänzt wird. Diese Methodik ist aber sehr rechenintensiv und dieser Aufwand steht zu keinem Verhältnis zum Ergebnis, zumindest nicht im Rahmen dieser Arbeit.

6.2 MLP mit Optimierung

Das (150-15-1)-MLP wird, wie in Kap. 5.4 beschrieben, optimiert. Dazu wird während der Optimierung eine Lernrate von $\eta = 0,2$ verwendet und nach dem Ausdünnen diese auf 0,1 reduziert. Die Optimierung benötigt etwa eine Stunde Rechenzeit, das anschließende Training des optimierten Netzes wenige Minuten. Die Optimierung hat zu einer sehr starken Komplexitätsverminderung der Netze geführt. Das kleinste Netz benutzt nur 4 Eingabeneuronen, 2 innere Neuronen und hat 6 anzupassende Gewichte. Das größte besteht aus 26 Eingaben, 5 verborgenen Neuronen und 34 Gewichten. Das beste Netz ist ein (20-4-1)-MLP mit 24 statt ursprünglich 2265 Gewichten. Das Prognose-Ergebnis dieses Netzes ist in

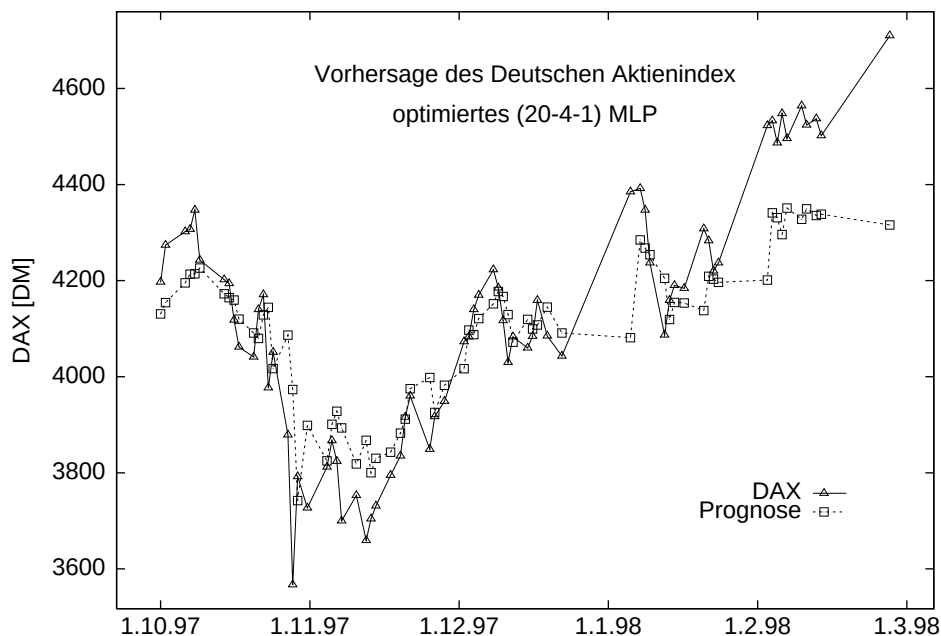


Abbildung 6.2: DAX-Prognose mit dem (20-4-1)-MLP

Abb. 6.2 dargestellt. Die Fehler der Trainings- und Validierungsmengen betragen 0,2056 bzw. 0,0730 bezüglich der logarithmierten und skalierten DAX-Werte. Die mittlere Standardabweichung beträgt 133,43 im Vorhersagezeitraum.

Der Verlauf der DAX-Prognose ist ähnlich dem des (150-15-1) MLP. Die Anpassung an den tatsächlichen DAX-Wert ist jedoch schon etwas besser. In beiden Fällen konnte aber der Börsencrash nicht vorhergesagt werden. Vielmehr zeigt sich, daß das Netz eine einfache Heuristik für die Vorhersage ableitet. Es benutzt den heutigen DAX-Wert als Basiswert und addiert bzw. subtrahiert (je nach vorherrschenden Trend) einen kleinen Beitrag zu dieser Basis. Denn dann wird der Fehler, den man macht, gering bleiben. Diese Heuristik beruht auf der Tatsache, daß eine Kurstendenzänderung unwahrscheinlicher ist, als eine Beibehaltung der Kurstendenz. Zu dieser Erkenntnis ist bereits E. Schöneburg in seinen Simulationen mit KNN gekommen [SHG90] (vgl. Kap. 4.7).

Die Analyse der 10 besten optimierten Netzen deckt folgende Zusammenhänge auf. Es stellt sich heraus, daß in keinem der 10 Netze, die Eingaben Wochentag, Diskontsatz BRD, Geldmenge M3 USA und BRD, Gold-Silber-Index, Schwedische Kronen, Italienische Lire, NASDAQ Composite, FTSE 100 und DAX als Eingaben verwendet werden. Letzteres verwundert, weil man eigentlich vermutet, daß die eigene Historie am wichtigsten für die weitere Werteentwicklung sein müßte. In nur einem Netz sind der Diskontsatz der USA, japanische Yen und der KOSPI als Eingaben benutzt worden. Die wichtigste Eingabegröße in diesen Simulationen ist der Öl-Gas-Index. In 6 der 10 Netzen werden 3 oder mehr der insgesamt 5 Eingabenneuronen dieses Inputs zur Berechnung des Outputs benutzt.

6.3 Naive Prognose

Aufgrund der Feststellung, daß die benutzten KNN eine einfache Heuristik zur Vorhersage ableiten, die in erster Linie den Wert des Vortages berücksichtigt, werden die Simulationsergebnisse mit den Ergebnisse einer naiven Prognose verglichen. Diese nimmt an, daß der DAX-Wert von morgen dem heutigen Wert entspricht. Den mittleren Fehler, den man so macht, beträgt 101,07 im Untersuchungszeitraum.

Damit ist die Generalisierungsleistung der naive Prognose zwar besser als die Vorhersagen beider KNN, aber absolut betrachtet auch nicht überzeugend. Geht man von einem DAX-Wert von 4000 aus, dann entspricht eine Abweichung von 100 einem relativen Fehler von 2,5 %. Das Ergebnis ist in Abb. 6.3 dargestellt.

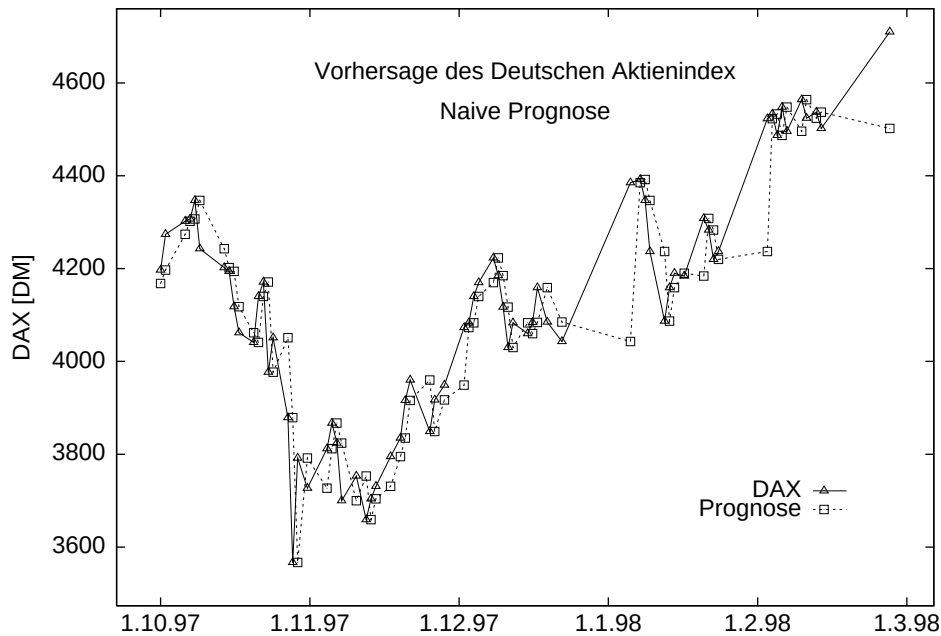


Abbildung 6.3: Naive DAX-Prognose

6.4 Anlageentscheidungen

Basierend auf den DAX-Prognosen können Kauf/Verkauf-Entscheidungen für DAX-Zertifikate³⁷ getroffen werden. Dazu wird am 30.9.1997 ein fiktives Depot mit dem Wert von 100000 DM eingerichtet, das komplett in ein DAX-Zertifikat investiert wird. Der DAX-Wert zu diesem Zeitpunkt beträgt 4168. Das Modell geht von sehr einfachen Annahmen aus, was den Vorteil einer leichten Handhabung auf Kosten von z.T. unrealistischen Bedingungen mit sich bringt. Es wird angenommen, daß die DAX-Prognose, die um 15:00 Uhr MEZ durchgeführt wird, nur wenig Zeit in Anspruch nimmt, denn das Netz ist bereits trainiert. Es müssen lediglich die Eingaben vorverarbeitet werden. Weiterhin wird angenommen, daß die Kauf- oder Verkaufaufträge nach der Vorhersage noch zu den Preisen der Schlußnotierungen des amtlichen Handels ausgeführt werden, d.h. es wird unterstellt, daß sich der DAX im weiteren Tagesverlauf nicht mehr ändert und die Möglichkeit besteht, zu dieser Zeit Aktien zu kaufen (IBIS³⁸- oder Xetra-Handel). Bei den Käufen und Verkäufen sollen weder Transaktionskosten anfallen, noch wird eine Spekulationssteuer (Kapitalertragssteuer) auf die erwirtschafteten Erträge berücksichtigt.

³⁷Index-Zertifikate sind Investmentfonds, die den Index nachbilden und deren Wertentwicklung identisch mit dem des Index ist.

³⁸Am 5.4.1991 wurde das Integrierten Börsenhandels- und Informationssystem (IBIS) eingeführt, an das bundesweit etwa 250 Kreditinstitute und Makler angeschlossen sind.

Die Anlagestrategien beider Prognosearten sind klar, wenn man rationales Verhalten der Anleger unterstellt. Bei der naiven Prognose kommt nur die Buy-and-Hold-Strategie in Frage, denn der prognostizierte Kurs ändert sich nicht. Die DAX-Prognosen mittels KNN führen zu Kaufentscheidungen, wenn ein steigender DAX und entsprechend zu Verkaufsentscheidungen, wenn ein fallender DAX vorhergesagt wird. Abgerechnet werden die Depots am 27.2.98, der DAX nimmt dort einen Wert von 4710 an. Somit ergibt sich ein Depotwert von 113004 DM für die Buy-and-Hold-Strategie, was einer Steigerung (Rendite) von 13 % entspricht.

Das nicht-optimierte (150-15-1)-MLP trifft 7 Verkauf- und 6 Kaufentscheidungen. Der Depotwert beträgt am Ende des Untersuchungszeitraumes 122516 DM (22,5 % Rendite). Das optimierte (20-4-1)-MLP trifft 6 Verkauf- und 5 Kaufentscheidungen. Der Depotwert beträgt 115489 DM (15,5 % Rendite). Damit liegen die Ergebnisse beider KNN etwas über dem der naiven Prognose.

Es lassen sich vier interessante Beobachtungen machen (siehe Abb. E):

- Die erzielte Rendite des ursprünglichen KNN liegt über der des optimierten.
- Jede einzelne Kauf- und Verkaufsentscheidung beider KNN ist erfolgreich.
- Die letzte Verkaufsentscheidung wird am 13.1.1998 getroffen. An diesem Tag ist der Wert des DAX 4159 und liegt damit um 0,2 % unter dem Startwert vom 30.9.1997. Im nachfolgenden Zeitraum werden keine Transaktionen mehr durchgeführt, da der DAX unterbewertet prognostiziert wird (s.o.). In dieser Zeit legt der DAX aber noch einmal um 13 % zu, was genau der Rendite der Buy-and-Hold-Strategie entspricht.
- Je nach Auswertungskriterium (Generalisierungsfähigkeit oder Rendite) können die Vergleiche zwischen KNN und anderen Verfahren (naive Prognose bzw. Buy-and-Hold-Strategie) zu unterschiedlichen Ergebnissen führen.

Wenn nur der Zeitraum bis zum 13.1.1998 betrachtet wird, dann schlagen die getroffenen Anlageentscheidungen auf Grundlage der Prognosefähigkeit KNN die Buy-and-Hold-Strategie deutlich. Dies kann aber auch Zufall sein. Desweiteren würde man erwarten, daß das KNN mit der besseren Generalisierungsleistung auch die besseren Anlageentscheidungen trifft. Dies ist aber hier nicht der Fall. Um diese Aussagen zu verifizieren oder falsifizieren, sind weitere Simulation notwendig. Abschließend sei angemerkt, daß es einerseits bessere Analysemethoden als diese naive Prognose gibt und andererseits Transaktionskosten und Steuern den erzielten Gewinn deutlich zusammenschrumpfen lassen.

Die in Kap. 4.7 vorgestellten Modelle zur Aktienkursprognose mit Hilfe KNN haben als Vergleichsmaßstab immer die Rendite (oder die Trefferquote der Steigt/Fällt-Prognosen) benutzt und als herkömmliches Verfahren die Buy-and-Hold-Strategie dem KNN-Ergebnis gegenübergestellt. Wie gut oder schlecht die Generalisierungsleistung der KNN dieser Arbeiten ist, wird nicht angegeben. Nach den Ergebnissen und Erfahrungen dieser Arbeit, ist es auch möglich, daß die (scheinbaren) Erfolge der früheren Arbeiten Zufall gewesen sind.

7 Zusammenfassung und Ausblick

Die vorliegende Arbeit behandelt den Einsatz Künstlicher Neuronaler Netze auf dem Aktienmarkt. Ausgehend von dem Modell der Intermarket-Analyse werden Simulationen mit KNN durchgeführt, die versuchen, den Absolutwert des Deutschen Aktienindex für den nächsten Börsentag im Zeitraum zwischen dem 1.10.1997 und 28.2.1998 vorherzusagen.

Als Eingabegrößen dienen Wirtschafts- und Börseninformationen aus den Sektoren Aktienindizes, Devisen, Renten und Rohstoffe seit dem 1.1.1990, sowie Handelsvolumen und monetäre Größen (Diskontsätze und Geldmengen). Zu 150 Eingabeneuronen werden transformierte Zeitreihen aus 34 Rohdatenreihen gebildet. 7 Zeitreihen konnten zuvor bereits mit Hilfe von Korrelationsanalysen eliminiert werden.

Die besten Simulationsergebnisse werden mit einem vollständig vernetzten Mehrschichten-Perzeptron unter Verwendung des Backpropagation-Algorithmus mit Momentum-Term als Lernverfahren erzielt. Als Optimierungstechniken werden Optimal Brain Damage kombiniert mit Input-Pruning und Hidden-Merging angewendet. Für das reduzierte (20-4-1)-MLP müssen nur noch 24 der ursprünglich 2265 Gewichte angepasst werden. Die wichtigsten Informationen stammen aus dem Öl-Gas-Index, die Historie des DAX wird nicht als Eingabe benutzt.

Die Generalisierungsfähigkeit beider KNN ist schlecht (mittlerer Fehler von 159,82 bzw. 133,43), verglichen mit dem Ergebnis der naiven Prognose (mittlerer Fehler von 101,07), die als Vorhersage den heutigen Wert benutzt. Dies läßt sich auf die starken Schwankungen des Kursverlaufs des DAX während des Prognosezeitraumes zurückführen. Die Differenz zwischen Minimum und Maximum beträgt 1143, was einer Steigerung von 32 % entspricht. Der Börsencrash am 28. Oktober kann mit keinem dieser Netze vorhergesagt werden.

Im Prognosezeitraum sind 3 größere Lücken (bis zu 14 Tagen) aufgrund der Weihnachtsfeiertage und dem Fehlen von Börsendaten aus Singapur, Hongkong und Toronto enthalten. Da bei der Berechnung der stetigen Wachstumsraten zwischen zwei Daten nicht die Zahl der tatsächlichen Tage, sondern die der Börsentage verwendet wird, führt dies zu einer Unterbewertung der DAX-Werte zu diesen Zeiten.

Ausgehend von den drei Prognosen werden Anlageentscheidungen für die Verwaltung eines fiktiven Depots getroffen. Die Ergebnisse der KNN (15,5 % bzw. 22,5 % Rendite) liegen etwas über dem der Buy-and-Hold-Strategie (13 %). Es fällt auf, daß das Ergebnis des nicht-optimierten KNN besser ist als das des optimierten. Insbesondere werden nach dem 13.1.1998 keine Anlageentscheidungen aufgrund der minderbewerteten DAX-Werte getroffen. Bis zu diesem Zeitpunkt hat die Buy-and-Hold-Strategie ein Minus von 0,2 % erwirtschaftet. Es kann allerdings nicht ausgeschlossen werden, daß die guten Renditeergebnisse beider

KNN Zufälle sind. Infolge des spärlich vorliegenden Datenmaterials läßt sich dieses Ergebnis nicht hinreichend exakt untermauern. Desweiteren konnte keine Korrelation zwischen der Generalisierungsleistung KNN und der erzielten Rendite festgestellt werden.

Weitere Simulationen zur Verbesserung der statistischen Datenlage sind zur Klärung dieser Fragen erforderlich. Die Berechnung der stetigen Wachstumsraten sollte dabei nach einer anderen Formel geschehen: als Divisor sollte die tatsächliche Zahl der Tage zwischen den beiden zu vergleichenden Werten benutzt werden. Dann stellen große Lücken in den Zeitreihen keine Probleme mehr da. Möglicherweise verbessern sich die Ergebnisse bei Verwendung eines höheren Differentiationsgrads, denn dann werden auch längerfristige Einflußgrößen berücksichtigt. Vorstellbar sind auch andere Differentiationsmethoden, beispielsweise die Bildung von gleitenden Durchschnitten. Es ist fraglich, ob die Erweiterung der Trainingsmenge in die Vergangenheit eine Verbesserung bringt (vgl. Kap. 6).

Berechtigt erscheint jedoch die Frage, ob eine Aktienkursprognose mit KNN angesichts der Einführung von elektronischen Börsen-Handelssystemen, die minütlich neue Kurse liefern, überhaupt sinnvoll ist. Das KNN benötigt viele Datensätze zum erfolgreichen Trainieren, so daß diese aktuellen Informationen, momentan noch nicht verwendet werden können (vgl. Kap. 5.2). Außerdem sind die vereinfachten Modellannahmen (konstanter DAX-Wert nach Parketthandel, keine Transaktionskosten) unrealistisch. Dem kann entgegengestellt werden, daß zusätzliche Informationen, wie beispielsweise die Werte des Xetra-DAX nach 13:30 Uhr MEZ, dazu verwendet werden können, einen Vergleich zwischen der Prognose für den nächsten Tag und dem gerade aktuellen Wert durchzuführen. Infolge dieses Vergleichs läßt sich erkennen, ob der prognostizierte Trend bereits bestätigt wird.

