

Jan Vogt

Implementierung von Systemen zur
Zeitreihenprognose mittels Neuronaler
Netze und Evolutionärer Algorithmen in
JAVA und Anwendung dieser Systeme auf
Kapitalmarktdaten

Diplomarbeit

BEI GRIN MACHT SICH IHR WISSEN BEZAHLT



- Wir veröffentlichen Ihre Hausarbeit, Bachelor- und Masterarbeit
- Ihr eigenes eBook und Buch - weltweit in allen wichtigen Shops
- Verdienen Sie an jedem Verkauf

Jetzt bei www.GRIN.com hochladen
und kostenlos publizieren



Bibliografische Information der Deutschen Nationalbibliothek:

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

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Impressum:

Copyright © 2002 GRIN Verlag
ISBN: 9783668742574

Dieses Buch bei GRIN:

<https://www.grin.com/document/432109>

Jan Vogt

**Implementierung von Systemen zur Zeitreihenprognose
mittels Neuronaler Netze und Evolutionärer Algorithmen
in JAVA und Anwendung dieser Systeme auf
Kapitalmarktdaten**

GRIN - Your knowledge has value

Der GRIN Verlag publiziert seit 1998 wissenschaftliche Arbeiten von Studenten, Hochschullehrern und anderen Akademikern als eBook und gedrucktes Buch. Die Verlagswebsite www.grin.com ist die ideale Plattform zur Veröffentlichung von Hausarbeiten, Abschlussarbeiten, wissenschaftlichen Aufsätzen, Dissertationen und Fachbüchern.

Besuchen Sie uns im Internet:

<http://www.grin.com/>

<http://www.facebook.com/grincom>

http://www.twitter.com/grin_com

DIPLOMARBEIT

Implementierung von Systemen zur Zeitreihenprognose mittels
Neuronaler Netze und Evolutionärer Algorithmen in JAVA und
Anwendung dieser Systeme auf Kapitalmarktdaten

von

Jan Vogt

eingereicht am 21.05.2002 beim
Institut für Angewandte Informatik
und Formale Beschreibungsverfahren
der Universität Karlsruhe

Inhaltsverzeichnis

Abkürzungsverzeichnis.....	5
1 Einleitung	6
1.1 Einführung in die Thematik	6
1.2 Ziel der Arbeit	7
1.3 Aufbau der Arbeit.....	8
2 Neuronale Netze	10
2.1 Was sind neuronale Netze ?	10
2.2 Anwendungsgebiete neuronaler Netze.....	11
2.3 Eigenschaften neuronaler Netze.....	12
2.4 Geschichte neuronaler Netze.....	15
2.5 Biologische Neuronen	17
2.5.1 Aufbau einer Nervenzelle.....	17
2.6 Neuronen in der Simulation vs. Neuronen in der Natur.....	19
2.7 Konzepte des Konnektionismus	21
2.7.1 Aufbau und Darstellung KNN.....	21
2.7.2 Bestandteile neuronaler Netze.....	23
2.7.3 Zelltypen nach Position im Netzwerk	24
2.7.4 Aktivierungszustand.....	24
2.7.5 Ausgabefunktion	25
2.7.6 Arten von Verbindungsnetzwerken.....	25
2.7.6.1 Netze ohne Rückkopplung (Feedforward-Netze)	25
2.7.6.2 Netze mit Rückkopplungen.....	25
2.7.7 Propagierungsregel und Aktivierungsfunktion	28
2.7.8 Lernregel	28
2.8 Lernen in neuronalen Netzen	29
2.8.1 Arten des Lernens.....	29
2.8.1.1 Überwachtes Lernen.....	29
2.8.1.2 Bestärkendes Lernen	30
2.8.1.3 Unüberwachtes Lernen.....	30
2.8.2 Realisierungsmöglichkeiten von Lernprozessen in KNN	31
2.8.2.1 Hebbsche Lernregel	32
2.8.2.2 Delta-Regel	32
2.8.2.3 Backpropagation-Regel.....	32
2.8.2.4 Backpropagation mit Momentum-Term	35
3 Evolutionäre Algorithmen.....	36
3.1 Motivation	36
3.2 Was sind evolutionäre Algorithmen ?.....	36
3.3 Struktureller Aufbau evolutionärer Algorithmen.....	39
3.4 Eigenschaften genetischer Algorithmen.....	40
3.5 Entwicklung evolutionärer Algorithmen.....	41

