To appear: 3. GI-Workshop, "Fuzzy-Neuro-Systeme '95", 15.–17.11.1995, TH Darmstadt

Abverkaufsprognose mit paralleler Backpropagation

Frank M. Thiesing, Ulrich Middelberg, Oliver Vornberger
Universität Osnabrück, Fachbereich Mathematik / Informatik
D-49069 Osnabrück, frank@informatik.uni-osnabrueck.de

Kurzfassung

Es wird der Einsatz künstlicher neuronaler Netze zur Prognose von Zeitreihen vorgestellt. Netze vom Typ Feedforward Multilayer Perceptron werden mit dem Backpropagation-Verfahren trainiert, zukünftige Abverkäufe von Artikeln einer Produktgruppe in einer Supermarktkette vorherzusagen. Um die Trainingsdauer zu verkürzen, wurde der Backpropagation-Algorithmus parallelisiert und für ein Parallelrechnersystem implementiert. Erste empirische Ergebnisse für die Bestimmung sowohl einer optimalen Netzkonfiguration als auch der Lernparameter werden diskutiert.

Abstract

In this paper artificial neural networks are adapted to time series prediction. Feedforward multilayer perceptron networks are trained by the back-propagation algorithm in order to forecast the future sales of articles of a certain group of related products in supermarkets. The back-propagation algorithm is parallelized and implemented for multi-processor computers to reduce the duration of training time. First empirical results concerning the optimal network topology and training parameters are discussed.

1 Einleitung

Zur Prognose von Zeitreihen werden immer häufiger neuronale Netze eingesetzt [1]. Als Beispiele hierfür finden sich die Vorhersage des Strom- und Fernwärmebedarfs [2, 3] sowie die Finanzdatenprognose [4]. Insbesondere der Wunsch von Unternehmen nach Reduzierung der Lagerhaltungskosten macht eine Prognose der zu erwartenden Nachfrage und des Absatzes für die Disposition von Waren oder Vorprodukten interessant.

Im vorgestellten Projekt sollen neuronale Netze für die kurzfristige Abverkaufsprognose von Produkten in einer Supermarktkette eingesetzt werden. Die Netze werden durch Präsentation der bekannten Abverkaufszahlen der Vergangenheit trainiert. Zusätzlich fließen Informationen über Preisveränderungen und Werbeaktionen mit ein. Das Training der Netze erfolgt auf einem Zeitfenster für eine ganze Gruppe verwandter Artikel, deren Abverkauf zueinander in Beziehung steht. Zur Beschleunigung des verwendeten Backpropagation-Algorithmus sind verschiedene Parallelisierungen auf Multiprozessor-Rechnern implementiert.

2 Abverkaufsprognose

Prognostiziert wird der Abverkauf von 53 Artikeln einer Produktgruppe in einem Supermarkt. Die Daten über die Anzahl der verkauften Artikel und deren Preis liegt wochenweise seit September 1994 vor. Zusätzlich gibt es Informationen über lokale Werbeaktionen, die häufig mit einer vorübergehenden Preissenkung verbunden sind. Diese Aktionen haben signifikanten Einfluß auf den Absatz des betreffenden Artikels. Die wöchentlichen Abverkaufszahlen und der Wochendurchschnittspreis eines exemplarischen Artikels sind in Bild 1 dargestellt.

Zur Prognose werden Feedforward Multilayer Perceptron (MLP) Netzwerke mit einer verborgenen Schicht eingesetzt (siehe Bild 2), die mit Hilfe der Backpropagation-Lernregel trainiert werden. Für die Vorhersage des Abverkaufs in einer zukünftigen Woche werden in der Eingangsschicht die Daten von n vergangenen Wochen präsentiert.

3 Datenmodellierung

Eine effiziente Vorverarbeitung der anfallenden Daten ist notwendig, um sie dem Netz eingeben zu können. Insbesondere muß aufgrund der gewählten Implementation eine Skalierung der unterschiedlichen Werte auf das Intervall $[0, 1]$ erfolgen. Gegeben seien die relevanten Informationen für T Wochen. Mit den folgenden Definitionen

$$\begin{aligned} ADV_i^t &:= \text{Anzahl der Aktionstage für Artikel } i \text{ in Woche } t \\ SAL_i^t &:= \text{Abverkauf von Artikel } i \text{ in Woche } t \text{ und} \\ MAXSAL_i &:= \max_{1 \le t \le T} \left\{ SAL_i^t \right\} \end{aligned}$$

werden für jeden Artikel i und jede Woche t folgende Netzeingaben berechnet:

$$\begin{aligned} adv_i^t &:= \frac{ADV_i^t}{6} \text{ (Aktionstage von 6 Werktagen)} \\ pri_i^t &:= \left. \begin{cases} 1.0 & : \text{ Preis steigt} \\ 0.5 & : \text{ Preis bleibt gleich} \\ 0.0 & : \text{ Preis fällt} \end{cases} \right\} \text{in Woche } t \end{aligned}$$