Bei einer erfolgreichen Ein-Tagesprognose mittels KNN - meiner Meinung nach ist das Potential für das Gelingen dieser kurzfristigen Vorhersagen bereits vorhanden -, wären sowohl sehr kurzfristige als auch längerfristige Prognosen möglich. Vorstellbar ist beispielsweise die Vorhersage der nächsten 5 Börsentage mit dem bisherigen Modell, wenn Eingaben mit einem höheren Differentiationsgrad berücksichtigt werden. Die Vorhersage von sehr kurzfristigen Kursänderungen (im Minuten-Bereich) wäre für sogenannte Daytrader³⁹ interessant. Die Datenbeschaffung könnte in Echtzeit über Anbieter im Internet erfolgen, die Datenverarbeitung würde vollständig automatisiert ablaufen. Die Rechenleistung heutiger Rechner dürften hierfür schon ausreichend sein, vorausgesetzt das KNN ist bereits fertig trainiert. Gerade im letzten Punkt liegt aber des Pudels Kern.

³⁹Ein Daytrader ist ein Anleger, der zu Beginn und am Ende des Börsentags zwar Bargeld aber keine Aktien besitzt und während der Handelszeit fleißig Wertpapiere kauft und verkauft.

8 Literaturverzeichnis

- [ACW97] A. ABHYANKAR, L.S. COPELAND und W. WONG, *Uncovering Non-linear Structure in Real-Time Stock-Market Indexes: The S&P 500, the DAX, the Nikkei 225, and the FTSE-100*, Journal of Business and Economic Statistics **15** (1997) 1-14.
- [AHS85] D.H. ACKLEY, G.E. HINTON und T.J. SEJNOWSKI, *A Learning Algorithm for Boltzmann Machines*, Cognitive Science **9** (1985) 147-169.
- [Ale89] I. ALEKSANDER, *Neural Computing Architectures - The Design of Brain-like Machines*, North Oxford Academic Publishers Ltd., Guildford, 1989.
- [All95] P. ALLEN, *Intelligent, Self-organizing Models in Economics and Finance*, in S. GOONTILAKE et al. (Hrsg.), *Intelligent Systems for Finance and Business*, John Wiley & Sons, New York, 1995.
- [AMV94] E.I. ALTMANN, G. MARCO und F. VARETTO, *Corporate Distress Diagnosis: Comparisons Using Linear Discriminant Analysis and Neural Networks (the Italian Experience)*, Journal of Banking and Finance **18** (1994) 505-529.
- [And70] J.A. ANDERSON, *Two Models for Memory Organization*, Mathematical Bioscience **8** (1970) 137-160.
- [And72] J.A. ANDERSON, *A Simple Neural Network Generating an Interactive Memory*, Mathematical Bioscience **14** (1972) 197-220.
- [Bac00] L.J.B.A. BACHELIER, *Théorie de la Spéculation*, Dissertation, Annales de l'École Normale Supérieure, Paris, 1900.
- [Bar91] T. BARR, *Netze im Aufwind*, c't **4** (1991) 78-92.
- [Bau94] S. BAUN, *Neuronale Netze in der Aktienkursprognose*, in H. REH-KUGLER und H.G. ZIMMERMANN (Hrsg.), *Neuronale Netze in der Ökonomie*, Verlag Franz Vahlen, München, 1994, 131-207.
- [Be^a94] J. BAETGE et al., *Bonitätsbeurteilung von Jahresabschlüssen nach neuen Recht (HGB 1985) mit Künstlichen Neuronalen Netzen auf der Basis von Clusteranalysen*, Der Betrieb **7** (1994) 337-343.

- [Bec94] J. BECKER, *Konstruktionsbegleitende Kalkulation mit Neuronalen Netzen innerhalb einer integrierten Informationssystemarchitektur*, in A.-W. SCHEER (Hrsg.), *Rechnungswesen und EDV. Innovatives Controlling - Der Weg zum Turnaround*, Physica-Verlag, Heidelberg, 1994, 157–167.
- [BH89] E.B. BAUM und D. HAUSSLER, *What Size Net Gives Valid Generalization?*, *Neural Computation* **1** (1989) 151-160.
- [BP93] J. BECKER und M. PRISCHMANN, *Supporting the Design Process with Neural Networks - A Complex Application of Cooperating Neural Networks and its Implementation*, *Journal of Information Science and Technology* **3** (1993) 1.
- [Bra95] R. BRAUSE, *Neuronale Netze*, Teubner Verlag, Stuttgart, 1995.
- [BS92] S. BAUN und A. STORM, *Neuronale Netze kennen heute schon die Zukunft*, *Banking and Finance* **6** (1992) 8-11.
- [BW93] H. BRAUN und J. WEISBROD, *Evolving Neural Networks for Application Oriented Problems*, in D.B. FOGEL (Hrsg.), *Proceedings of the Conference on Evolutionary Programming*, La Jolla (CA), 1993.
- [Ces96] G. CESAR, *Aktienanalyse heute*, Gabler Verlag, Wiesbaden, 1996.
- [CF93] P.K. COATS und L.F. FANT, *Recognizing Financial Distress Patterns Using a Neural Network Tool*, *Financial Management* **22** (1993) 142-155.
- [CGL91] K.G. COLEMAN, T.J. GRAETTINGER und W.F. LAWRENCE, *Neural Networks for Bankruptcy Prediction: The Power to Solve Financial Problems*, *AI Review* **Jul/Aug** (1991) 48-50.
- [Cha89] Y. CHAUVIN, *A Back-Propagation Algorithm with Optimal Use of Hidden Units*, in R. LIPPMANN, J. MOODY und D. TOURETZKY (Hrsg.), *Advance in Neural Information Processing Systems*, Morgan Kaufmann, San Francisco, 1989.
- [CM95] H. CORSTEN und C. MAY, *Unterstützungspotential Künstlicher Neuronaler Netze für die Produktionsplanung und -steuerung*, *Information Management* **2** (1995) 44-55.
- [Coo64] P. COOTNER, *The Random Character of Stock Market Prices*, MIT Press, Cambridge (MA), 1964.

- [dBS95] B. MARTIN DEL BRIO und C. SERRANO-CINCA, *Self-organizing Neural Networks: The Financial State of Spanish Companies*, in A.-P. REFENES (Hrsg.), *Neural Networks in the Capital Markets*, John Wiley & Sons, New York, 1995, 341–357.
- [Did95] R.S. DIDNER, *Intelligent Systems at American Express*, in S. GOON-TILAKE et al. (Hrsg.), *Intelligent Systems for Finance and Business*, John Wiley & Sons, New York, 1995.
- [E^{ea}92] K. ERXLEBEN et al., *Klassifikation von Unternehmen: Ein Vergleich von Neuronalen Netzen und Diskriminanzanalyse*, Zeitschrift für Betriebswirtschaft **62** (1992) 1237-1262.
- [EM76] R.D. EDWARDS und J. MAGEE, *Technische Analyse von Aktien-trends*, Hoppenstedt, Darmstadt, 1976.
- [Fai93] J. FAISST, *Hierarchische Planung unter Einsatz Neuronaler Netze - Illustriert an Untersuchungen zum flexiblen Personalmanagement*, Springer Verlag, Heidelberg, 1993.
- [Fam65] E.F. FAMA, *Random Walks in Stock Market Prices*, Financial Analysts Journal **Sep/Okt** (1965) 55-59.
- [Fri93] W. FRISCH, *Die Architektur- und Werteeinstellungsproblematik der Parameter Neuronaler Netze*, in W. FRISCH und A. TAODES (Hrsg.), *Tagungsband Symposium Informationswirtschaft*, Physica-Verlag, 1993, 42–73.
- [Fuk80] K. FUKUSHIMA, *Neocognitron: A Self-Organized Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position*, Biological Cybernetics **36** (1980) 193-202.
- [Füs95] K. FÜSER, *NeuroBond? - Zinsprognose mit Neuronalen Netzwerken*, Anlage Praxis **1** (1995) 6-10.
- [Füs96] K. FÜSER, *Neuronale Netze in der Finanzwirtschaft*, Gabler Verlag, Wiesbaden, 1996.
- [Gab93] *Gabler-Wirtschafts-Lexikon*, Gabler Verlag, Wiesbaden, 1993.
- [GDW94] W. GAUL, R. DECKER und F. WARTENBERG, *Neuronale Netze in der Kaufverhaltensforschung: Alternativ-Modell*, Absatzwirtschaft **7** (1994) 66-69.
- [GJ79] M. GAREY und D. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York, 1979.

- [Goo95] S. GOONTILAKE, *Intelligent Systems in Finance and Business*, John Wiley & Sons, New York, 1995.
- [Gra76] J.E. GRANVILLE, *Granville's New Strategy of Daily Stock Market Timing for Maximum Profit*, Prentice Hall, Englewood Cliffs (NJ), 1976.
- [Gra91] J. GRAF, *Stock Market Prediction with Neural Networks*, in P. GRITZMANN (Hrsg.), *Operations Research 91 - 16 th Symposium on Operations Research*, Heidelberg, 1991, 496–499.
- [Gro76] S. GROSSBERG, *Adaptive Pattern Classification and Universal Recording I + II*, *Biological Cybernetics* **23** (1976) 187-202.
- [GS91] U. GÖCKEN und K.-W. SCHULTE, *Fundamentale Aktienanalyse - Die Praxis deutscher Kreditinstitute*, Verlag Josef Eul, Bergisch Gladbach, 1991.
- [GU68] R. GOSCINNY und A. UDERZO, *Asterix der Gallier*, Egmont Ehapa Verlag, Stuttgart, 1968.
- [Heb49] D.O. HEBB, *The Organization of Behavior*, John Wiley & Sons, New York, 1949.
- [HFZ92] F. HERGET, W. FINNOFF und H.G. ZIMMERMANN, *A Comparison of Weight Elimination Methods for Reducing Complexity in Neural Networks*, Arbeitspapier, Siemens AG, Corporate Research and Development, München, 1992.
- [HKP91] J. HERTZ, A. KROGH und R.G. PALMER, *Introduction to the Theory of Neural Networks*, Addison-Wesley, Redwood City, 1991.
- [HN87a] R. HECHT-NIELSEN, *Counterpropagation Networks*, *Applied Optics* **26** (1987) 4979-4984.
- [HN87b] R. HECHT-NIELSEN, *Kolmogorov's Mapping Neural Network Existence Theorem*, *Proceedings of the First IEEE International Conference on Neural Networks* (San Diego), Vol. III, 1987, 11–14.
- [HN93] H. HRUSCHKA und M. NATTER, *Analyse von Marktsegmenten mit Hilfe konnektionistischer Modelle*, *Zeitschrift für Betriebswirtschaft* **63** (1993) 425-442.
- [Hoh91] B. HOHLER, *Neuronale Netze sichern Produktqualität*, *Computerwoche* **40** (1991) 58-64.

- [Hop82] J.J. HOPFIELD, *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, Proceedings of the National Academy of Sciences **79** (1982) 2554-2558.
- [Hru91] H. HRUSCHKA, *Einsatz Künstlicher Neuronaler Netzwerke zur Datenanalyse im Marketing*, Marketing-ZFP **4** (1991) 217-225.
- [HZ92] G. HANTSCHHEL und H.G. ZIMMERMANN, *Neuronale Netze zur Prognose in der Finanz- und Automobilwirtschaft*, Intelligente Software-Technologien **3** (1992) 46-48.
- [K^{ea}90] T. KIMOTO et al., *Stock Market Prediction Systems with Modular Neural Networks*, Proceedings of the International Joint Conference on Neural Networks (San Diego), Vol. I, IEEE Network Council, 1990, 1-6.
- [KH92] A. KROGH und J. HERTZ, *A Simple Weight Decay can Improve Generalization*, in J. MOODY, S. HANSON und R. LIPPMANN (Hrsg.), *Advances in Neural Processing Systems*, Morgan Kaufman, San Francisco, 1992, 950-958.
- [Kna94] R. KNAUS, *Neuronale Netze: Ermutigende Dollarkursprognosen*, Märkte und Trends **Dez** (1994) 9-13.
- [Koh72] T. KOHONEN, *Correlation Matrix Memories*, IEEE Transactions on Computers C- **21** (1972) 353-359.
- [Koh82] T. KOHONEN, *Self-Organized Formation of Topologically Correct Feature Maps*, Biological Cybernetics **43** (1982) 59-69.
- [Kol57] A.N. KOLMOGOROV, *On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition*, Doklady Akademii Nauk SSR **114** (1957) 953-956.
- [Kos87] B. KOSKO, *Adaptive Bidirectional Associative Memories*, Applied Optics **26** (1987) 4947-4960.
- [KP94] M. KERLING und T. PODDING, *Klassifikation von Unternehmen mittels KNN*, in H. REHKUGLER und H.G. ZIMMERMANN (Hrsg.), *Neuronale Netze in der Ökonomie*, Verlag Franz Vahlen, München, 1994, 427-490.
- [KW94] S. KIRN und C. WEINHARDT, *Künstliche Intelligenz in der Finanzberatung*, Gabler Verlag, Wiesbaden, 1994.

- [Las50] K.S. LASHLEY, *In Search of the Engram*, in J.R. HILLMAN (Hrsg.), *Society of Experimental Biological Symposium No.4*, Cambridge University Press, London, 1950.
- [LDS89] Y. LE CUN, J.S. DENKER und S.A. SOLLA, *Optimal Brain Damage*, in R. LIPPMANN, J. MOODY und D. TOURETZKY (Hrsg.), *Advance in Neural Information Processing Systems*, Morgan Kaufmann, San Francisco, 1989, 598.
- [Le 86] Y. LE CUN, *Learning Processes in an Asymmetric Threshold Network*, in E. BIENESTOCK, F. FOGELMAN SOULI und G. WEISBUCH (Hrsg.), *Disordered Systems and Biological Organization*, Springer Verlag, Berlin, 1986.
- [Lev68] R.A. LEVY, *The Relative Strength Concept of Common Stock Price Forecasting*, New York, 1968.
- [Loh94] T. LOHRBACH, *Einsatz von Künstlichen Neuronalen Netzen für ausgewählte betriebswirtschaftliche Aufgabenstellungen und Vergleich mit konventionellen Lösungsverfahren*, unitext, Göttingen, 1994.
- [Loi91] O. LOISTL, *Kapitalmarkttheorie*, Oldenbourg, München, 1991.
- [Lut93] J. LUTZ, *Motor-Diagnose mit Neuronalen Netzen*, in E. SCHÖNEBURG (Hrsg.), *Industrielle Anwendung Neuronaler Netze*, Addison-Wesley, Bonn, 1993, 29–40.
- [Mal94] M.E. MALLIARIS, *Modeling the Behavior of the S&P 500 Index: A Neural Network Approach*, in *Proceedings of the tenth Conference on Artificial Intelligence for Applications*, San Antonio, Texas, März 1994.
- [Man77] B.B. MANDELBROT, *Fractals: Form, Change, and Dimension*, Freeman, San Francisco, 1977.
- [Mat94] R. MATTHES, *Zinsprognosen: Fehlerkorrekturmodelle vs. Neuronale Netze*, in G. BOL, G. NAKHAEIZADEH und K.-H. VOLLMER (Hrsg.), *Finanzmarktanwendungen Neuronaler Netze und ökonomische Verfahren*, Physica-Verlag, Heidelberg, 1994, 41–60.
- [May96] C. MAY, *Reihenfolgenplanung mit funktionsbildenden Neuronalen Netzen*, Schriften zum Produktionsmanagement der Universität Kaiserslautern **Nr. 3** (1996) 1.
- [Mec95] B. MECHLER, *Intelligente Informationssysteme - Fuzzy Logic und Neuronale Systeme zur Unterstützung von Managemententscheidungen*, Addison-Wesley, Reading (MA), 1995.

- [MHP95] A.J. MAREN, C. HARSTON und R. PAP, *Handbook of Neural Computing Applications*, Academic Press, San Diego, 1995.
- [Mil90] R.M. MILLER, *Computer Aided Financial Analysis*, Addison-Wesley, Reading (MA), 1990.
- [MMM93] R.W. MCLEOD, D.K. MALHOTRA und R. MALHOTRA, *Predicting Credit Risk: A Neural Network Approach*, Journal of Retail Banking **15** (1993) 37-40.
- [MMM94] D.K. MALHOTRA, R. MALHOTRA und R.W. MCLEOD, *Artificial Neural Systems in Commercial Lending*, The Bankers Magazine **177** (1994) 40-44.
- [MP43] W.S. MCCULLOCH und W. PITTS, *A Logical Calculus of the Ideas Immanent in Nervous Activity*, Bulletin of Mathematical Biophysics **5** (1943) 115-133.
- [MP69] M. MINSKY und S. PAPERT, *Perceptrons*, MIT Press, Cambridge (MA), 1969.
- [MRS95] B. MÜLLER, J. REINHARDT und M.T. STRICKLAND, *Neural Networks - An Introduction*, Springer Verlag, Berlin, 1995.
- [Mur92] J.J. MURPHY, *Technische Intermarket-Analyse*, Campus Verlag, Frankfurt, 1992.
- [NKK96] D. NAUCK, F. KRAUSE und R. KRUSE, *Neuronale Netze und Fuzzy-Systeme*, Vieweg, Braunschweig, 1996.
- [OS90] M.D. ODOM und R. SHARDA, *A Neural Network for Bankruptcy Prediction*, Proceedings of the International Joint Conference on Neural Networks **2** (1990) 163-167.
- [Par85] D. PARKER, *Learning Logic*, Arbeitspapier TR-87, Center for Computational Research in Economics and Management Science, MIT, Cambridge (MA), 1985.
- [Pat96] D. PATTERSON, *Künstliche Neuronale Netze*, Prentice Hall, München, 1996.
- [Pen91] R. PENROSE, *Computerdenken - Des Kaisers neue Kleider oder Die Debatte um Künstliche Intelligenz, Bewußtsein und die Gesetze der Physik*, Spektrum der Wissenschaft, Heidelberg, 1991.
- [Per90] R. PERK, *Professionelle Aktienanalyse für jedermann*, dtv, München, 1990.