4	Finanzprognosen	43
4.1	Motivation	43
4.2	Methoden und Probleme der Finanzanalyse	46
4.2.1	Renditegenerierungsprozesse	46
4.2.1.1	Lineare Renditegenerierungsprozesse.....	48
4.2.1.2	Nichtlineare, dynamische Renditegenerierungsprozesse ...	51
4.3	These effizienter Kapitalmärkte	53
4.4	Technische Analyse.....	55
4.5	Fundamentale Analyse	56
4.6	Nichtlineare Analyse von Finanzmärkten	56
4.6.1	NN als Instrument zur nichtlinearen Analyse von Finanzdaten.....	57
5	Einsatz NN in der Finanzwirtschaft	59
5.1	Modell von Kimoto et al.	60
5.2	Modell von SNI/SENN	61
5.3	Modell der SGZ-Bank AG	62
6	Zeitreihenprognose mittels neuronaler Netze	66
6.1	Schwierigkeiten und Probleme bei Zeitreihenprognosen.....	66
6.2	Verschiedene Architekturen neuronaler Netze zur Zeitreihenprognose ...	67
7	Eigener Ansatz.....	72
7.1	Einführung.....	72
7.2	Grundlagen	73
7.2.1	Aufteilung der Datenmenge	73
7.2.2	Partiell rekurrente neuronale Netze.....	77
7.2.2.1	Hierarchische Elman-Netze	78
7.2.3	Ein objektorientierter Ansatz für neuronale Netze.....	80
7.2.4	Evolutionäre Algorithmen.....	95
7.2.4.1	EA zur Optimierung der Netzstruktur.....	97
7.2.4.2	EA zur Optimierung der Fühlerstruktur.....	98
7.3	Die einzelnen Prognosetools	101
7.3.1	Weitere Java-Klassen	101
7.3.2	tsprogNN	107
7.3.3	tsprogNN2	108
7.3.4	tsprogNNEA.....	109
7.3.5	tsprogNNEA2.....	113
7.3.6	createprog	113
7.3.7	createprog2	114
7.3.8	Dateistrukturen	114
7.3.8.1	Dateiformat der Zeitreihendatei	114
7.3.8.2	Dateiformat der Zeitreihendatei bei Finanzmarktdaten ...	114
7.3.8.3	Dateiformat der trainierten PRNN	115
7.3.8.4	Dateiformat der Prognosedateien.....	116

7.4	Ergebnisse	117
7.4.1	Sägezahn.....	119
7.4.2	$\sin(x) + \sin(x^2)$	122
7.4.3	DAX	124
7.4.4	BP.....	128
7.4.5	IBM	131
7.4.6	JPM.....	135
7.4.7	MSFT	137
8	Zusammenfassung und Ausblick.....	140
	Anhang A	143
	Anhang B.....	145
	Abbildungsverzeichnis	156
	Tabellenverzeichnis	157
	Literaturverzeichnis.....	158