- [Pod93] T. PODDING, *Short-Term Forecasting of the USD/DM-Exchange Rate*, in A.N. REFENES (Hrsg.), *Proceedings of the First International Workshop on Neural Networks in the Capital Markets*, London Business School, London, 1993.
- [Pod94] T. PODDING, *Mittelfristige Zinsprognose mittels KNN und ökonometrischer Verfahren*, in H. REHKUGLER und H.G. ZIMMERMANN (Hrsg.), *Neuronale Netze in der Ökonomie*, Verlag Franz Vahlen, München, 1994, 209–289.
- [Pod95] T. PODDING, *Bankruptcy Prediction: A Comparison with Discriminant Analysis*, in A.-P. REFENES (Hrsg.), *Neural Networks in the Capital Markets*, John Wiley & Sons, New York, 1995, 311–323.
- [Pri90] M.J. PRING, *Handbuch Technische Kursanalyse*, Hoppenstedt, Darmstadt, 1990.
- [PT92] W. PIETSCH und A. TEUBNER, *Konnektionistische Tourenplanung für die dezentrale Vertriebsaußendienststeuerung*, in W. GAUL et al. (Hrsg.), *Operation Research Proceedings 1991*, Berlin, 1992, 346–353.
- [PW94] T. PODDING und A. WALLEM, *Wechselkursprognose*, in H. REHKUGLER und H.G. ZIMMERMANN (Hrsg.), *Neuronale Netze in der Ökonomie*, Verlag Franz Vahlen, München, 1994, 291–336.
- [Re^a96] E. RAHIMIAN et al., *Bankruptcy Prediction by Neural Network*, in R.R. TRIPPI und E. TURBAN (Hrsg.), *Neural Networks in Finance and Investigation*, Irwin, 1996.
- [Ree93] R. REED, *Pruning Algorithms - A Survey*, IEEE Transaction on Neural Networks **4** (1993) 740–747.
- [Ref95] A.-P. REFENES, *Neural Networks in the Capital Markets*, John Wiley & Sons, New York, 1995.
- [Reh91] H. REHKUGLER, *Prognose von Wertpapierkursen mit Künstlichen Neuronalen Netzen?*, in L.J. HEINRICH, G. POMBERGER und R. SCHAUER (Hrsg.), *Die Informationswirtschaft im Unternehmen*, Trauner Universitäts-Verlag, Linz, 1991.
- [Reh94] H. REHKUGLER, *Gefährliche Insolvenzprognosen*, Banken & Versicherungen **10** (1994) 38–41.
- [Reh95a] H. REHKUGLER, *Kurs- und Renditeprognose-Systeme*, in J.-E. CRAMER und B. RUDOLF (Hrsg.), *Handbuch für Anlageberatung und Vermögensverwaltung*, Frankfurt, 1995, 384–393.

- [Reh95b] H. REHKUGLER, *Neuronale Netze als Instrument der Jahresabschlußanalyse*, in A.-W. SCHEER (Hrsg.), *Rechnungswesen und EDV - Aus Turbulenzen zum gestärkten Konzept?*, Saarbrücker Arbeitstagung, Heidelberg, 1995, 247–263.
- [RHW86a] D.E. RUMMELHART, G.E. HINTON und R.J. WILLIAMS, *Learning Internal Representations by Error Propagation*, in D.E. RUMMELHART und J.L. MCCELLAND (Hrsg.), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, MIT Press, Cambridge (MA), 1986.
- [RHW86b] D.E. RUMMELHART, G.E. HINTON und R.J. WILLIAMS, *Learning Representations by Back-Propagating Errors*, *Nature* **323** (1986) 533–536.
- [Rie92] M. RIESS, *Ihr Netzwerk denkt mit - Neuronale Netze im Einsatz bei Allianz Lebensversicherung*, *Versicherungsbetriebe* **7/8** (1992) 6–9.
- [Rie94] M. RIESS, *Die Eignung Neuronaler Netze zur Prognose in der Ökonomie*, in G. BOL, G. NAKHAEIZADEH und K.-H. VOLLMER (Hrsg.), *Finanzmarktanwendungen Neuronaler Netze und ökonomische Verfahren*, Physica-Verlag, Heidelberg, 1994, 183–222.
- [RK95] H. REHKUGLER und M. KERLING, *Einsatz Neuronaler Netze für Analyse- und Prognosezwecke*, *Betriebswirtschaftliche Forschung und Praxis* **3** (1995) 306–324.
- [RM93] D. RITTINGHAUS-MEYER, *Die Anwendung Neuronaler Netze in der Marketingforschung*, Akademie Verlag, München, 1993.
- [RMS90] H. RITTER, T. MARTINEZ und K. SCHULTEN, *Neuronale Netze*, Addison-Wesley, Bonn, 1990.
- [Roj94] R. ROJAS, *Was können neuronale Netze?*, in R.-H. SCHULZ (Hrsg.), *Mathematische Aspekte der angewandten Informatik*, BI Wissenschaftsverlag, Mannheim, 1994.
- [Roj96] R. ROJAS, *Neural Networks*, Springer Verlag, Berlin, 1996.
- [Ros58] F. ROSENBLATT, *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*, *Psychological Review* **65** (1958) 386–408.
- [RP90a] H. REHKUGLER und T. PODDING, *Entwicklung leistungsfähiger Prognosesysteme auf Basis Künstlicher Neuronaler Netzwerke am Beispiel des Dollars - Eine Fallstudie*, *Bamberger Betriebswissenschaftliche Beiträge* 76, Universität Bamberg, 1990.

- [RP90b] H. REHKUGLER und T. PODDING, *Statistische Methoden versus Künstliche Neuronale Netzwerke zur Aktienkursprognose - Eine vergleichende Studie*, Bamberger Betriebswissenschaftliche Beiträge 73, Universität Bamberg, 1990.
- [RP91] H. REHKUGLER und T. PODDING, *Künstliche Neuronale Netze in der Finanzanalyse: Eine neue Ära der Kursprognosen?*, Wirtschaftsinformatik **5** (1991) 365-374.
- [RP92a] H. REHKUGLER und T. PODDING, *Klassifikation von Jahresabschlüssen mittels Multilayer-Perceptrons - Erste Ergebnisse und weiterführende Fragestellungen*, Bamberger Betriebswissenschaftliche Beiträge 87, Universität Bamberg, 1992.
- [RP92b] H. REHKUGLER und T. PODDING, *Neuronale Netze im Bankbetrieb*, Die Bank **7** (1992) 413-419.
- [RP94] H. REHKUGLER und T. PODDING, *Kurzfristige Wechselkursprognosen mit Künstlichen Neuronalen Netzwerken*, in G. BOL, G. NAKHAEIZADEH und K.-H. VOLLMER (Hrsg.), *Finanzmarktanwendungen neuronaler Netze und ökonomische Verfahren*, Physica-Verlag, Heidelberg, 1994, 1-24.
- [RS92] A. RUPPEL und J. SIEDENTOPF, *Konnektionistisch motivierte Reihenfolgeplanung im Fertigungsleistständen*, in W. GÖRKE et al. (Hrsg.), *Information als Produktionsfaktor*, Springer Verlag, Berlin, 1992, 554-563.
- [RS94] R. RETZKO und M. SCHUHMAN, *Lösungsmöglichkeiten von Tourenplanungsproblemen mit Selbstorganisierenden Neuronalen Netzen*, Arbeitspapier 11, Georg-August-Universität Göttingen, Abteilung Wirtschaftsinformatik II, 1994.
- [RSR91] W. RAGHUPATHI, L.L. SCHKADE und B.S. RAJU (Hrsg.), *A Neural Network Approach to Bankruptcy Prediction*, IEEE, 24th Annual Hawaii International Conference on Systems Science, 1991.
- [Rug94] M.A. RUGGIERO JR., *Training Neural Nets for Intermarket Analysis*, Futures **Aug** (1994) 42-44.
- [RvR93] H. REHKUGLER und A. SCHMIDT VON RHEIN, *Kreditwürdigkeitsanalyse und -prognose für Privatkundenkredite mittels statistischer Methoden und Künstlicher Neuronaler Netze - Eine empirisch-vergleichende Studie*, Bamberger Betriebswissenschaftliche Beiträge 93, Universität Bamberg, 1993.

- [Sch90] E. SCHÖNEBURG, *Stock Price Prediction Using Neural Networks*, *Neurocomputing* **2** (1990) 17-27.
- [Sch93a] B. SCHNEIDER, *Neuronale Netze für betriebliche Anwendungen: Anwendungspotentiale und existierende Systeme*, Arbeitsbericht 22, Universität Münster, 1993.
- [Sch93b] E. SCHÖNEBURG, *Industrielle Anwendung Neuronaler Netze*, Addison-Wesley, Bonn, 1993.
- [SD91] J. SIETSMA und R. DOW, *Creating Artificial Neural Networks that Generalize*, *Neural Networks* **4** (1991) 67-79.
- [SGR89] E. SCHÖNEBURG, M. GANTERT und M. REINER, *Aktienkursprognose mit Neuronalen Netzwerken*, *Computerwoche* **40** (1989) 121-124.
- [SHG90] E. SCHÖNEBURG, N. HANSEN und A. GAWELCZYK, *Neuronale Netze*, Markt und Technik, Haar bei München, 1990.
- [SL93] M. SCHUHMANN und T. LOHRBACH, *Comparing Artificial Neural Networks with Statistical Methods within the Field of Stock Market Prediction*, *International Joint Conference on System Science* (1993) 597-606.
- [SL94] M. SCHUHMANN und T. LOHRBACH, *Kurzfristige Aktienkursprognose - Vergleich Künstlicher Neuronaler Netze und statistischer Verfahren*, in G. BOL, G. NAKHAEIZADEH und K.-H. VOLLMER (Hrsg.), *Finanzmarktanwendungen neuronaler Netze und ökonomische Verfahren*, Physica-Verlag, Heidelberg, 1994, 247-269.
- [SLB92] M. SCHUHMANN, T. LOHRBACH und P. BÄHRS, *Versuche zur Kreditwürdigkeitsprognose mit Künstlichen Neuronalen Netzen*, Arbeitspapier 2, Universität Göttingen, Januar 1992.
- [SN93] E. SCHÖNEBURG und NICKOLAY, *Metalevel*, unveröffentlicht, 1993.
- [SS97] S. STRECKER und A.C. SCHWICKERT, *Künstliche Neuronale Netze - Einordnung, Klassifikation und Abgrenzung aus betriebswirtschaftlicher Sicht*, Arbeitspapier WI 4, Universität Mainz, 1997.
- [Ste93] R. STEIN, *Selecting Data for Neural Networks*, *AI Expert* **2** (1993) 42-47.
- [SW93] M. STEINER und H.-G. WITTKEMPER, *Neuronale Netze: Ein Hilfsmittel für betriebswirtschaftliche Probleme*, *DBW* **53** (1993) 447-463.

- [TK90] K.Y. TAM und M.Y. KIANG, *Predicting Bank Failures: A Neural Network Approach*, Applied Artificial Intelligence **4** (1990) 265-282.
- [TK91] K.Y. TAM und M.Y. KIANG, *Neural Networks Models and the Prediction of Bank Bankruptcy*, OMEGA, International Journal of Management Science **19** (1991) 429-445.
- [TK92] K.Y. TAM und M.Y. KIANG, *Managerial Applications of Neural Networks: The Case of Bank Failure Predictions*, Management Science **38** (1992) 926-947.
- [TT96] R.R. TRIPPI und E. TURBAN, *Neural Networks in Finance and Investing*, Irwin, Chicago, 1996.
- [Vog97] R. VOGL, *Aktien*, Verlag R. Vogl, Fürth, 1997.
- [vRR94] A. SCHMIDT VON RHEIN und H. REHKUGLER, *KNN zur Kreditwürdigkeitsprüfung bei Privatkundenkrediten*, in H. REHKUGLER und H.G. ZIMMERMANN (Hrsg.), *Neuronale Netze in der Ökonomie*, Verlag Franz Vahlen, München, 1994, 491-545.
- [WdG92] D. WÜRTZ und C. DE GROOT, *Ein neuronales Informationssystem zur Wechselkursanalyse*, Intelligente Software-Technologien **2** (1992) 44-46.
- [Wel78] J. WELLES WILDER, *New Concepts in Technical Trading Systems*, Trend Research, Greensboro (NC), 1978.
- [Wel91] J. WELCKER, *Technische Aktienanalyse*, Verlag Moderne Industrie, Zürich, 1991.
- [Wer74] P.J. WERBOS, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Dissertation, Harvard University, 1974.
- [WH60] B. WIDROW und M.E. HOFF, *Adaptive Switching Circuits*, 1960 IRE WESCON Convention Record, New York, 1960.
- [WHR92] A.S. WEIGEND, B.A. HUBERMANN und D.E. RUMMELHART, *Predicting Sunspots and Exchange Rates with Connectionist Networks*, in E. CASDAGLI (Hrsg.), *Nonlinear Modelling and Forecasting - SFI Studies in the Science of Complexity*, Addison-Wesley, Reading (MA), 1992, 395-432.
- [Wil91a] R. WILBERT, *Kreditwürdigkeitsanalyse im Konsumentenkreditgeschäft auf der Basis Neuronaler Netze*, Zeitschrift für Betriebswirtschaft **61** (1991) 1377-1393.

- [Wil91b] K.-D. WILD, *Neuronale Netze erstellen leistungsfähige Zinsprognosen*, Sparkasse **6** (1991) 257-260.
- [Wil94] K.-D. WILD, *Zinsprognose und Rentenmanagement*, Sparkasse **3** (1994) 127-130.
- [WL90] B. WIDROW und M.A. LEHR, *30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation*, IEEE Proceedings **78** (1990) 1415-1441.
- [WRH91] A.S. WEIGEND, D.E. RUMMELHART und B.A. HUBERMANN, *Generalization by Weight-Elimination with Application to Forecasting*, in R. LIPPMANN, J. MOODY und D. TOURETZKY (Hrsg.), *Advances in Neural Information Processing Systems*, Morgan Kaufmann, San Francisco, 1991, 875-882.
- [WS94] R.L. WILSON und R. SHARDA, *Bankruptcy Prediction Using Neural Networks*, Decision Support Systems **11** (1994) 545-557.
- [WT81] J. WELCKER und E. THOMAS, *Finanzanalyse*, Verlag Franz Vahlen, München, 1981.
- [Yod94] M. YODA, *Predicting the Tokyo Stock Market*, in G.J. DEBOECK (Hrsg.), *Trading on the Edge - Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets*, John Wiley & Sons, New York, 1994, 66-79.
- [Z^{ea}95] A. ZELL et al., *Stuttgarter Neural Network Simulator User Manual Version 4.1*, Institut für Parallele und Verteilte Höchstleistungsrechner, Universität Stuttgart, 1995.
- [Zel94] A. ZELL, *Simulation Neuronaler Netze*, Addison-Wesley, Bonn, 1994.
- [ZHF92] H.G. ZIMMERMANN, F. HERGET und W. FINNOFF, *Neuron Pruning and Merging Methods for Use in Conjunction with Weight Elimination*, Arbeitspapier, Siemens AG, Corporate Research and Development, München, 1992.

A Graphische Darstellungen der Eingabedaten

Die Eingabedaten der DAX-Prognose lassen sich im Rahmen der Intemarket-Analyse in die Sektoren Aktienindizes, Devisen, Renten und Rohstoffe untergliedern. Innerhalb dieser Sektoren wird zumeist noch zwischen den Märkten Nordamerika, Asien/Pazifik und Europa unterschieden. Außerdem werden auch monetäre Größen betrachtet.