Abkürzungsverzeichnis

ANN	Artificial Neural Networks
AR	Autoregressiver (Prozess)
ARMA	Autoregressiver Moving Average (Prozess)
BAM	Bidirectional Associative Memory
BP	Backpropagation
DAX	Deutscher Aktienindex
EA	Evolutionärer Algorithmus
EMH	Efficient Market Hypothesis
GP	Genetische Programmierung
ifo	Information und Forschung
ifo Institut	Institut für Wirtschaftsforschung e.V.
KI	Künstliche Intelligenz
KNN	Künstliche neuronale Netze
NLDM	Nichtlineare dynamische Modelle
NN	Neuronale Netze
MA	Moving Average (Prozess)
MIT	Massachusetts Institute of Technology
NYSE	New York Stock Exchange
OO	Objektorientiert
PRNN	Partial Recurrent Neural Network
SENN	Software Entwicklungsumgebung für neuronale Netze
SNI	Siemens-Nixdorf Informationssysteme AG
SNNS	Stuttgarter Neuronale Netze Simulator
S&P500	Standard and Poors 500 – Aktienindex in den USA
TAR	Threshold Auto Regression (Prozess)
TOPIX	Tokyo Stock Exchange Price Index

1 Einleitung

1.1 Einführung in die Thematik

The works of nature must all be accounted good.

Cicero, *De Senectute*

Die Kapitalmarktforschung bzw. die Finanzanalyse¹ beschäftigen sich unter anderem mit Themen wie der Effizienz von Kapitalmärkten², dem Einfluss relevanter Informationen auf den Preisbildungsprozess von Wertpapieren, der am Markt gültigen Rendite/Risiko-Beziehung und Möglichkeiten bzw. Handelsstrategien zur Erzielung von signifikanten Überrenditen, bezogen auf eine als geltend betrachtete Rendite/Risiko-Beziehung.

Durch die heutigen Informationssysteme ist der Nachrichten- bzw. Informationsaustausch binnen Bruchteilen von Sekunden um den kompletten Globus möglich, was die berechtigte Frage aufwirft, ob die Finanzmärkte von heute bzw. deren Funktionsweise mit denen aus der Vergangenheit noch viele Gemeinsamkeiten haben.

Ist aus ehemals separaten Finanzmärkten ein globaler, integrierter Finanzmarkt³ entstanden, dessen Verständnis und Analyse anderer Methoden bedarf als der bisher für die separaten Märkte angewandten ?

Wie sich im Kapitel 4 dieser Arbeit noch zeigen wird, werden gerade für die Analyse integrierter Finanzmärkte neuere Verfahren⁴ im Bereich der empirischen Kapitalmarktforschung eingesetzt.

Zwei dieser neueren Verfahren (neuronale Netze und evolutionäre Algorithmen) aus dem Bereich der künstlichen Intelligenz bzw. aus dem Bereich maschinelles Lernen (siehe unten) werden im Rahmen dieser Arbeit verwendet und in einem in JAVA realisierten hybriden System zur Zeitreihenprognose für Kapitalmarktzeitreihen implementiert.

Seit den fünfziger Jahren wird versucht, Computern die Fähigkeit beizubringen, „selbständig“ zu lernen. Der Oberbegriff hierfür ist „Maschinelles Lernen“, eine Formulierung, die 1959

¹ vgl. hierzu Kapitel 4.1 „Finanzwirtschaft, Finanzanalyse, Finanzprognose“

² vgl. hierzu Kapitel 4.3

³ vgl. hierzu Kapitel 4.1 „Segmentierte Finanzmärkte, Integrierte Finanzmärkte“

⁴ Für eine Übersicht verschiedener Methoden der Finanzanalyse siehe Tabelle 4-1.

von Samuel geschaffen wurde, dem es als Erster gelang, eine ernsthafte Aufgabe durch einen Computer zu „lernen“.

Samuel wollte durch „maschinelles Lernen“ sich selbst programmierende Computer verstanden haben, was sich im nachhinein als etwas zu ehrgeizig herausstellte. Eine neuere Definition für „maschinelles Lernen“ stammt von Mitchell: „[machine learning] is the study of computer algorithms that improve automatically through experience“¹.

Genetische Algorithmen, evolutionäre Strategien und evolutionäre Programmierung sind Begriffe, die sich in den sechziger Jahren durch verschiedene Forscher herausgebildet haben, in dieser Arbeit jedoch synonym für *evolutionäre Algorithmen* verwendet werden.

Evolutionäre Algorithmen² fallen ebenso wie neuronale Netze³ unter den Begriff „Maschinelles Lernen“.