A.1 Aktienindizes

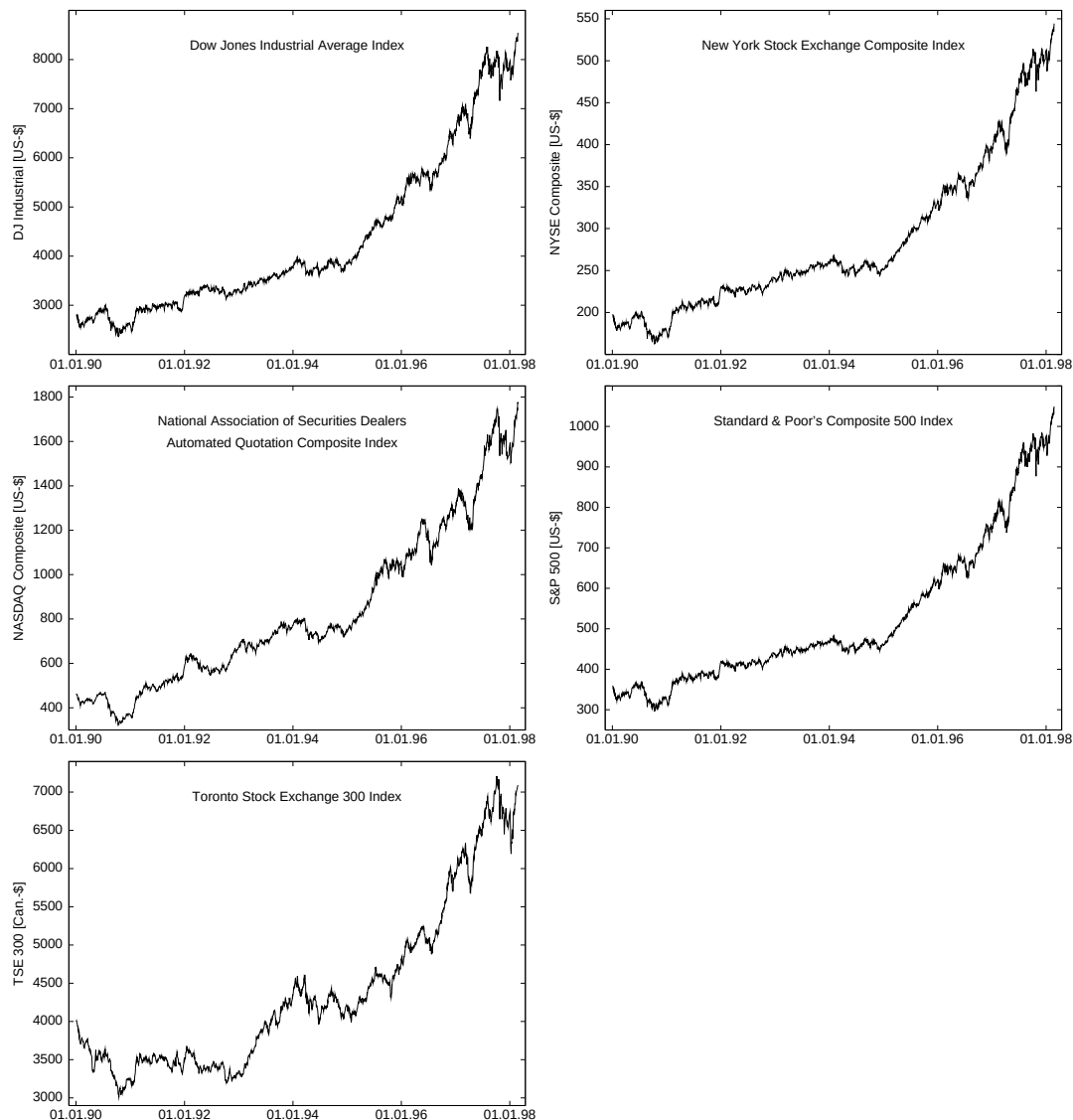


Abbildung A.1: Zeitlicher Verlauf der Aktienindizes Nordamerikas

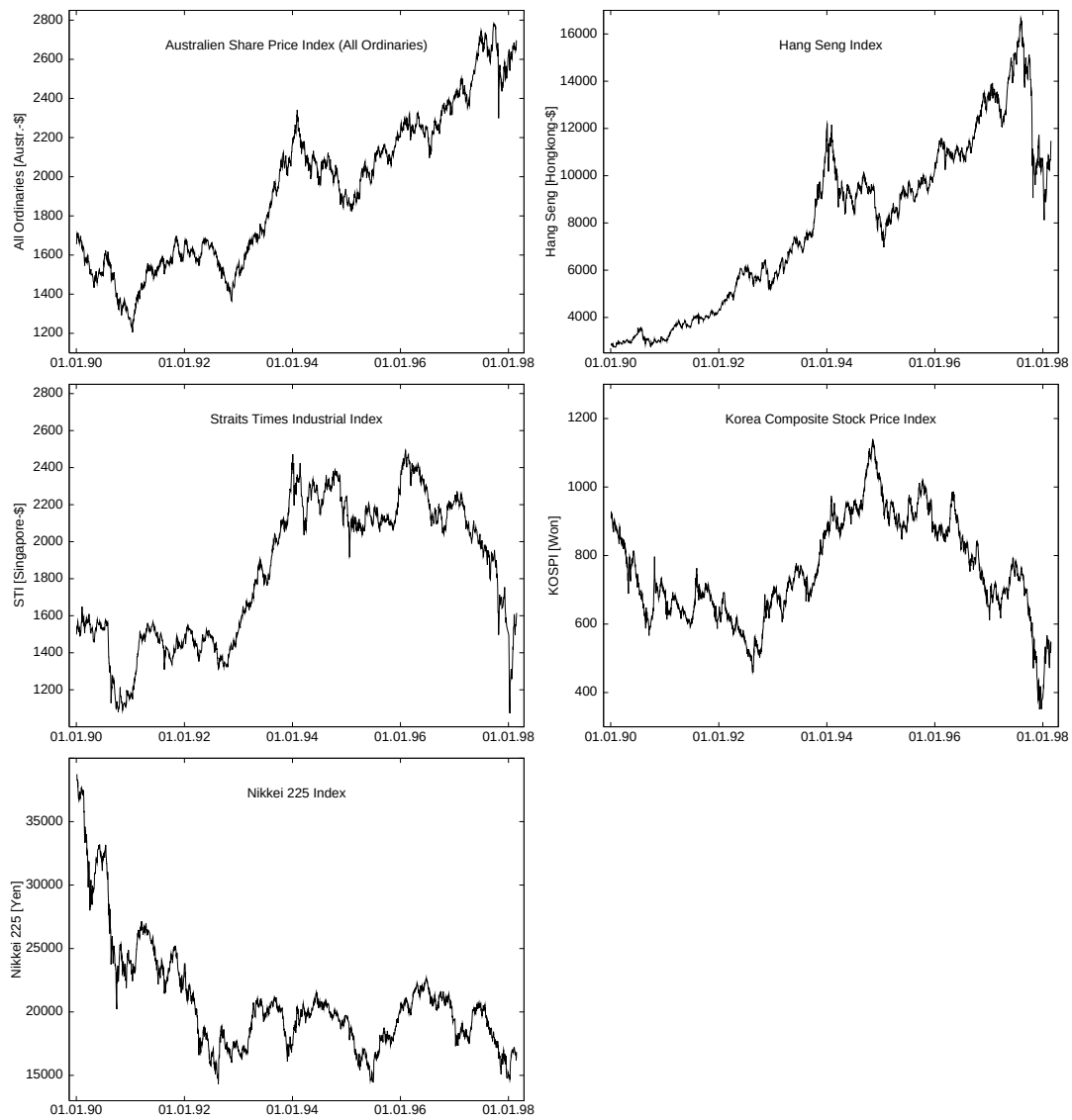


Abbildung A.2: Zeitlicher Verlauf der Aktienindizes Asien/Pazifik

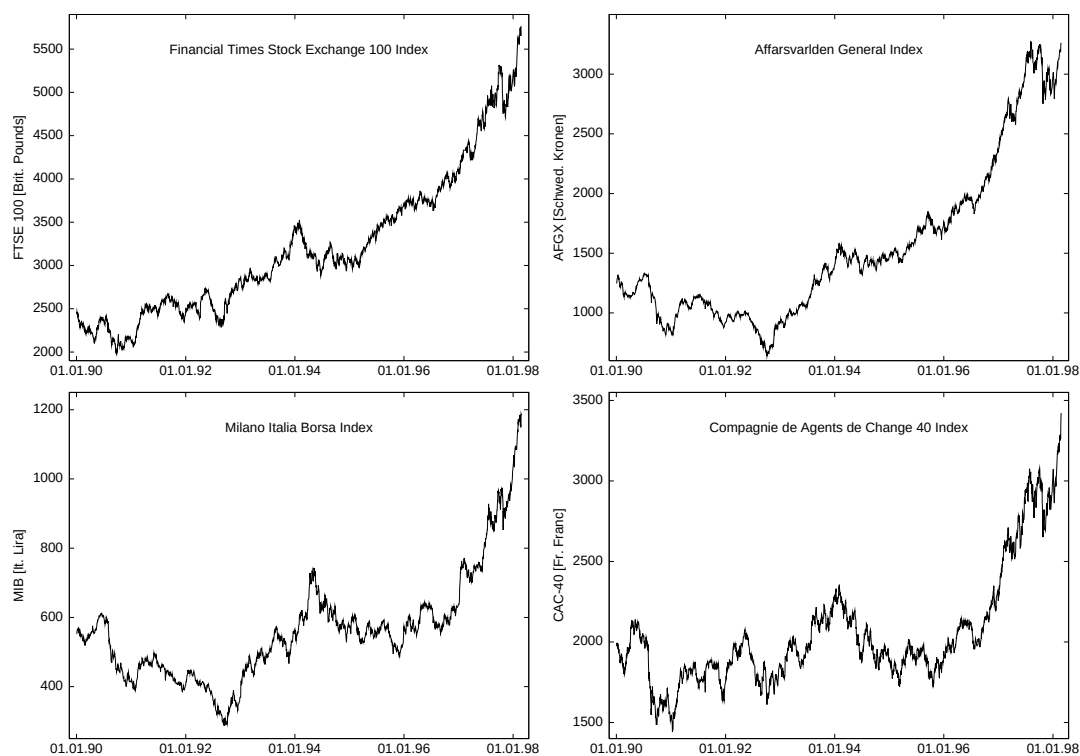


Abbildung A.3: Zeitlicher Verlauf der Aktienindizes Europas

A.2 Devisen

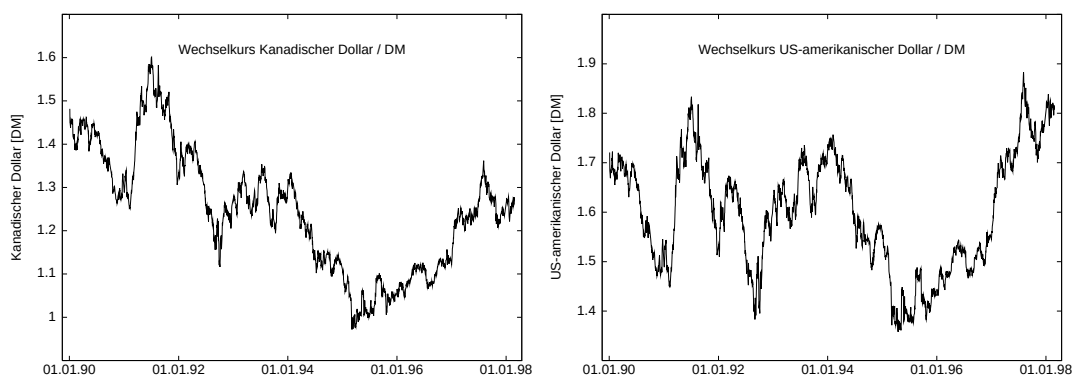


Abbildung A.4: Zeitlicher Verlauf der Wechselkurse Nordamerikas

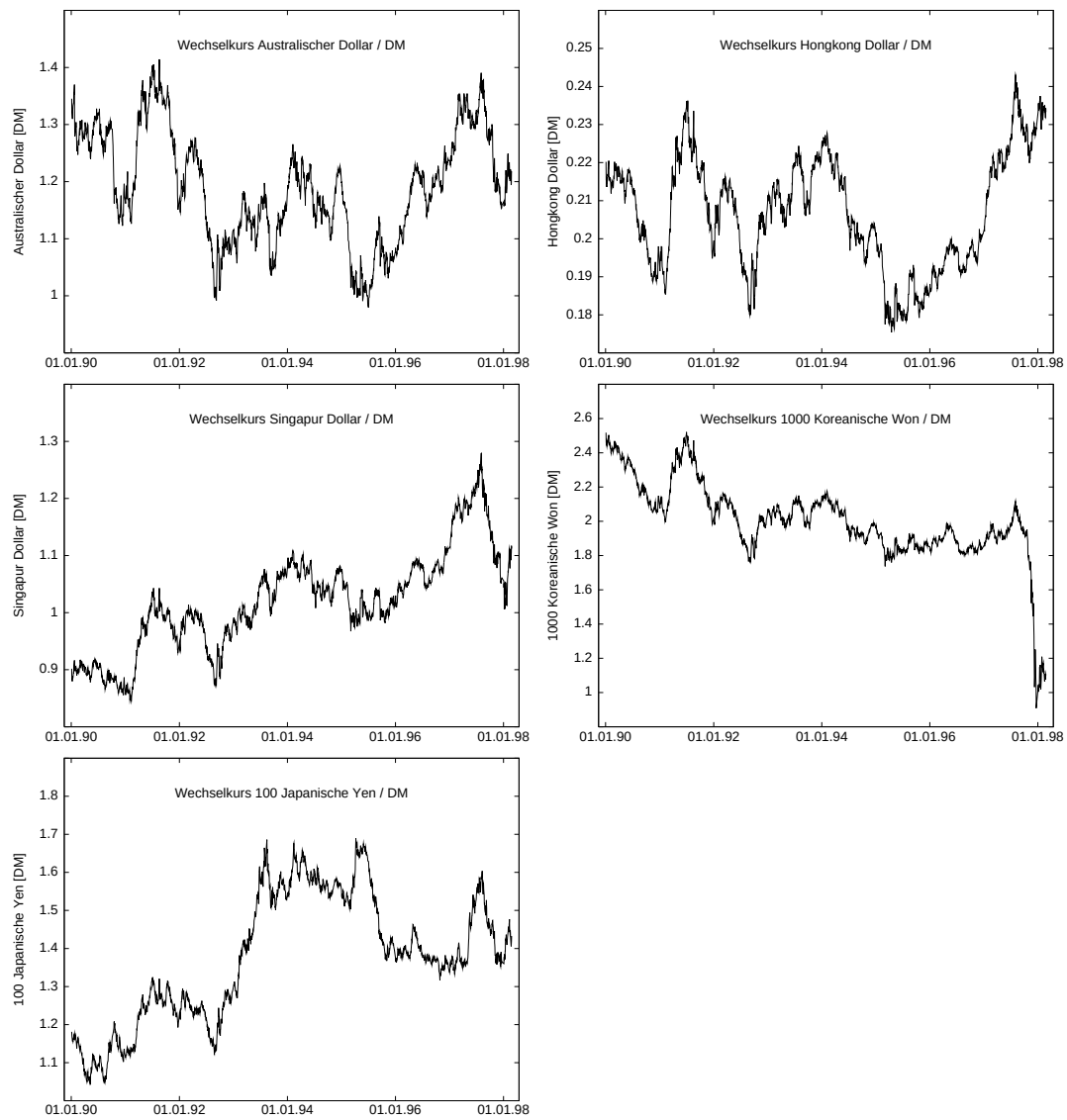


Abbildung A.5: Zeitlicher Verlauf der Wechselkurse Asien/Pazifik

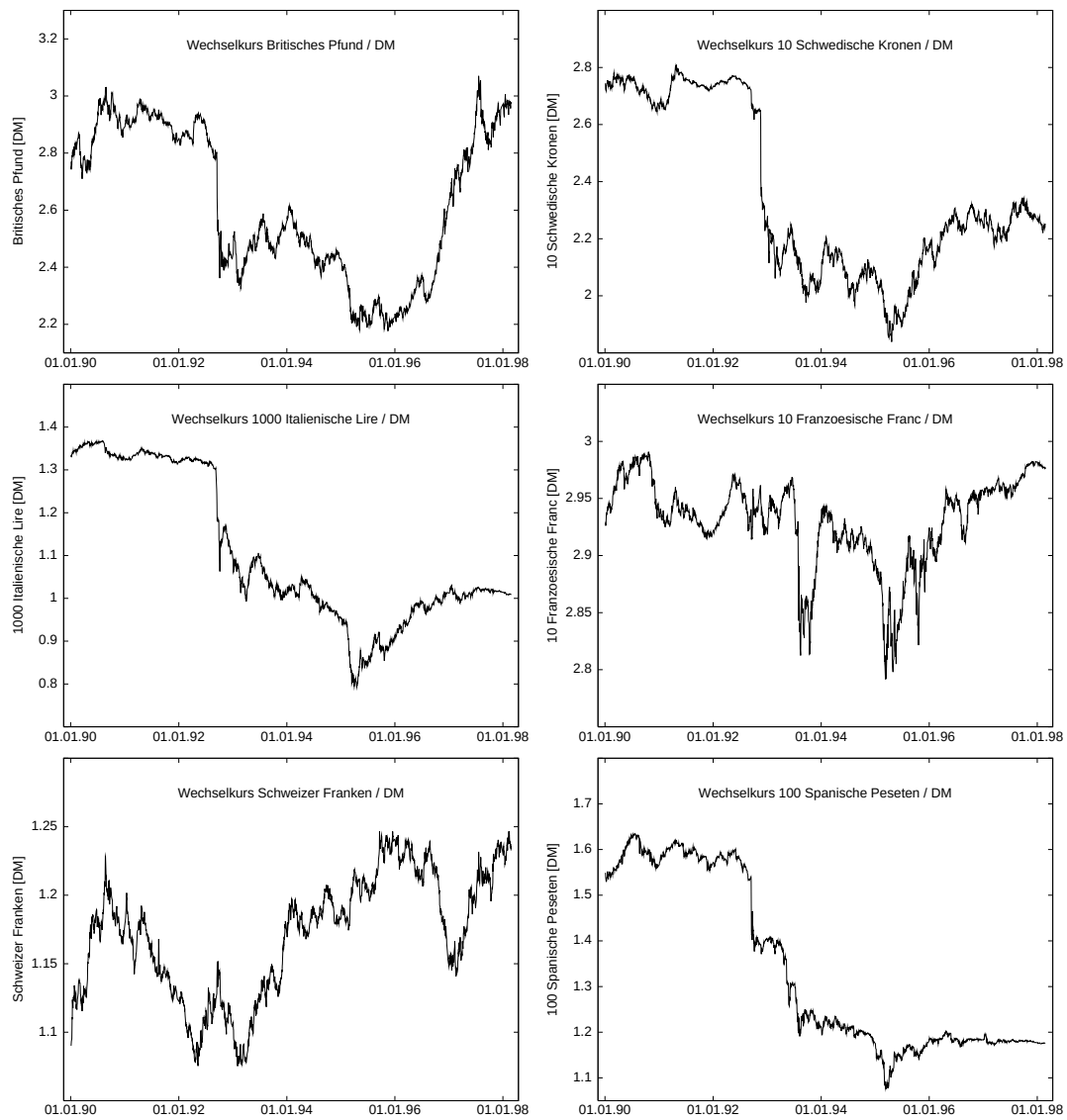


Abbildung A.6: Zeitlicher Verlauf der Wechselkurse Europas

A.3 Renten

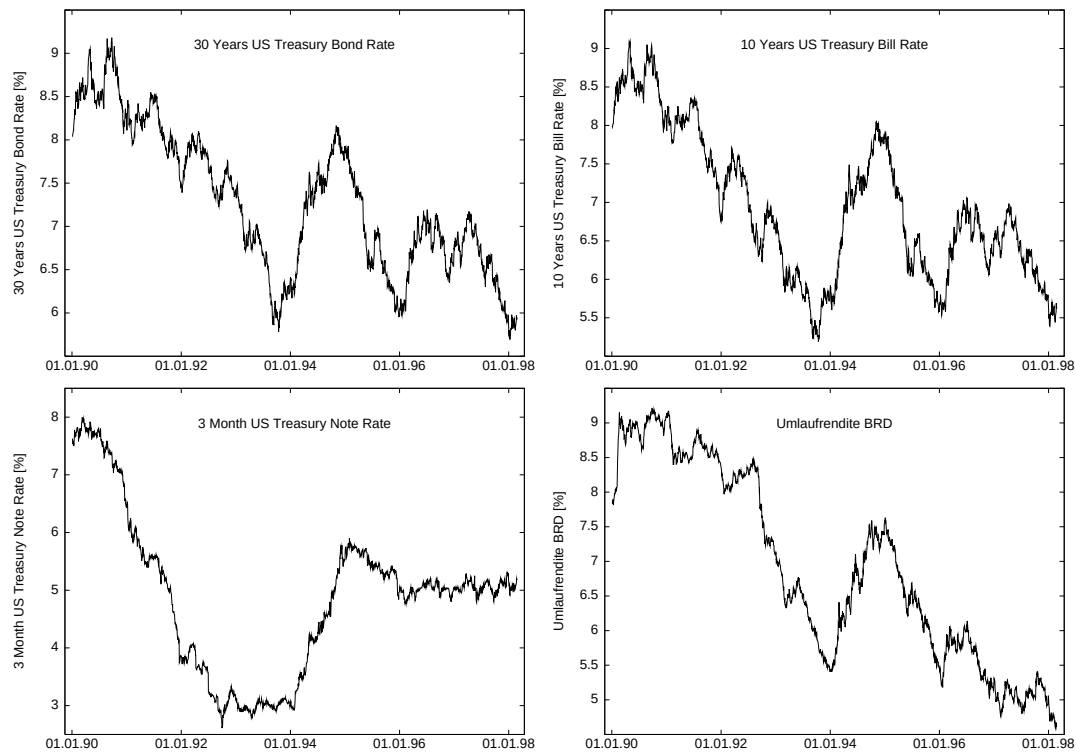


Abbildung A.7: Zeitlicher Verlauf der Zinssätze der Renten von USA und BRD

A.4 Rohstoffe

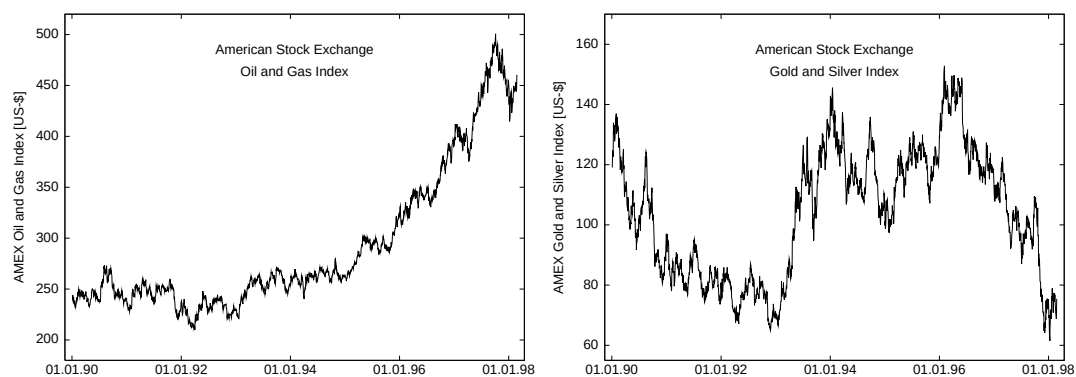


Abbildung A.8: Zeitlicher Verlauf der Rohstoff-Indizes

A.5 Monetäre Größen

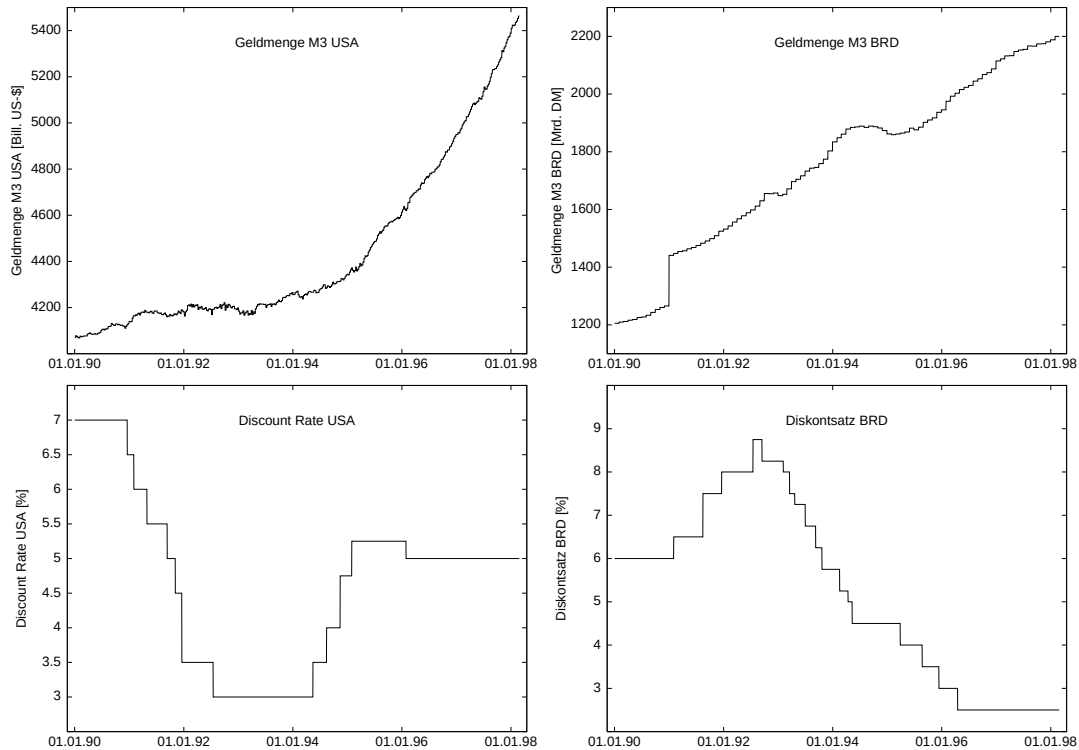


Abbildung A.9: Zeitlicher Verlauf von Geldmenge M3 und Diskontsatz für USA und BRD⁴¹

A.6 Handelsvolumen

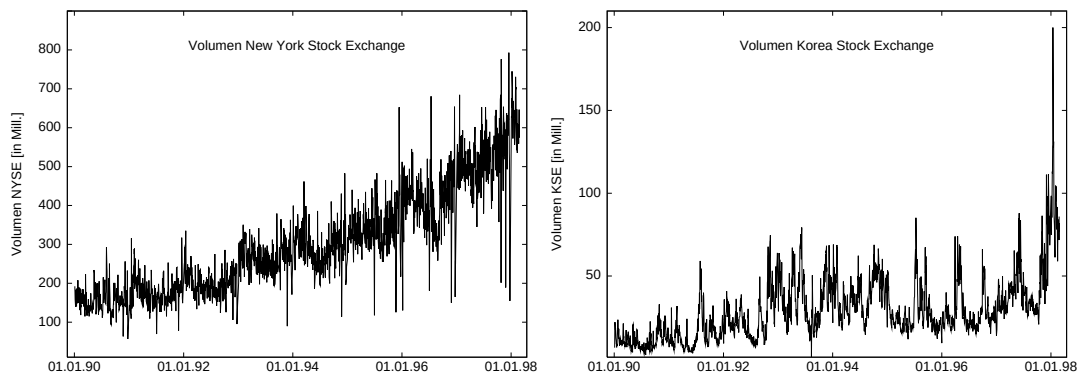


Abbildung A.10: Zeitlicher Verlauf der Volumen an gehandelten Aktien an den Börsen von New York und Seoul

⁴⁰Der Sprung in der Geldmenge M3 der BRD am 1.1.1991 kommt daher, daß ab diesem Zeitpunkt die fünf neuen Länder (ehemals DDR) zur Geldmenge zugerechnet wurden.

B Suche nach historischen Zeitreihen

Es gibt zahlreiche Anbieter von historischen Börsen- und Wirtschaftsdaten meist in Verbindung mit Aktienanalyse-Softwareprogrammen, deren Leistungen i.d.R. nicht kostenfrei sind. Durch Online-Recherchen im World Wide Web (WWW oder W3) des Internets ist es mir aber gelungen, Seiten zu finden, die historische Kurse enthalten oder zumindest weitere Hypertext-Links zu anderen Seiten enthalten, deren Abfrage kostenlos ist. Die dazugehörigen Uniform Resource Locators (URLs) bzw. Hypertext-Links lauten:

NMZF - Historische Kurse	http://www-public.tu-bs.de:8080/~y0003876/index.html
NYSE Statistics Archive	http://www.nyse.com/public/market/2c/2c2/2c2fm.htm
Pacific Exchange Rate Service	http://pacific.commerce.ubc.ca/xr/data.html
Korea Stock Exchange	http://www.kse.or.kr/stat/index.html
FRED U.S. Financial Data	http://www.stls.frb.org/fred/data/wkly.html
Neuro Stock	http://neurostock.com/histdata.html
Free Historical Data	http://www.fortitude.com/data.htm
Moby Data	http://www.mobydata.com/
Quote.Com Inc.	http://www.quote.com/
Deutsche Bundesbank	http://www.bundesbank.de/

Desweiteren existieren (hoffentlich noch immer) einige Links zu Seiten mit Lesezeichen (engl. bookmarks) zu dieser Thematik:

The Financial Data Finder	http://www.cob.ohio-state.edu/dept/fin/osudata.htm
Barra Financial Links	http://www.barra.com/ResearchResources/OtherLinks.asp
Capial PC User Group	http://cpcug.org/user/invest/datahist.html
Börsen-Links	http://www.phone-soft.de/cyber-world/0348d.htm
Yahoo -Finance	http://quote.yahoo.com/
Dino - Finanzen - Aktienmarkt	http://www.dino-online.de/seiten/go13faa.htm
FINWeb	http://www.finweb.com/

Außerdem haben mir die folgenden Firmen geholfen, Lücken in den Zeitreihen zu schliessen:

Lenz + Partner GmbH	http://www.lp-software.de/
Neue Wirtschaftspresse Medien GmbH	http://www.nwp.de/
Market Marker Software GmbH	http://www.market-maker.de/

C Korrelationsanalyse

Das Fortran 77-Programm CORANL berechnet paarweise die linearen Korrelationskoeffizienten der vorliegenden Datenreihen und zwar vom ersten bis zum K -ten Wert der Datenreihen. Anschließend werden Start- und Endwerte um Eins erhöht und die Berechnung erneut ausgeführt. Dies erfolgt solange, bis der Endwert der Berechnung mit dem Endwert der Datenreihe übereinstimmt. Von den so berechneten Korrelationskoeffizienten wird schließlich der arithmetische Mittelwert gebildet und ausgegeben.

Die Korrelationsanalyse wird für die Parameter $K = \{10, 38, 70, 400\}$ durchgeführt. Dies entspricht den Zeiträumen von etwa zwei Wochen, zwei Monaten, dem Vorhersagezeitraum (1.10.1997 bis 28.2.1998) und zwei Jahren. Bei der Untersuchung der 41 Zeitreihen hat sich gezeigt, daß nur hohe Korrelationen innerhalb der einzelnen Sektoren der Intermarket-Analyse auftreten. Aus diesem Grund und wegen der besseren Übersicht und Darstellung ist es daher sinnvoll, die Sektoren Aktienindizes, Devisen und Sonstiges (Renten, Rohstoffe, monetäre Größen und Handelsvolumen) separat zu betrachten. Es werden nur die Daten zur Korrelationsanalyse herangezogen, die auch tatsächlich vor der ersten Vorhersage vorliegen, d.h. $N = 1702$. Je nach Eingabedatei (Aktienindizes, Devisen, Sonstiges) muß des Parameter $M = \{15, 13, 13\}$ und je nach zeitlicher Untersuchungsdauer der Parameter K (s.o.) neu eingestellt werden und das Programm vor der Ausführung neu compiliert werden.

C.1 Quellprogramm in FORTRAN 77

```

      PROGRAM CORANL
C*****
C
C      correlation analysis of data time series
C
C      author: S. Selle, 16.04.1998
C
C      CHECK = .true. for normal end, .false. otherwise
C      NINP  = number of input file
C      NOUT  = number of output file
C      N     = number of data points
C      M     = number of data series
C      K     = number of data points for analysis
C      X(N,M) = input data
C      V(M,M) = correlation coefficients
C      X1(K),X2(K),X3(K),HELP,I,J,II,JJ = auxiliary variables
C
C*****

```

```

C*****
C      type declarations                                     *
C*****
      IMPLICIT NONE
      INTEGER NINP,NOUT,N,M,K,I,J,II,JJ
      PARAMETER (NINP=3,NOUT=6,N=1702,M=13,K=38)
      DOUBLE PRECISION X(N,M),V(M,M),X1(K),X2(K),X3(K),HELP
      LOGICAL CHECK
C*****
C      start of program -CORANL-                             *
C*****
      WRITE(NOUT,900)
      WRITE(NOUT,901)
      WRITE(NOUT,900)
      WRITE(NOUT,902) N
      WRITE(NOUT,903) K
      IF (K.GT.N) THEN
        WRITE(NOUT,997) K,N
        WRITE(NOUT,999)
        GOTO 1000
      ENDIF
      WRITE(NOUT,900)
C*****
C      read input data                                       *
C*****
      CALL INPDAT(NINP,N,M,X,CHECK)
      IF (.NOT.CHECK) THEN
        WRITE(NOUT,998) K,N
        WRITE(NOUT,999)
        GOTO 1000
      ENDIF
C*****
C      calculation of correlation coefficients                 *
C*****
      DO I=1,M
        DO J=1,I-1
          V(I,J)=0.0D0
          DO II=1,N-K+1
            DO JJ=1,K
              X1(JJ)=X(II+JJ-1,I)
              X2(JJ)=X(II+JJ-1,J)
            ENDDO
            CALL COEF(K,X1,X2,X3,HELP)
            V(I,J)=V(I,J)+HELP
          ENDDO
          V(I,J)=V(I,J)/DBLE(N-K+1)
          V(J,I)=V(I,J)
        ENDDO
      ENDDO

```



```

C*****
C    print output: table of correlation coefficients          *
C*****
      WRITE(NOUT,904) (I,I=1,M-1)
      DO I=2,M
        WRITE(NOUT,905) I,(V(I,J),J=1,I-1)
      ENDDO
      WRITE(NOUT,900)
C*****
C    print output: correlation coefficients > 0.85          *
C*****
      DO I=1,M
        DO J=1,I-1
          IF (V(I,J).GT.0.85D0) THEN
            WRITE(NOUT,906) J,I,V(I,J)
          ENDIF
        ENDDO
      ENDDO
      WRITE(NOUT,900)
      WRITE(NOUT,910)
C*****
C    format statements          *
C*****
900  FORMAT(80('*'))
901  FORMAT('*    CORRELATION ANALYSIS OF DATA TIME SERIES --- CORRELAT
      FION COEFFICIENTS    *')
902  FORMAT(5X,'DATA POINTS of TIME SERIES :          ',I5)
903  FORMAT(5X,'DATA POINTS of CORRELATION ANALYSIS : ',I5)
904  FORMAT(2X,100(1X,I5))
905  FORMAT(I2,100(1X,F5.2))
906  FORMAT(5X,I2,' and ',I2,' : ',F7.4)
910  FORMAT(5X,'PROGRAM END')
997  FORMAT(5X,'ERROR in -CORANL- : K > N : ',I5,' > ',I5)
998  FORMAT(5X,'ERROR in -CORANL- : wrong output from -INPDAT-')
999  FORMAT(5X,'PROGRAM TERMINATED')
C*****
C    end of program -CORANL-          *
C*****
1000 END

```

```

      SUBROUTINE MEAN(N,X,V)
C*****
C      mean value V of N data points X(N)
C*****
      IMPLICIT NONE
      INTEGER N,I
      DOUBLE PRECISION X(N),V
      V=0.0D0
      DO I=1,N
        V=V+X(I)
      ENDDO
      V=V/DBLE(N)
      RETURN
      END

```

```

      SUBROUTINE VAR(N,X,Z,V)
C*****
C      variance V of N data points X(N)
C*****
      IMPLICIT NONE
      INTEGER N,I
      DOUBLE PRECISION X(N),Z(N),V,HELP
      DO I=1,N
        Z(I)=X(I)*X(I)
      ENDDO
      CALL MEAN(N,Z,V)
      CALL MEAN(N,X,HELP)
      V=V-HELP*HELP
      RETURN
      END

```

```

      SUBROUTINE COV(N,X,Y,Z,V)
C*****
C      covariance V of N data points X(N) and Y(N)
C*****
      IMPLICIT NONE
      INTEGER N,I
      DOUBLE PRECISION X(N),Y(N),Z(N),V,HELP
      CALL MEAN(N,X,V)
      CALL MEAN(N,Y,HELP)
      HELP=HELP*V
      DO I=1,N
        Z(I)=X(I)*Y(I)
      ENDDO
      CALL MEAN(N,Z,V)
      V=V-HELP
      RETURN
      END

```

```

      SUBROUTINE COEF(N,X,Y,Z,V)
C*****
C      correlation coefficients V of N data points X(N) and Y(N)      *
C*****
      IMPLICIT NONE
      INTEGER N
      DOUBLE PRECISION X(N),Y(N),Z(N),V,HELP,HELP2
      CALL COV(N,X,Y,Z,V)
      CALL VAR(N,X,Z,HELP)
      CALL VAR(N,Y,Z,HELP2)
      IF ((HELP2*HELP).GT.0.0D0) THEN
        V=V/DSQRT(HELP*HELP2)
      ELSE
        V=0.0D0
      ENDIF
      RETURN
      END

      SUBROUTINE INPDAT(NINP,N,M,X,CHECK)
C*****
C      read input data from file      *
C*****
      IMPLICIT NONE
      INTEGER NINP,N,M,I,J
      DOUBLE PRECISION X(N,M)
      LOGICAL CHECK
      CHECK=.TRUE.
      OPEN(NINP,ERR=100)
      REWIND NINP
      DO I=1,N
        READ(NINP,*,END=100) (X(I,J),J=1,M)
      ENDDO
      GOTO 200
100  CONTINUE
      CHECK=.FALSE.
200  CLOSE(NINP)
      RETURN
      END

```

C.2 Korrelationskoeffizienten

Aktienindizes

```
(1) = TSE 300
(2) = DJ Industrial
(3) = NYSE Composite
(4) = NASDAQ Composite
(5) = S&P 500
(6) = All Ordinaries
(7) = Nikkei 225
(8) = STI
(9) = Hang Seng
(10) = KOSPI
(11) = FTSE 100
(12) = AFGX
(13) = MIB
(14) = CAC-40
(15) = DAX
```

```
*****
*   CORRELATION ANALYSIS OF DATA TIME SERIES --- CORRELATION COEFFICIENTS   *
*****
      DATA POINTS of TIME SERIES :           1702
      DATA POINTS of CORRELATION ANALYSIS :      10
*****
      1      2      3      4      5      6      7      8      9      10      11      12      13      14
2  0.60
3  0.62  0.89
4  0.57  0.66  0.77
5  0.60  0.90  0.99  0.76
6  0.39  0.44  0.44  0.32  0.43
7  0.21  0.25  0.26  0.18  0.26  0.31
8  0.25  0.30  0.29  0.24  0.28  0.30  0.24
9  0.30  0.34  0.36  0.29  0.34  0.37  0.22  0.37
10 0.13  0.02  0.03  0.07  0.03  0.10  0.07  0.10  0.12
11 0.31  0.36  0.40  0.33  0.39  0.34  0.24  0.23  0.33  0.11
12 0.30  0.37  0.38  0.37  0.37  0.34  0.21  0.26  0.32  0.12  0.42
13 0.21  0.28  0.29  0.25  0.28  0.25  0.22  0.26  0.23  0.07  0.33  0.34
14 0.22  0.31  0.32  0.24  0.31  0.26  0.24  0.22  0.28  0.06  0.54  0.42  0.40
15 0.31  0.42  0.41  0.35  0.41  0.37  0.24  0.27  0.34  0.05  0.51  0.48  0.41  0.57
*****
      2 and 3 : 0.8902
      2 and 5 : 0.8951
      3 and 5 : 0.9903
*****
```

```
*****
*   CORRELATION ANALYSIS OF DATA TIME SERIES --- CORRELATION COEFFICIENTS   *
*****
```

DATA POINTS of TIME SERIES : 1702

DATA POINTS of CORRELATION ANALYSIS : 38

```
*****
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	0.63													
3	0.64	0.89												
4	0.58	0.67	0.80											
5	0.62	0.90	0.99	0.78										
6	0.52	0.51	0.52	0.39	0.50									
7	0.24	0.24	0.27	0.17	0.25	0.37								
8	0.36	0.34	0.34	0.28	0.34	0.40	0.27							
9	0.39	0.47	0.45	0.36	0.45	0.42	0.12	0.44						
10	0.13	0.04	0.05	0.11	0.05	0.13	0.17	0.17	0.13					
11	0.45	0.49	0.55	0.48	0.54	0.46	0.22	0.31	0.42	0.17				
12	0.41	0.48	0.47	0.42	0.45	0.42	0.21	0.28	0.36	0.09	0.47			
13	0.28	0.25	0.28	0.28	0.26	0.28	0.25	0.30	0.21	0.08	0.35	0.44		
14	0.34	0.39	0.40	0.30	0.39	0.36	0.22	0.25	0.30	0.07	0.56	0.44	0.44	
15	0.38	0.47	0.45	0.36	0.44	0.41	0.16	0.33	0.39	-0.01	0.55	0.52	0.43	0.61