1.2 Ziel der Arbeit

Das Ziel der Arbeit war die Entwicklung eines Systems in Java, mit dem Strukturen innerhalb von Zeitreihen analysiert und zur Vorhersage zukünftiger Realisationen genutzt werden können.

Untersuchungsschwerpunkt sollten dabei Kapitalmarktzeitreihen, genauer Renditezeitreihen von Aktien bzw. Indizes, sein.

Diese Strukturanalyse der Daten wird mittels neuronaler Netze realisiert, welche die Eigenschaft besitzen, selbständig, d.h. rein durch die Präsentation der vergangenen Zeitreihe, Zusammenhänge zu „lernen“.

Für die Realisierung neuronaler Netze in Java wurde der von Joey Rogers in C++ entwickelte objektorientierte Ansatz zur Modellierung neuronaler Netze⁴ erweitert, so dass sich sogenannte hierarchische Elman-Netze⁵, eine spezielle Variante partiell rekurrenter neuronaler Netze, welche für die Zeitreihenanalyse besonders geeignet sind, implementieren lassen.

¹ vgl. [BNKF98,S.4]

² vgl. Kapitel 3

³ vgl. Kapitel 2

⁴ vgl. [Rog97]

⁵ siehe Kapitel 7.2.2.1

Für die Optimierung der genauen Netzstrukturen der Elman-Netze wird ein evolutionärer Algorithmus verwendet, der durch einen zweiten evolutionären Algorithmus zur Optimierung der Datenabgriffe¹ eines NN innerhalb der zur analysierenden Zeitreihe ergänzt wird.

Auf diese Weise entsteht das oben erwähnte hybride System zur Zeitreihenanalyse, bestehend aus einem zweistufigen EA zur Optimierung partiell rekurrenter NN.

Bei der Anwendung von Prognosevarianten, welche gute Ergebnisse in einem gewissen Zeitraum einer Zeitreihe liefern, kommt es oftmals zu der Kritik, dass diese guten Resultate eben nur in diesem bestimmten Zeitraum der speziellen Zeitreihe erzielbar sind.

Um dieser Kritik zu entgehen, sollten die hier entwickelten Prognosetools die Möglichkeit besitzen, den Lern- oder Trainingsprozess sowie den Prognoseprozess in variabel festlegbaren Zeiträumen der Zeitreihe zu ermöglichen.²

1.3 Aufbau der Arbeit

Die vorliegende Arbeit ist in 8 Kapitel gegliedert. Das dem einleitenden Kapitel 1 folgende Kapitel 2 beschäftigt sich mit der Thematik „neuronale Netze“. Nach einer kurzen Einführung, einem historischen Überblick und der Verbindung zu dem biologischem Vorbild folgt eine Darstellung des Aufbaus von simulierten, künstlichen neuronalen Netzen. Das letzte Unterkapitel 2.8 beschäftigt sich schließlich mit dem „Lernen“ in neuronalen Netzen. Es enthält eine Unterscheidung der Arten des Lernens, sowie anschließend eine formale Darstellung von einigen bedeutenden Lernregeln.

Kapitel 3 behandelt kurz den Nutzen und den Aufbau evolutionärer Algorithmen, bevor im Anschluss daran deren Eigenschaften und Entwicklung erklärt werden.

Im Kapitel 4 werden einige Grundlagen zum Bereich Finanzprognosen dargestellt. Es wird sowohl auf Renditegenerierungsprozesse als auch auf die Markteffizienzthesen von Fama, die technische und fundamentale Analyse, sowie die nichtlineare Analyse von Finanzmärkten eingegangen.

Im sich anschließenden Kapitel 5 werden drei bestehende Ansätze neuronaler Netze im Bereich der Finanzwirtschaft vorgestellt, bevor im Kapitel 6 Möglichkeiten und Probleme der reinen Zeitreihenprognose mittels NN erörtert werden.