```
*****
```

2 and 3 : 0.8915

2 and 5 : 0.8971

3 and 5 : 0.9922

```
*****
```

```
*   CORRELATION ANALYSIS OF DATA TIME SERIES --- CORRELATION COEFFICIENTS   *
*****
```

DATA POINTS of TIME SERIES : 1702

DATA POINTS of CORRELATION ANALYSIS : 70

```
*****
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	0.66													
3	0.66	0.89												
4	0.61	0.69	0.81											
5	0.64	0.90	0.99	0.79										
6	0.60	0.58	0.55	0.44	0.53									
7	0.20	0.16	0.20	0.15	0.17	0.28								
8	0.41	0.36	0.34	0.31	0.33	0.43	0.22							
9	0.45	0.58	0.53	0.44	0.53	0.50	0.07	0.46						
10	0.08	0.02	0.02	0.13	0.02	0.08	0.21	0.13	0.06					
11	0.48	0.58	0.63	0.53	0.62	0.55	0.17	0.32	0.51	0.14				
12	0.47	0.56	0.51	0.47	0.50	0.47	0.17	0.27	0.36	0.08	0.53			
13	0.34	0.29	0.31	0.34	0.29	0.31	0.31	0.34	0.20	0.06	0.36	0.53		
14	0.37	0.44	0.44	0.35	0.43	0.42	0.22	0.26	0.34	0.03	0.58	0.48	0.49	
15	0.44	0.51	0.46	0.36	0.45	0.46	0.13	0.32	0.39	-0.04	0.56	0.59	0.52	0.64

```
*****
```

2 and 3 : 0.8923

2 and 5 : 0.8976

3 and 5 : 0.9935

```
*****
```

```

*****
*   CORRELATION ANALYSIS OF DATA TIME SERIES --- CORRELATION COEFFICIENTS   *
*****
      DATA POINTS of TIME SERIES :           1702
      DATA POINTS of CORRELATION ANALYSIS :    400
*****
      1      2      3      4      5      6      7      8      9      10     11     12     13     14
2  0.73
3  0.71  0.94
4  0.71  0.90  0.96
5  0.69  0.95  1.00  0.95
6  0.80  0.70  0.68  0.67  0.65
7  0.26 -0.09 -0.08 -0.07 -0.10  0.22
8  0.53  0.41  0.39  0.38  0.37  0.55  0.38
9  0.63  0.79  0.77  0.73  0.75  0.76 -0.04  0.53
10 0.06 -0.08 -0.11 -0.02 -0.09 -0.02  0.30  0.37 -0.09
11 0.72  0.79  0.85  0.83  0.83  0.73  0.02  0.45  0.76  0.02
12 0.79  0.62  0.56  0.59  0.57  0.65  0.29  0.40  0.42  0.16  0.67
13 0.48  0.21  0.16  0.20  0.15  0.41  0.52  0.49  0.20  0.27  0.30  0.61
14 0.59  0.49  0.49  0.48  0.46  0.70  0.23  0.47  0.58 -0.07  0.62  0.50  0.56
15 0.74  0.68  0.58  0.59  0.57  0.79  0.22  0.52  0.63  0.00  0.66  0.77  0.58  0.74
*****
      2 and 3 : 0.9396
      2 and 4 : 0.8958
      3 and 4 : 0.9554
      2 and 5 : 0.9456
      3 and 5 : 0.9958
      4 and 5 : 0.9491
*****

```

Devisen

- (1) = Kanadischer Dollar
- (2) = US Dollar
- (3) = Australischer Dollar
- (4) = Japanische Yen
- (5) = Singapur Dollar
- (6) = Hongkong Dollar
- (7) = Koreanische Won
- (8) = Britische Pfund
- (9) = Schwedische Kronen
- (10) = Italienische Lire
- (11) = Franzoesische Franc
- (12) = Schweizer Franken
- (13) = Spanische Peseten

```

*****
*   CORRELATION ANALYSIS OF DATA TIME SERIES --- CORRELATION COEFFICIENTS   *
*****
      DATA POINTS of TIME SERIES :           1702
      DATA POINTS of CORRELATION ANALYSIS :      10
*****
      1      2      3      4      5      6      7      8      9      10     11     12
2  0.88
3  0.38  0.39
4  0.26  0.30  0.30
5  0.40  0.44  0.69  0.46
6  0.44  0.49  0.74  0.41  0.89
7  0.42  0.46  0.71  0.43  0.84  0.92
8  0.25  0.25  0.41  0.22  0.37  0.41  0.37
9  0.25  0.24  0.33  0.25  0.36  0.40  0.37  0.40
10 0.26  0.25  0.35  0.25  0.40  0.43  0.38  0.42  0.53
11 0.13  0.15  0.27  0.20  0.28  0.30  0.27  0.34  0.44  0.53
12 0.03  0.04  0.00  0.05 -0.02 -0.02 -0.01  0.09  0.05  0.08  0.10
13 0.18  0.14  0.26  0.19  0.29  0.30  0.26  0.38  0.45  0.53  0.45  0.03
*****
      1 and 2 : 0.8795
      5 and 6 : 0.8882
      6 and 7 : 0.9235
*****
*   CORRELATION ANALYSIS OF DATA TIME SERIES --- CORRELATION COEFFICIENTS   *
*****
      DATA POINTS of TIME SERIES :           1702
      DATA POINTS of CORRELATION ANALYSIS :      38
*****
      1      2      3      4      5      6      7      8      9      10     11     12
2  0.90
3  0.66  0.70
4  0.34  0.39  0.36
5  0.68  0.75  0.77  0.53
6  0.76  0.85  0.82  0.44  0.91
7  0.73  0.80  0.78  0.48  0.88  0.94
8  0.42  0.43  0.46  0.24  0.46  0.46  0.42
9  0.38  0.37  0.36  0.19  0.43  0.44  0.41  0.42
10 0.41  0.40  0.36  0.19  0.46  0.46  0.41  0.48  0.61
11 0.26  0.26  0.28  0.12  0.30  0.31  0.26  0.33  0.42  0.47
12 -0.07 -0.07 -0.05  0.02 -0.07 -0.09 -0.08  0.06  0.03 -0.01  0.07
13 0.27  0.22  0.25  0.14  0.29  0.28  0.25  0.41  0.46  0.57  0.40 -0.02
*****
      1 and 2 : 0.9003
      5 and 6 : 0.9128
      5 and 7 : 0.8779
      6 and 7 : 0.9403
*****

```

```

*****
*   CORRELATION ANALYSIS OF DATA TIME SERIES --- CORRELATION COEFFICIENTS   *
*****
      DATA POINTS of TIME SERIES :           1702
      DATA POINTS of CORRELATION ANALYSIS :      70
*****
      1      2      3      4      5      6      7      8      9      10     11     12
2  0.91
3  0.73  0.77
4  0.33  0.40  0.36
5  0.74  0.82  0.79  0.54
6  0.84  0.92  0.83  0.43  0.91
7  0.80  0.86  0.79  0.48  0.87  0.94
8  0.48  0.49  0.50  0.23  0.50  0.50  0.44
9  0.39  0.38  0.36  0.13  0.41  0.43  0.39  0.47
10 0.46  0.44  0.37  0.17  0.46  0.47  0.41  0.54  0.62
11 0.34  0.33  0.34  0.12  0.35  0.36  0.28  0.43  0.46  0.46
12 -0.13 -0.15 -0.13  0.01 -0.16 -0.16 -0.16  0.06  0.04 -0.03  0.06
13 0.30  0.25  0.26  0.09  0.29  0.29  0.26  0.42  0.47  0.60  0.41 -0.06
*****
      1 and 2 : 0.9109
      2 and 6 : 0.9184
      5 and 6 : 0.9102
      2 and 7 : 0.8590
      5 and 7 : 0.8720
      6 and 7 : 0.9363
*****
*   CORRELATION ANALYSIS OF DATA TIME SERIES --- CORRELATION COEFFICIENTS   *
*****
      DATA POINTS of TIME SERIES :           1702
      DATA POINTS of CORRELATION ANALYSIS :      400
*****
      1      2      3      4      5      6      7      8      9      10     11     12
2  0.89
3  0.79  0.81
4  0.14  0.34  0.16
5  0.69  0.87  0.79  0.46
6  0.87  0.98  0.81  0.35  0.89
7  0.83  0.87  0.75  0.32  0.73  0.89
8  0.62  0.52  0.61 -0.03  0.41  0.52  0.46
9  0.50  0.34  0.52 -0.32  0.30  0.35  0.31  0.49
10 0.60  0.43  0.53 -0.34  0.24  0.42  0.42  0.71  0.75
11 0.29  0.24  0.39 -0.31  0.20  0.25  0.21  0.38  0.50  0.54
12 -0.26 -0.21 -0.05 -0.05 -0.16 -0.22 -0.07 -0.13 -0.08 -0.18 -0.11
13 0.51  0.28  0.40 -0.36  0.12  0.28  0.35  0.55  0.67  0.80  0.53 -0.14
*****
      1 and 2 : 0.8897           5 and 6 : 0.8884
      2 and 5 : 0.8672           2 and 7 : 0.8721
      1 and 6 : 0.8734           6 and 7 : 0.8871
      2 and 6 : 0.9834
*****

```


Renten, Rohstoffe, monetäre Größen und Handelsvolumen

- (1) = Wochentag
- (2) = Diskontsatz USA
- (3) = Diskontsatz BRD
- (4) = Geldmenge M3 USA
- (5) = Geldmenge M3 BRD
- (6) = Volumen NYSE
- (7) = Volumen KSE (Seoul)
- (8) = 30 Years US Treasury Bond Rate
- (9) = 10 Years US Treasury Bill Rate
- (10) = 3 Month US Treasury Note Rate
- (11) = Umlaufrendite BRD
- (12) = AMEX Oil and Gas Index
- (13) = AMEX Gold and Silver Index

```

*****
*   CORRELATION ANALYSIS OF DATA TIME SERIES --- CORRELATION COEFFICIENTS   *
*****
      DATA POINTS of TIME SERIES :           1702
      DATA POINTS of CORRELATION ANALYSIS :       10
*****
      1      2      3      4      5      6      7      8      9      10     11     12
2  0.00
3 -0.01 -0.01
4 -0.01  0.01  0.00
5 -0.02 -0.03 -0.01  0.09
6  0.21 -0.01  0.00 -0.06  0.01
7  0.14  0.01  0.02 -0.02  0.00  0.09
8 -0.01  0.01 -0.02  0.02 -0.07 -0.09  0.06
9 -0.03  0.01 -0.02  0.03 -0.07 -0.10  0.06  0.95
10 -0.02  0.05 -0.01  0.05 -0.05 -0.08  0.04  0.38  0.45
11 -0.01  0.01  0.01 -0.04 -0.06 -0.04 -0.02  0.38  0.39  0.15
12  0.01 -0.01  0.00  0.09  0.08  0.08  0.04 -0.19 -0.19 -0.03 -0.14
13  0.02 -0.01 -0.02 -0.01  0.04 -0.02  0.05  0.08  0.05 -0.04  0.04  0.08
*****
      8 and 9 :  0.9499
*****

```

```

*****
*   CORRELATION ANALYSIS OF DATA TIME SERIES --- CORRELATION COEFFICIENTS   *
*****
      DATA POINTS of TIME SERIES :           1702
      DATA POINTS of CORRELATION ANALYSIS :      38
*****
      1      2      3      4      5      6      7      8      9      10     11     12
2  0.00
3 -0.01 -0.07
4 -0.01  0.00 -0.06
5 -0.01 -0.13 -0.12  0.38
6  0.18 -0.02  0.01 -0.04 -0.01
7  0.10  0.00  0.02 -0.01  0.01  0.13
8 -0.02  0.02 -0.05  0.05 -0.06 -0.12  0.13
9 -0.02  0.04 -0.04  0.04 -0.09 -0.14  0.13  0.96
10 -0.01  0.17 -0.03  0.02 -0.18 -0.13 -0.02  0.38  0.47
11  0.00  0.03  0.04 -0.12 -0.13 -0.09 -0.01  0.42  0.42  0.22
12  0.01  0.00 -0.02  0.16  0.16  0.10  0.08 -0.19 -0.20 -0.08 -0.15
13  0.02 -0.03 -0.09 -0.01  0.06 -0.01  0.06  0.02 -0.03 -0.12 -0.01  0.12
*****
      8 and 9 :  0.9587
*****
*   CORRELATION ANALYSIS OF DATA TIME SERIES --- CORRELATION COEFFICIENTS   *
*****
      DATA POINTS of TIME SERIES :           1702
      DATA POINTS of CORRELATION ANALYSIS :      70
*****
      1      2      3      4      5      6      7      8      9      10     11     12
2  0.00
3 -0.01 -0.14
4 -0.01  0.00 -0.12
5  0.00 -0.21 -0.18  0.47
6  0.17 -0.03  0.01 -0.01  0.02
7  0.07 -0.01  0.01  0.02  0.06  0.12
8 -0.01  0.06 -0.08  0.06 -0.06 -0.16  0.17
9 -0.02  0.08 -0.07  0.06 -0.09 -0.17  0.17  0.97
10 -0.01  0.28 -0.09 -0.01 -0.29 -0.14 -0.04  0.42  0.50
11  0.00  0.07  0.06 -0.17 -0.18 -0.16  0.00  0.47  0.46  0.29
12  0.01  0.02 -0.07  0.19  0.19  0.11  0.05 -0.21 -0.23 -0.12 -0.15
13  0.02 -0.04 -0.12 -0.03  0.02 -0.01  0.07 -0.02 -0.06 -0.13 -0.02  0.11
*****
      8 and 9 :  0.9711
*****

```

```

*****
*   CORRELATION ANALYSIS OF DATA TIME SERIES --- CORRELATION COEFFICIENTS   *
*****
      DATA POINTS of TIME SERIES :           1702
      DATA POINTS of CORRELATION ANALYSIS :    400
*****
      1      2      3      4      5      6      7      8      9      10     11     12
2  0.01
3  0.00 -0.33
4  0.00 -0.10 -0.41
5 -0.01 -0.36 -0.40  0.76
6  0.14 -0.10 -0.21  0.34  0.39
7  0.04 -0.23  0.06  0.08  0.18  0.17
8  0.00  0.26 -0.14 -0.20 -0.26 -0.31  0.01
9  0.00  0.30 -0.23 -0.15 -0.22 -0.28  0.00  0.98
10 0.01  0.81 -0.31 -0.11 -0.34 -0.17 -0.19  0.55  0.61
11 0.01  0.44  0.11 -0.44 -0.53 -0.42 -0.12  0.74  0.71  0.59
12 0.00  0.16 -0.66  0.41  0.41  0.27 -0.04 -0.12 -0.08  0.08 -0.28
13 0.01  0.05 -0.45  0.06  0.08  0.09  0.03 -0.19 -0.16 -0.06 -0.18  0.35
*****
      8 and 9 :  0.9805
*****

```

D Vorverarbeitung und Datenkonvertierung

Das Fortran 77-Programm PREPRO liest die Zeitreihen⁴² ein, verarbeitet diese (Logarithmierung, Differenzierung und Skalierung) und schreibt die Ausgaben in eine Datei, die vom SNNS gelesen werden kann. Das Programm ist sehr flexibel gehalten, d.h. die Anzahl der Tage, die dem KNN als Ausgabe dient (der Vorhersagezeitraum einer Simulation, hier der nächste Tag) und der Differentiationsgrad der Eingabedaten (die Anzahl der verhangenen Tage, die zur Vorhersage benutzt werden) können als Parameter frei eingestellt werden.

Das Programm benutzt eine parameter.f-Datei, die die maximalen Größen der zu dimensionierenden Variablen enthält, weil Fortran 77 keine dynamische Speicher-verwaltung zuläßt. Vorteilhaft ist, daß das Programm nicht immer neu kompiliert werden muß, wenn sich die Dimensionen der Eingabedaten ändern, allerdings wird der Nachteil des größeren Speicherplatzbedarfs in Kauf genommen. Sollten die Dimensionen der Eingabedaten die maximalen übersteigen, dann muß jedoch neu kompiliert werden. Änderungen sind aber nur in der parameter.f-Datei notwendig.

Zeile	Parameter	Bedeutung
1		Kommentarzeile
2	<i>M</i>	Zahl der Spalten (Eingaben)
3	<i>NI</i>	Differentiationsgrad der Eingaben (siehe Gl. 5.6)
4	<i>NO</i>	Zahl der Vorhersage-Tage der Ausgabe
5	<i>NN</i>	Zahl der Eingaben ohne Differentiation
6	<i>NN1</i>	Zahl der Eingaben ohne (Dif. & Log.)
7	<i>NN2</i>	Zahl der Eingaben ohne Logarithmierung
8	<i>NV</i>	Zahl des Validierungsschritts
9	<i>NG</i>	Zahl des Generalisierungsstarts
10	<i>XLO</i>	Untere Intervallgrenze (Skalierung)
11	<i>XHI</i>	Obere Intervallgrenze (Skalierung)
ab 12		eigentliche Daten

Abbildung D.1: Aufbau der Eingabedatei

Der Aufbau der Eingabedatei ist in Abb. D.1 dargestellt. Mit dem Validierungsschritt *NV* ist die Schrittweite gemeint, bei der das Muster der Validierungsmenge zugeordnet wird, ansonsten gehört es zur Trainingsmenge. Ab dem *NG*-ten Muster beginnt die Generalisierungsmenge. Den Wert kann man ablesen, wenn man die ersten drei Spalten der eigentlichen Daten betrachtet. In diesen stehen Informationen, die nicht direkt zur Simulation verwendet werden, aber bei der

⁴²Die unterschiedlichen Eingabegrößen sind in Spalten angeordnet und die Zeitpunkte in chronologischer Reihenfolge in Reihen.

Auswertung der Simulationsergebnisse herangezogen werden. Es sind die fortlaufende Zeilennummer (hier kann man NG ablesen), die Nummer des Tages ($1 = (1.1.1990), \dots, 2980 = (27.2.1998)$) und das Datum. Die Nummer des Tages ist für die richtige Auftragung der Daten bei der graphischen Darstellung wichtig, denn zwischen zwei aufeinanderfolgenden Börsentagen muß nicht notwendigerweise ein realer Tag liegen. Auf die Spaltenreihenfolge der nachfolgenden Daten muß geachtet werden. Ab der vierten Spalte kommen die Daten, die weder logarithmiert noch differenziert werden (z.B. Diskontsätze), die Daten, die logarithmiert aber nicht differenziert werden (z.B. Geldmengen $M3$), anschließend die Daten, die zwar differenziert aber nicht logarithmiert werden (z.B. Renten) und schließlich die Daten die sowohl logarithmiert als auch differenziert werden (z.B. Devisen). In der letzte Spalte müssen die Daten der Ausgabe stehen (der DAX).