¹ vgl. hierzu Kapitel 7.2.4.2

² siehe Thematik Aufteilung der Datenmenge Kapitel 7.2.1

In den, in Kapitel 7 vorgestellten, in JAVA realisierten Prognosetools, wird die reine Zeitreihenprognose bzw. technische Zeitreihenanalyse verwendet, im Gegensatz zu Prognosetechniken, welche verschiedene weitere (ökonomische) Faktoren benutzen. Diese Beschränkung musste gemacht werden, um nicht den Rahmen der hier vorliegenden Arbeit zu sprengen.

Nach einer Erklärung der Struktur der verwendeten neuronalen Netze, auf denen die Prognosetools aufbauen, folgt eine Darstellung eines objektorientierten Ansatzes zu Realisierung neuronaler Netze nach J. Rogers¹ in Java.

Anschließend wird der Nutzen evolutionärer Algorithmen in dem implementierten Ansatz erläutert, bevor die Prognosetools einzeln vorgestellt werden.

Nach einer Darstellung der Ergebnisse der verwendeten Prognosetools für zwei künstlich erzeugte Zeitreihen und fünf Renditezeitreihen von real existierenden Wertpapieren bzw. für einen Aktienindex schließt sich Kapitel 8 mit der Zusammenfassung der in dieser Arbeit gewonnenen Erkenntnisse und Verbesserungsmöglichkeiten des implementierten Ansatzes an.

¹ vgl. [Rog97]

2 Neuronale Netze¹

If I only had a brain.

The Scarecrow, from *The Wizard of Oz*

2.1 Was sind neuronale Netze ?

Künstliche neuronale Netze sind Systeme, deren Aufbau und Funktionsweise grob dem von Gehirnen von Säugetieren ähneln², bei denen Informationsverarbeitung durch das gleichzeitige Zusammenspiel (Parallelität) von vielen, einfachen Nervenzellen (Neuronen) stattfindet.

Diese Art der Informationsverarbeitung (große Anzahl von einfachen Verarbeitungselementen, Kommunikation durch Nachrichtenaustausch) wird auch als *Konnektionismus*³ bezeichnet.

A. Zell definiert neuronale Netze in seinem Buch „Simulation Neuronaler Netze“ wie folgt:

„Neuronale Netze (NN), oft auch als künstliche neuronale Netze (KNN) oder *artificial neural networks* (ANN) bezeichnet, sind informationsverarbeitende Systeme, die aus einer großen Anzahl einfacher Einheiten (Zellen, Neuronen) bestehen, die sich Information in Form der Aktivierung der Zellen über gerichtete Verbindungen (*connections, links*) zusenden.“ [Zell94,S.23]

Ein wesentliches Element von neuronalen Netzen ist deren Lernfähigkeit. Zusammenhänge in Daten, Klassifikationsprobleme etc. können alleine anhand von Trainingsbeispielen „gelernt“ werden, ohne dass hierzu explizit „Wissen“ implementiert werden muss.

Die Wissenschaft rund um neuronale Netze erstreckt sich von Gebieten der Medizin und Biologie, die hauptsächlich an der Simulation möglichst naturnaher Modelle von Nervensystemen interessiert sind, über Mathematik, Elektrotechnik bis hin zur Informatik, bei der die Eigenschaften der massiven Parallelität, Lernfähigkeit und Effizienz im Vordergrund stehen.

¹ Der hier dargestellte Aufbau des Kapitels über NN orientiert sich stark an [Zell94], was einen umfangreichen Überblick über die gesamte Thematik neuronaler Netze bietet.

² vgl. hierzu Kapitel 2.5 „biologische Neuronen“

³ vgl. Kapitel 2.7 „Konzepte des Konnektionismus“

2.2 Anwendungsgebiete neuronaler Netze

Die Probleme, für welche der Einsatz von neuronalen Netzen besonders geeignet ist, lassen sich grob in folgende vier Klassen einteilen¹ :

- **Klassifikation:** Verschiedene Eingangsmuster werden, nach korrektem Lernen des Netzes, auf verschiedene, definierte Ausgangsmuster abgebildet.
- **Autoassoziative Speicher:** An den Eingängen angelegte Muster werden an den Ausgängen reproduziert. Diese Speicher sind besonders beim Einsatz von Mustererkennungen interessant.
- **Heteroassoziative Speicher:** Hier werden, im Gegensatz zu den autoassoziativen Speichern, Muster erzeugt, die sich von den Eingangsmustern unterscheiden.
- **Musterfolgeerkennung / Prognose:** Hierbei wird versucht, Zusammenhänge in den verschiedenen, zeitlich aufeinanderfolgenden Eingabemustern zu erkennen.

Neuronale Netze finden ihren Einsatz in allen Bereichen der Wirtschaft. Sie stellen zu bisherigen Programmen bzw. Algorithmen eine interessante Ergänzung vor allem „in den Bereichen, wo unvollständige, vage, widersprüchliche, verzerrte, ungenaue Informationen vorhanden sind, in jenen Feldern, wo ein passender Algorithmus fehlt, oder dort, wo die nötige hohe Ausführungsgeschwindigkeit Parallelismus erfordert“ [Maz92,S.93] dar.

Folgende Anwendungsgebiete scheinen daher für neuronale Netze prädestiniert zu sein² :

- Erkennung/Klassifikation von Bildern, Texten, Stimmen, Signalen, Mustern
- Geräuschfilter bei Signalen, Sprachsynthese
- Planung/Optimierung
- Prozess- und Qualitätskontrolle
- Entscheidungshilfe
- Finanzanalyse
- Medizinische und industrielle Diagnostik

¹ vgl. [Hof93,S. 138]

² vgl. [Maz92]

- Wettervorhersage

2.3 Eigenschaften neuronaler Netze¹

Neuronale Netze haben gegenüber herkömmlichen Ansätzen viele positive Eigenschaften :

- **Lernfähigkeit:** Neuronale Netze werden zumeist nicht programmiert, sondern durch ein Lernverfahren mit einer großen Menge von Trainingsmustern trainiert. Damit sind sie anders als fest programmierte Algorithmen in der Lage, ihr Verhalten (d.h. ihre Ausgaben) geänderten Eingaben anzupassen.
- **Parallelität:** Neuronale Netze sind bereits vom Ansatz her massiv parallel und daher für eine Implementierung oder Simulation auf Parallelrechnern geeignet.
- **Verteilte Wissensrepräsentation:** Bei fast allen neuronalen Modellen ist das „Wissen“ des neuronalen Netzes in den Gewichten der Verbindungen verteilt gespeichert. Zum einen ermöglicht dies erst die hochgradig parallele Verarbeitung, zum anderen bewirkt es eine höhere Fehlertoleranz des Gesamtsystems gegenüber dem Ausfall einzelner Neuronen oder Verbindungen.
- **Höhere Fehlertoleranz:** Durch die verteilte Repräsentation können neuronale Netze eine höhere Fehlertoleranz bei Ausfall einzelner Komponenten besitzen als herkömmliche Algorithmen. Dies gilt allerdings nur, wenn die Fehlertoleranz beim Entwurf des Systems (z.B. bei der Dimensionierung des Netzes, der Kodierung der Werte und beim Lernverfahren) mit berücksichtigt wurde. Nicht jedes trainierte neuronale Netz ist automatisch fehlertolerant.
- **Assoziative Speicherung von Information:** Information wird hier inhaltsbezogen, d.h. assoziativ gespeichert und nicht adressbezogen, wie in konventionellen Rechnerarchitekturen und Programmen. Mit neuronalen Netzen ist es leicht, ein, einem zuvor eingegebenen Muster ähnliches Muster abzurufen.

¹ Die Darstellung der Eigenschaften NN ist wegen der prägnant formulierten und übersichtlichen Darstellungsweise nahezu 1:1 aus [Zell94,S. 26-28] übernommen.