Es werden drei Ausgabedateien erzeugt, die die Trainings-, Validierungs- und Generalisierungsmuster enthalten und vom SNNS als Eingabe benutzt werden können. Desweiteren wird eine Datei erzeugt, in der die Skalierungsfaktoren der Ausgaben stehen. Diese Informationen benötigt man bei der Rücktransformation der Daten nach der Simulation. Außerdem werden die zu den drei disjunkten Mengen gehörigen Daten (laufende Nummer, Nummer des Tages, Datum) in drei Dateien geschrieben.

Die Auswahl der Muster der Generalisierungsmenge ist nur bei $NO = 1$ eindeutig. Wenn aber beispielsweise die nächsten fünf Tage ($NO = 5$) vorhergesagt werden sollen, dann überlappen sich die Ausgaben der einzelnen Muster im Bereich des Vorhersagezeitraumes. Das Programm verfährt wie folgt: Das erste Muster sagt die nächsten NO Tage voraus, dann werden die nächsten $NO - 1$ Muster übersprungen und das darauffolgende Muster enthält dann die Ausgaben zu den Tagen $NO + 1$ bis $2 \cdot NO$. Problematisch ist nur die Wahl des letzten Muster. Wenn der Modulo zwischen der Anzahl der Tage des Prognosezeitraumes NPG und der Ausgabeneuronen NO ungleich Null ist, dann wird als letztes Muster, dasjenige genommen, das als letzte der NO Ausgaben den letzten Tag des Vorhersagezeitraumes enthält. Bei der Auswertung der Ergebnisse der Simulation muß dieser Umstand dann berücksichtigt werden. In diesem Fall werden die NO Ergebnisse des letzten Musters erst ab dem $\text{MOD}(NPG, NO) + 1$ -ten verwendet, wobei mit MOD der Modulo gemeint ist.

D.1 Quellprogramm in FORTRAN 77

```

PROGRAM PREPRO
C*****
C
C   preprocessing and conversion of data time series
C
C   author: S. Selle, 21.04.1998
C
C   CHECK      = .true. for normal end, .false. otherwise
C   NINP       = number of input file (ASCII table)
C   NSTD       = number of output file (standard output)
C   NSCF       = number of output file (scale factors)
C   NTRAIN     = number of output file (training set)
C   NVALID     = number of output file (validation set)
C   NGENER     = number of output file (generalization set)
C   NDATT      = number of output file (dates of training set)
C   NDATV      = number of output file (dates of validation set)
C   NDATG      = number of output file (dates of generaliz. set)
C   NDUMMY     = number of output file (dummy file)
C   NOUT       = number of output file (NTRAIN, NVALID, NGENER)
C   NOUTD      = number of output file (date)
C   NMAX       = max. number of data points
C   MMAX       = max. number of input/output variables
C   IMAX       = max. number of input steps
C   OMAX       = max. number of output steps
C   N          = number of data points
C   M          = number of input/output variables
C   NI         = number of input steps
C   NO         = number of output steps
C   NN         = number of input steps (without differentiation)
C   NN1        = number of input steps (without differentiation
C                and without taking the logarithm)
C   NN2        = number of input steps (with differentiation
C                and without taking the logarithm)
C   NV         = number of validation output step
C   NG         = number of generalization output start
C   NPA        = number of all pattern
C   NPT        = number of training pattern
C   NPV        = number of validation pattern
C   NPG        = number of generalization pattern
C   NPG1       = number of used generalization pattern
C   NTI        = number of transformed inputs
C   NDAT(3,NMAX) = input data (date and number of line)
C   DAT(MMAX,NMAX) = input data (ASCII table)
C   OUT(MMAX,IMAX+OMAX,NMAX) = output data (input for snns)
C   XMIN,XMAX  = min. (max.) value of data series
C   XLO,XHI    = lower (upper) value of intervall
C   IT,IV,IG   = counter of pattern sets
C   I,J,K,L,LL,II= auxiliary variables
C   FDATE      = date and time (sgi specific function)
C
C*****

```

```

C*****
C      type declarations                                     *
C*****
      IMPLICIT NONE
      INCLUDE 'parameter.f'
      INTEGER N,M,NI,NO,NN,NN1,NN2,NV,NG,NPA,NPT,NPV,NPG,NPG1,NTI
      INTEGER NOUT,NOUTD,I,J,K,L,LL,II,IT,IV,IG,NDAT(3,NMAX)
      DOUBLE PRECISION DAT(MMAX,NMAX),OUT(MMAX,IMAX+OMAX,NMAX)
      DOUBLE PRECISION XMIN,XMAX,XLO,XHI
      LOGICAL CHECK
      CHARACTER*24 FDATE
      EXTERNAL FDATE
C*****
C      initialization                                       *
C*****
      CHECK=.TRUE.
C*****
C      start of program -PREPRO-                             *
C*****
      WRITE(NSTD,900)
      WRITE(NSTD,901)
      WRITE(NSTD,900)
C*****
C      read input data from file (ASCII table)              *
C*****
      WRITE(NSTD,902)
      CALL INPDAT(NINP,NSTD,NMAX,MMAX,IMAX,OMAX,N,M,NI,NO,NN,NN1,NN2,NV,
F          NG,XLO,XHI,NDAT,DAT,CHECK)
      IF(.NOT.CHECK) THEN
          WRITE(NSTD,998)
          WRITE(NSTD,999)
          GOTO 1000
      ENDIF
C*****
C      write to standard output: parameter                  *
C*****
      WRITE(NSTD,903) N
      WRITE(NSTD,904) M
      WRITE(NSTD,905) NI
      WRITE(NSTD,906) NO
      WRITE(NSTD,907) NN
      WRITE(NSTD,908) NN1
      WRITE(NSTD,909) NN2
      WRITE(NSTD,910) NV
      WRITE(NSTD,911) NG
      WRITE(NSTD,900)

```

```

C*****
C    calculate number of pattern, inputs, etc.                *
C*****
      NPA=N-(NI+NO)+1
      NPG=N-NG+1
      NPV=(NPA-NPG)/NV
      NPT=NPA-NPG-NPV
      NTI=NN+NI*(M-NN)
      NPG1=IDNINT(DBLE(NPG)/DBLE(NO))
C*****
C    write to standard output: calculated parameter          *
C*****
      WRITE(NSTD,912) NPA
      WRITE(NSTD,913) NPT
      WRITE(NSTD,914) NPV
      WRITE(NSTD,915) NPG1
      WRITE(NSTD,916) NTI
      WRITE(NSTD,900)
C*****
C    taking the logarithm of input data (given in absolute prices) *
C*****
      WRITE(NSTD,917)
      DO I=1,N
        DO J=NN1+1,NN
          DAT(J,I)=DLOG(DAT(J,I))
        ENDDO
        DO J=NN+NN2+1,M
          DAT(J,I)=DLOG(DAT(J,I))
        ENDDO
      ENDDO
C*****
C    last input = output (e.g. DAX values)                  *
C*****
      DO I=1,N
        DAT(M+1,I)=DAT(M,I)
      ENDDO
C*****
C    time differentiation of data                          *
C*****
      WRITE(NSTD,918)
      DO J=1,M
        DO I=1,NPA
          DO K=1,NI
            IF ((K.GT.1).AND.(J.GT.NN)) THEN
              OUT(J,K,I)=(DAT(J,I+NI-1)-DAT(J,I+NI-K))/DBLE(K-1)
            ELSE
              OUT(J,K,I)=DAT(J,I+NI-1)
            ENDIF
          ENDDO
        ENDDO
      ENDDO

```



```

C---- output data for snns (no time differentiation)
      DO I=1,NPA
        DO K=1,NO
          OUT(M+1,K,I)=DAT(M+1,I+NI+K-1)
        ENDDO
      ENDDO

C*****
C      scale data to intervall [XLO, XHI] *
C*****
      WRITE(NSTD,919) XLO,XHI
C---- input data for snns
      DO J=1,M
        DO K=1,NI
          XMIN= 1.0D100
          XMAX=-1.0D100
          DO I=1,NPA
            IF (OUT(J,K,I).LT.XMIN) THEN
              XMIN=OUT(J,K,I)
            ENDIF
            IF (OUT(J,K,I).GT.XMAX) THEN
              XMAX=OUT(J,K,I)
            ENDIF
          ENDDO
          DO I=1,NPA
            OUT(J,K,I)=(XHI-XLO)*((OUT(J,K,I)-XMIN)/(XMAX-XMIN))+XLO
          ENDDO
        ENDDO
      ENDDO

C---- output data for snns
      WRITE(NSCF,925) NO
      DO K=1,NO
        XMIN= 1.0D100
        XMAX=-1.0D100
        DO I=1,NPA
          IF (OUT(M+1,K,I).LT.XMIN) THEN
            XMIN=OUT(M+1,K,I)
          ENDIF
          IF (OUT(M+1,K,I).GT.XMAX) THEN
            XMAX=OUT(M+1,K,I)
          ENDIF
        ENDDO
        DO I=1,NPA
          OUT(M+1,K,I)=(XHI-XLO)*((OUT(M+1,K,I)-XMIN)/(XMAX-XMIN))+XLO
        ENDDO
        WRITE(NSTD,920) K,(XMAX-XMIN)/(XHI-XLO),XLO,XMIN
        WRITE(NSCF,926) (XMAX-XMIN)/(XHI-XLO),XLO,XMIN
      ENDDO

```

```

C*****
C    write output : snns readable format
C*****
    WRITE(NSTD,921)
C---- write header
    WRITE(NTRAIN,930) FDATE()
    WRITE(NVALID,930) FDATE()
    WRITE(NGENER,930) FDATE()
    WRITE(NTRAIN,931) NPT,NTI,NO
    WRITE(NVALID,931) NPV,NTI,NO
    WRITE(NGENER,931) NPG1,NTI,NO
    L=0
    LL=0
    IT=0
    IV=0
    IG=0
    DO I=1,NPA
        IF (I.LT.(NPT+NPV+1)) THEN
            L=L+1
            IF (L.NE.NV) THEN
C---- training pattern
                IT=IT+1
                II=IT
                NOUT=NTRAIN
                NOUTD=NDATT
            ELSE
C---- validation pattern
                IV=IV+1
                II=IV
                NOUT=NVALID
                NOUTD=NDATV
                L=0
            ENDIF
        ELSE
C---- generalization pattern
            LL=LL+1
            IF((LL.EQ.NO).OR.((I.EQ.NPA).AND.(MOD(NPG,NPG1).NE.0))) THEN
                IG=IG+1
                II=IG
                NOUT=NGENER
                NOUTD=NDATG
                LL=0
            ELSE
                II=IG
                NOUT=NDUMMY
                NOUTD=NDATG
            ENDIF
        ENDIF
    ENDIF

```

```

C---- write output data (snns)
      WRITE(NOUT,932) II
      WRITE(NOUT,934) (OUT(J,1,I),J=1,NN),
F      ((OUT(J,K,I),J=NN+1,M),K=1,NI)
      WRITE(NOUT,933) II
      WRITE(NOUT,934) (OUT(M+1,K,I),K=1,NO)
C---- write output data (date & number)
      WRITE(NOUTD,935) II,(NDAT(J,I+NO+NI-1),J=1,3)
      ENDDO
C*****
C      check counter *
C*****
      IF (IT.NE.NPT) THEN
        WRITE(NSTD,995) IT
        WRITE(NSTD,999)
        GOTO 1000
      ENDIF
      IF (IV.NE.NPV) THEN
        WRITE(NSTD,996) IV
        WRITE(NSTD,999)
        GOTO 1000
      ENDIF
      IF (IG.NE.NPG1) THEN
        WRITE(NSTD,997) IG
        WRITE(NSTD,999)
        GOTO 1000
      ENDIF
      WRITE(NSTD,922)
      WRITE(NSTD,900)
C*****
C      format statements *
C*****
900  FORMAT(80('*'))
901  FORMAT(4X,'PREPROCESSING OF DATA TIME SERIES')
902  FORMAT(' (1) READ INPUT DATA')
903  FORMAT(5X,'Number of Data Points           : ',I5)
904  FORMAT(5X,'Number of Input Series          : ',I5)
905  FORMAT(5X,'Number of Input Steps           : ',I5)
906  FORMAT(5X,'Number of Output Steps          : ',I5)
907  FORMAT(5X,'Number of Input Series (no Dif)   : ',I5)
908  FORMAT(5X,'Number of Input Series (no Dif & Log) : ',I5)
909  FORMAT(5X,'Number of Input Series (no Log)    : ',I5)
910  FORMAT(5X,'Number of Validation Output Step  : ',I5)
911  FORMAT(5X,'Number of Generalization Start    : ',I5)
912  FORMAT(5X,'Number of All Pattern             : ',I5)
913  FORMAT(5X,'Number of Training Pattern        : ',I5)
914  FORMAT(5X,'Number of Validation Pattern      : ',I5)
915  FORMAT(5X,'Number of Generalization Pattern  : ',I5)
916  FORMAT(5X,'Number of Transformed Inputs     : ',I5)
917  FORMAT(' (2) LOGARITHM DATA')
918  FORMAT(' (3) TIME DIFFERENTIATION OF DATA')

```

```

919 FORMAT(' (4) SCALING DATA to INTERVALL [',F6.3,' , ',F6.3,' ]')
920 FORMAT(5X,'Output Step Nr.',I3,' scale factors:',3(1X,D12.5))
921 FORMAT(' (5) WRITE OUTPUT DATA')
922 FORMAT(' (6) PROGRAM END')
925 FORMAT(I5)
926 FORMAT(D15.8)
930 FORMAT('SNNS pattern definition file V3.2',/, 'generated at ',A24,/
F,/)
931 FORMAT('No. of patterns : ',I5,/, 'No. of input units : ',I5,/, 'No.
F of output units : ',I5,/)
932 FORMAT('# Input pattern ',I5,':')
933 FORMAT('# Output pattern ',I5,':')
934 FORMAT(8(F9.6,1X))
935 FORMAT(4(1X,I6))
995 FORMAT(5X,'ERROR in -PREPRO- : wrong num. of train. pattern: ',I5)
996 FORMAT(5X,'ERROR in -PREPRO- : wrong num. of valid. pattern: ',I5)
997 FORMAT(5X,'ERROR in -PREPRO- : wrong num. of gener. pattern: ',I5)
998 FORMAT(5X,'ERROR in -PREPRO- : wrong output from -INPDAT-')
999 FORMAT(5X,'PROGRAM TERMINATED')
C*****
C      end of program -PREPRO-                                     *
C*****
1000 END

```

```

      SUBROUTINE INPDAT(NINP,NSTD,NMAX,MMAX,IMAX,OMAX,N,M,NI,NO,NN,NN1,
F      NN2,NV,NG,XLO,XHI,NDAT,DAT,CHECK)
C*****
C
C      subroutine for reading the input data                       *
C
C
C*****
C      type declarations                                           *
C*****
      IMPLICIT NONE
      INTEGER NINP,NSTD,NMAX,MMAX,IMAX,OMAX,N,M,NI,NO,NN,NN1,NN2,NV,NG
      INTEGER NDAT(3,NMAX),I,J
      DOUBLE PRECISION XLO,XHI,DAT(MMAX,NMAX)
      LOGICAL CHECK
C*****
C      initialization                                               *
C*****
      CHECK=.TRUE.
C*****
C      start of subroutine -INPDAT-                                 *
C*****

```

```

C*****
C      read header from file and check parameter values      *
C*****
      OPEN(NINP,ERR=100)
      REWIND(NINP)
      READ(NINP,*)
      READ(NINP,*) N
      IF ((N.LT.1).OR.(N.GT.NMAX)) THEN
        WRITE(NSTD,986) N
        CHECK=.FALSE.
        RETURN
      ENDIF
      READ(NINP,*) M
      IF ((M.LT.1).OR.((M+1).GT.MMAX)) THEN
        WRITE(NSTD,987) M
        CHECK=.FALSE.
        RETURN
      ENDIF
      READ(NINP,*) NI
      IF ((NI.LT.1).OR.(NI.GT.IMAX)) THEN
        WRITE(NSTD,988) NI
        CHECK=.FALSE.
        RETURN
      ENDIF
      READ(NINP,*) NO
      IF ((NO.LT.1).OR.(NO.GT.OMAX)) THEN
        WRITE(NSTD,989) NO
        CHECK=.FALSE.
        RETURN
      ENDIF
      READ(NINP,*) NN
      IF ((NN.LT.0).OR.(NN.GT.M)) THEN
        WRITE(NSTD,990) NN
        CHECK=.FALSE.
        RETURN
      ENDIF
      READ(NINP,*) NN1
      IF ((NN1.LT.0).OR.(NN1.GT.NN)) THEN
        WRITE(NSTD,991) NN1
        CHECK=.FALSE.
        RETURN
      ENDIF
      READ(NINP,*) NN2
      IF ((NN2.LT.0).OR.(NN2.GT.(M-NN))) THEN
        WRITE(NSTD,992) NN2
        CHECK=.FALSE.
        RETURN
      ENDIF

```

```

      READ(NINP,*) NV
      IF ((NV.LT.1).AND.(NV.GT.N)) THEN
        WRITE(NSTD,993) NV
        CHECK=.FALSE.
        RETURN
      ENDIF
      READ(NINP,*) NG
      IF ((NG.LT.1).OR.(NG.GT.N)) THEN
        WRITE(NSTD,994) NG
        CHECK=.FALSE.
        RETURN
      ENDIF
      READ(NINP,*) XLO
      IF (XLO.LE.0.0D0) THEN
        WRITE(NSTD,995) XLO
        CHECK=.FALSE.
        RETURN
      ENDIF
      READ(NINP,*) XHI
      IF ((XHI.LE.XLO).OR.(XHI.GT.1.0D0)) THEN
        WRITE(NSTD,996) XHI
        CHECK=.FALSE.
        RETURN
      ENDIF
C*****
C   read input values from file and check values
C*****
      DO I=1,N
        READ(NINP,*,END=200) (NDAT(J,I),J=1,3),(DAT(J,I),J=1,M)
        DO J=1,M
          IF (DAT(J,I).EQ.0.0D0) THEN
            WRITE(NSTD,997) I,J
            CHECK=.FALSE.
            RETURN
          ENDIF
        ENDDO
      ENDDO
      CLOSE(NINP)
      RETURN
C*****
C   error exit
C*****
100  CONTINUE
      WRITE(NSTD,998)
      CHECK=.FALSE.
      RETURN
200  CONTINUE
      WRITE(NSTD,999)
      CHECK=.FALSE.
      RETURN

```

```

C*****
C      format statements                                          *
C*****
986  FORMAT(5X,'ERROR in -INPDAT- : wrong number of N:  ',I5)
987  FORMAT(5X,'ERROR in -INPDAT- : wrong number of M:  ',I5)
988  FORMAT(5X,'ERROR in -INPDAT- : wrong number of NI:  ',I5)
989  FORMAT(5X,'ERROR in -INPDAT- : wrong number of N0:  ',I5)
990  FORMAT(5X,'ERROR in -INPDAT- : wrong number of NN:  ',I5)
991  FORMAT(5X,'ERROR in -INPDAT- : wrong number of NN1: ',I5)
992  FORMAT(5X,'ERROR in -INPDAT- : wrong number of NN2: ',I5)
993  FORMAT(5X,'ERROR in -INPDAT- : wrong number of NV:  ',I5)
994  FORMAT(5X,'ERROR in -INPDAT- : wrong number of NG:  ',I5)
995  FORMAT(5X,'ERROR in -INPDAT- : wrong value of XL0:  ',D12.5)
996  FORMAT(5X,'ERROR in -INPDAT- : wrong value of XHI:  ',D12.5)
997  FORMAT(5X,'ERROR in -INPDAT- : zero value of DAT(' ,I5,1X,I5,')')
998  FORMAT(5X,'ERROR in -INPDAT- : cannot open input file')
999  FORMAT(5X,'ERROR in -INPDAT- : end of file reached')
C*****
C      end of subroutine -INPDAT-                                *
C*****
      END

C*****
C
C      parameter.f file                                          *
C
C
C*****
C      type declarations                                          *
C*****
      INTEGER NMAX,MMAX,IMAX,OMAX,NINP,NSTD,NSCF,NTRAIN,NVALID,NGENER
      INTEGER NDATT,NDATV,NDATG,NDUMMY
      PARAMETER (NMAX=2000)
      PARAMETER (MMAX=50)
      PARAMETER (IMAX=20)
      PARAMETER (OMAX=5)
      PARAMETER (NINP=3)
      PARAMETER (NSTD=6)
      PARAMETER (NSCF=10)
      PARAMETER (NTRAIN=11)
      PARAMETER (NVALID=12)
      PARAMETER (NGENER=13)
      PARAMETER (NDATT=21)
      PARAMETER (NDATV=22)
      PARAMETER (NDATG=23)
      PARAMETER (NDUMMY=99)
C*****
C      end of PARAMETER.F file                                    *
C*****

```

E Simulationsergebnisse

E.1 Depotentwicklung

In der nachfolgenden Tabelle ist die wertmäßige Depotentwicklung dargestellt. MLP bezeichnet das (150-15-1) MLP, OPT das optimierte (20-4-1) MLP und NAIV die naive Prognose. Die Zahlenwerte sind nur gerundet ausgegeben, intern wird mit doppelter Genauigkeit (32-Bit Gleitpunktarithmetik) gerechnet.

Nr	Datum	DAX-Werte		Prognosen			Depot-Wert [DM]		
		gest.	heute	MLP	OPT	NAIV	MLP	OPT	NAIV
0	30.09.97		4168				100000	100000	100000
1	01.10.97	4168	4197	4098	4131	4168	100000	100000	100696
2	02.10.97	4197	4274	4113	4154	4197	100000	100000	102543
3	06.10.97	4274	4302	4133	4195	4274	100000	100000	103214
4	07.10.97	4302	4307	4148	4214	4302	100000	100000	103335
5	08.10.97	4307	4347	4149	4214	4307	100000	100000	104295
6	09.10.97	4347	4243	4138	4226	4347	100000	100000	101799
7	14.10.97	4243	4202	4111	4173	4243	100000	100000	100816
8	15.10.97	4202	4194	4110	4165	4202	100000	100000	100625
9	16.10.97	4194	4118	4086	4160	4194	100000	100000	98800
10	17.10.97	4118	4062	4052	4120	4118	100000	98641	97457
11	20.10.97	4062	4041	4038	4091	4062	100000	98130	96952
12	21.10.97	4041	4140	4016	4080	4041	100000	100535	99328
13	22.10.97	4140	4171	4077	4129	4140	100000	100535	100072
14	23.10.97	4171	3977	4064	4144	4171	100000	100535	95417
15	24.10.97	3977	4051	3992	4017	3977	101861	102406	97193
16	27.10.97	4051	3879	4029	4087	4051	101861	98059	93067
17	28.10.97	3879	3567	3971	3973	3879	93668	90172	85581
18	29.10.97	3567	3792	3796	3742	3567	99576	95858	90978
19	31.10.97	3792	3727	3950	3899	3792	97869	94216	89419
20	04.11.97	3727	3812	3875	3825	3727	100102	96365	91459
21	05.11.97	3812	3867	3895	3901	3812	101545	97754	92778
22	06.11.97	3867	3824	3951	3928	3867	100417	96668	91747
23	07.11.97	3824	3700	3905	3893	3824	97160	93533	88772
24	10.11.97	3700	3753	3863	3819	3700	98552	94873	90043
25	12.11.97	3753	3659	3874	3867	3753	96084	92497	87788
26	13.11.97	3659	3704	3844	3800	3659	97265	93634	88868
27	14.11.97	3704	3731	3860	3830	3704	97975	94317	89516
28	17.11.97	3731	3795	3877	3843	3731	99655	95934	91051
29	19.11.97	3795	3835	3918	3883	3795	100706	96946	92011
30	20.11.97	3835	3916	3969	3912	3835	102833	98994	93954
31	21.11.97	3916	3960	3989	3975	3916	103988	100106	95010
32	25.11.97	3960	3849	4025	3998	3960	101072	97299	92346
33	26.11.97	3849	3917	3959	3926	3849	102859	99019	93978
34	28.11.97	3917	3949	4002	3982	3917	103698	99827	94745

35	02.12.97	3949	4073	4022	4017	3949	106955	102962	97720
36	03.12.97	4073	4083	4062	4097	4073	106955	103216	97961
37	04.12.97	4083	4140	4069	4088	4083	106955	104656	99328
38	05.12.97	4140	4170	4102	4121	4140	106955	104656	100048
39	08.12.97	4170	4223	4126	4152	4170	106955	104656	101320
40	09.12.97	4223	4185	4129	4177	4223	106955	104656	100407
41	10.12.97	4185	4117	4126	4167	4185	106955	104656	98776
42	11.12.97	4117	4030	4098	4129	4117	106955	102445	96690
43	12.12.97	4030	4083	4040	4072	4030	108361	103792	97961
44	15.12.97	4083	4060	4075	4119	4083	108361	103207	97409
45	16.12.97	4060	4084	4071	4100	4060	109001	103817	97984
46	17.12.97	4084	4159	4097	4108	4084	111004	105724	99785
47	19.12.97	4159	4085	4110	4145	4159	111004	105724	98008
48	22.12.97	4085	4043	4077	4091	4085	111004	104637	97000
49	05.01.98	4043	4385	4073	4081	4043	120394	113489	105206
50	07.01.98	4385	4392	4221	4285	4385	120394	113489	105375
51	08.01.98	4392	4347	4198	4268	4392	120394	113489	104295
52	09.01.98	4347	4237	4188	4254	4347	120394	113489	101655
53	12.01.98	4237	4087	4157	4205	4237	120394	113489	98056
54	13.01.98	4087	4159	4099	4119	4087	122516	115489	99785
55	14.01.98	4159	4190	4149	4155	4159	122516	115489	100529
56	16.01.98	4190	4184	4148	4154	4190	122516	115489	100384
57	20.01.98	4184	4308	4137	4138	4184	122516	115489	103358
58	21.01.98	4308	4283	4178	4209	4308	122516	115489	102760
59	22.01.98	4283	4220	4174	4203	4283	122516	115489	101248
60	23.01.98	4220	4237	4149	4197	4220	122516	115489	101655
61	02.02.98	4237	4523	4158	4201	4237	122516	115489	108518
62	03.02.98	4523	4533	4278	4341	4523	122516	115489	108758
63	04.02.98	4533	4487	4255	4331	4533	122516	115489	107653
64	05.02.98	4487	4548	4257	4296	4487	122516	115489	109117
65	06.02.98	4548	4496	4281	4351	4548	122516	115489	107869
66	09.02.98	4496	4564	4247	4328	4496	122516	115489	109501
67	10.02.98	4564	4524	4259	4350	4564	122516	115489	108542
68	12.02.98	4524	4537	4278	4335	4524	122516	115489	108853
69	13.02.98	4537	4502	4252	4338	4537	122516	115489	108013
70	27.02.98	4502	4710	4245	4316	4502	122516	115489	113004

E.2 Topologie des optimierten MLP

Nachfolgend ist der Ausdruck, der vom SNNS erzeugten Datei des optimierten (20-4-1) MLP angegeben, das die besten Simulationsergebnisse erzielt hat. Nachträglich sind die Bedeutungen der einzelnen Eingabeneuronen eingezeichnet worden (allerdings unter der Spalte *typeName*). In der Spalte *st* steht das Symbol *i* für ein aktives Eingabeneuron, *si* für ein inaktives, *h* für ein verborgenes und *o* für ein Ausgabeneuron. Aus Darstellungsgründen sind bei den Gewichtsangaben nur 80 Zeichen pro Zeile ausgegeben worden.

SNNS network definition file V1.4-3D
generated at Mon May 11 12:41:16 1998

network name : sim17

source files :

no. of units : 155

no. of connections : 24

no. of unit types : 0

no. of site types : 0

learning function : BackpropMomentum

update function : Topological_Order

unit default section :

act	bias	st	subnet	layer	act func	out func
0.00000	0.00000	h	0	1	Act_Logistic	Out_Identity

unit definition section :

no.	typeName	unitName	act	bias	st	position
1	Wochentag		0.90000	0.20707	si	2, 2, 72
2	Disc USA		0.50000	-0.68097	si	2, 3, 72
3	Disk BRD		0.10000	0.29333	si	2, 4, 72
4	M3 USA		0.90000	0.10683	i	2, 5, 72
5	M3 BRD		0.90000	0.36261	si	2, 6, 72
6	Vol. NYSE		0.70235	-0.77286	si	2, 7, 72
7			0.46369	0.96883	si	2, 8, 72
8			0.13908	-0.16818	i	2, 9, 72
9			0.46567	0.68126	si	2, 10, 72
10			0.10000	0.23257	i	2, 11, 72
11	Vol. KSE		0.79282	-0.38201	i	2, 12, 72
12			0.27948	0.88228	si	2, 13, 72
13			0.80383	-0.79971	i	2, 14, 72
14			0.57414	0.67291	i	2, 15, 72
15			0.65047	0.68186	si	2, 16, 72
16	30Y US TB		0.82604	-0.06613	i	2, 17, 72
17			0.47724	-0.20829	si	2, 18, 72
18			0.45333	-0.16540	si	2, 19, 72
19			0.84707	0.46006	si	2, 20, 72
20			0.88261	0.21904	si	2, 21, 72
21	3M US TN		0.27108	0.41320	si	2, 22, 72
22			0.87041	-0.48475	si	2, 23, 72
23			0.90000	0.40295	si	2, 24, 72
24			0.89133	0.84892	si	2, 25, 72
25			0.85558	0.19003	i	2, 26, 72

26	Umlaufrend	0.23042	-0.71476	si	2, 27, 72
27		0.45069	-0.88394	si	2, 28, 72
28		0.68568	-0.32566	si	2, 29, 72
29		0.34931	-0.69295	si	2, 30, 72
30		0.89341	0.99449	i	2, 31, 72
31	Oil Gas	0.87895	0.07524	i	2, 32, 72
32		0.89611	-0.06519	si	2, 33, 72
33		0.88522	-0.57575	i	2, 34, 72
34		0.89116	0.46878	i	2, 35, 72
35		0.43786	-0.96550	si	2, 36, 72
36	Gold Silvr	0.57645	0.38643	si	2, 37, 72
37		0.37143	-0.20171	si	2, 38, 72
38		0.53871	-0.21743	si	2, 39, 72
39		0.36667	0.57512	si	2, 40, 72
40		0.44636	-0.77179	si	2, 41, 72
41	US-Dollar	0.48311	0.39069	si	2, 42, 72
42		0.46612	0.69628	si	2, 43, 72
43		0.36371	0.32468	si	2, 44, 72
44		0.19877	0.22505	si	2, 45, 72
45		0.56010	-0.28722	i	2, 46, 72
46	Austr. Dol	0.68105	-0.50711	si	2, 47, 72
47		0.53855	-0.20723	si	2, 48, 72
48		0.56556	-0.06326	si	2, 49, 72
49		0.36109	0.08128	si	2, 50, 72
50		0.47115	-0.30498	si	2, 51, 72
51	Jap. Yen	0.65721	0.34564	si	2, 52, 72
52		0.60647	-0.89274	si	2, 53, 72
53		0.48259	0.26392	si	2, 54, 72
54		0.49424	-0.88778	si	2, 55, 72
55		0.36234	0.98514	si	2, 56, 72
56	Brit. Pfun	0.32956	-0.03320	si	2, 57, 72
57		0.38595	0.45446	si	2, 58, 72
58		0.46459	0.95140	si	2, 59, 72
59		0.44566	-0.18391	si	2, 60, 72
60		0.41037	0.59007	si	2, 61, 72
61	S. Kronen	0.41338	-0.75539	si	2, 62, 72
62		0.43865	0.88126	si	2, 63, 72
63		0.50394	0.47831	si	2, 64, 72
64		0.57304	0.39434	si	2, 65, 72
65		0.47881	-0.78299	si	2, 66, 72
66	Ital. Lire	0.37042	0.00360	si	2, 67, 72
67		0.57368	-0.42005	si	2, 68, 72
68		0.39538	0.42717	si	2, 69, 72
69		0.45963	-0.72985	si	2, 70, 72
70		0.42457	-0.35108	si	2, 71, 72
71	Fr. Francs	0.48693	0.08302	si	2, 72, 72
72		0.39270	-0.91927	si	2, 73, 72
73		0.37566	-0.54935	si	2, 74, 72
74		0.68098	0.24690	si	2, 75, 72
75		0.70101	0.54272	si	2, 76, 72

76	S. Franken	0.63392	0.29093	si	2, 77, 72
77		0.62253	0.39936	si	2, 78, 72
78		0.49946	-0.33953	si	2, 79, 72
79		0.66421	-0.43793	si	2, 80, 72
80		0.63645	-0.13191	si	2, 81, 72
81	S. Peseten	0.61234	-0.64893	i	2, 82, 72
82		0.52375	0.21878	i	2, 83, 72
83		0.61588	-0.63753	si	2, 84, 72
84		0.42674	-0.68802	si	2, 85, 72
85		0.44804	0.63906	i	2, 86, 72
86	TSE 300	0.41790	-0.88679	si	2, 87, 72
87		0.31147	0.15023	si	2, 88, 72
88		0.40671	-0.04133	si	2, 89, 72
89		0.40159	0.86871	si	2, 90, 72
90		0.41625	-0.82702	si	2, 91, 72
91	Dow Jones	0.44431	-0.76517	si	2, 92, 72
92		0.56103	-0.52273	si	2, 93, 72
93		0.54655	0.20215	si	2, 94, 72
94		0.50898	-0.37651	i	2, 95, 72
95		0.40562	-0.52310	si	2, 96, 72
96	NASDAQ C	0.55454	0.18998	si	2, 97, 72
97		0.35366	0.74950	si	2, 98, 72
98		0.46427	0.03073	si	2, 99, 72
99		0.49132	0.55507	si	2,100, 72
100		0.56621	-0.74624	si	2,101, 72
101	All Ordin.	0.45297	-0.05764	si	2,102, 72
102		0.44368	0.99717	si	2,103, 72
103		0.62177	-0.72103	si	2,104, 72
104		0.69537	0.96261	si	2,105, 72
105		0.62253	-0.62978	si	2,106, 72
106	Nikkei 225	0.59877	0.51423	si	2,107, 72
107		0.48918	0.37086	si	2,108, 72
108		0.65466	-0.93147	si	2,109, 72
109		0.66029	-0.87112	si	2,110, 72
110		0.62424	-0.61886	i	2,111, 72
111	STI	0.52356	0.01578	i	2,112, 72
112		0.50070	0.55973	si	2,113, 72
113		0.39002	-0.15271	si	2,114, 72
114		0.60812	0.42194	i	2,115, 72
115		0.43002	0.35787	si	2,116, 72
116	Hang Seng	0.34086	0.47735	si	2,117, 72
117		0.42945	0.65202	si	2,118, 72
118		0.42736	0.99637	si	2,119, 72
119		0.40441	0.91043	si	2,120, 72
120		0.54490	0.51765	si	2,121, 72
121	KOSPI	0.49254	0.16910	si	2,122, 72
122		0.42786	-0.40325	si	2,123, 72
123		0.45243	0.55631	si	2,124, 72
124		0.40769	-0.00782	si	2,125, 72
125		0.58889	0.86594	si	2,126, 72

126	FTSE 100		0.35000	0.21387	si	2,127, 72
127			0.47522	0.04105	si	2,128, 72
128			0.57485	0.56372	si	2,129, 72
129			0.57305	0.56098	si	2,130, 72
130			0.48434	-0.83241	si	2,131, 72
131	AGFX		0.51789	0.29956	si	2,132, 72
132			0.59333	0.61461	si	2,133, 72
133			0.70264	0.57753	si	2,134, 72
134			0.63287	-0.26736	si	2,135, 72
135			0.57184	0.89287	si	2,136, 72
136	MIB		0.53123	-0.56682	si	2,137, 72
137			0.66746	0.93865	si	2,138, 72
138			0.70901	0.11181	si	2,139, 72
139			0.65617	0.64717	si	2,140, 72
140			0.53820	-0.86436	si	2,141, 72
141	CAC-40		0.57821	-0.81930	si	2,142, 72
142			0.42576	0.05431	si	2,143, 72
143			0.62527	-0.74339	si	2,144, 72
144			0.55620	-0.39113	si	2,145, 72
145			0.48766	-0.86038	si	2,146, 72
146	DAX		0.44770	0.08982	si	2,147, 72
147			0.44763	0.20287	si	2,148, 72
148			0.44039	-0.78661	si	2,149, 72
149			0.48870	-0.38709	si	2,150, 72
150			0.61319	-0.67747	si	2,151, 72
151			0.99410	4.70423	h	5, 3, 72
152			0.99117	4.20445	h	5, 7, 72
153			0.98780	4.46907	h	5, 9, 72
154			0.24508	1.17208	h	5, 13, 72
155			0.84495	33.61520	o	8, 2, 72
----	-----	-----	-----	-----	----	-----

connection definition section :

target	site	source:weight
-----	-----	-----
151		94: 0.82887
152		81: 0.84257
153		45:-0.13335
154		4: 0.09529, 8: 0.09589, 10:-0.24008, 11: 0.08192,
		13:-0.09752, 14: 0.08670, 16: 0.05408, 25:-0.00952,
		30:-0.14687, 31:-0.10861, 33:-0.09388, 34:-2.38862,
		82:-0.00738, 85:-0.01941, 110:-0.04497, 111: 0.04303,
		114: 0.01885
155		151:-9.21856, 152:-11.76663, 153:-9.26991, 154:-7.89957
-----	-----	-----