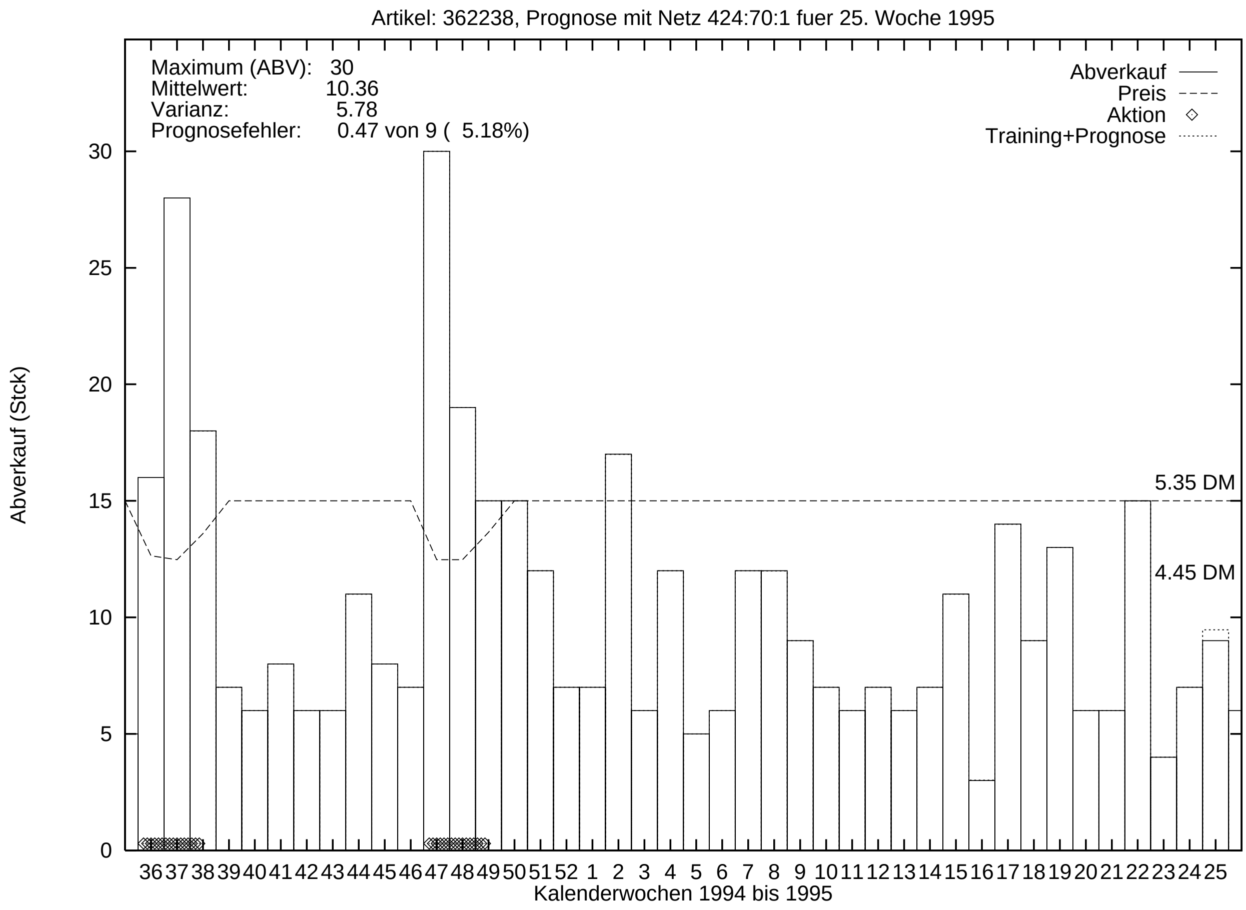

Bild 1: Abverkauf, Preis und Aktionen eines ausgewählten Artikels

$$sal_i^t \;:=\; \frac{SAL_i^t}{MAXSAL_i} \cdot 0.8$$

Für jeden Artikel i und jede Woche t in der Vergangenheit wird der folgende dreidimensionale Vektor definiert:

$$vec_i^t := \left(adv_i^t, pri_i^t, sal_i^t\right)$$

Für eine zukünftige Woche t wird dieser Vektor um den noch unbekannten Abverkauf reduziert:

$$\widehat{vec}_i^t := \left(adv_i^t, pri_i^t\right)$$

Um den Abverkauf eines Artikels für eine Woche t zu trainieren bzw. vorherzusagen wird ein Zeitfenster von n vergangenen Wochen verwendet. Dadurch ergibt sich für jeden Artikel i der Eingabevektor

$$input_i^t := \left(vec_i^{t-n}, vec_i^{t-n+1}, ..., vec_i^{t-1}, \widehat{vec}_i^t\right).$$

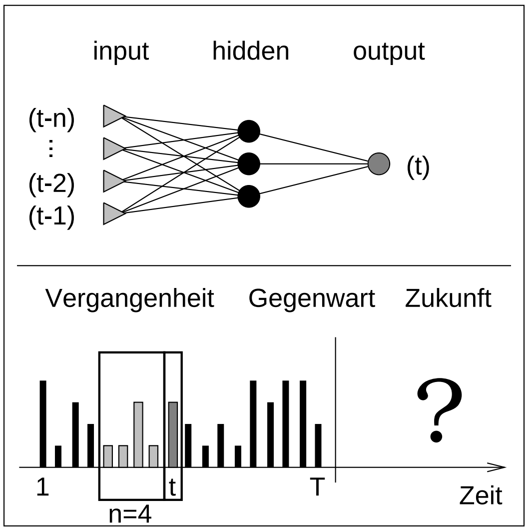

Bild 2: Zeitreihenprognose mit Feedforward Multilayer Perceptron

Da die betrachteten Artikel einer Produktgruppe angehören, korreliert ihr Absatz: Ein Absatzplus bei einem Artikel geht im allgemeinen zu Lasten anderer Artikel. Deshalb werden die bekannten Informationen aller p Artikel zu einem Eingabevektor konkateniert:

$$INPUT^t := \left(input_1^t, input_2^t, ..., input_p^t\right)$$

Mit dem Abverkauf sal_i^t jedes Artikels i in der Woche t als Sollwert des Ausgabeneurons wird ein Netz trainiert. Somit ergibt sich für das i-te der p Netze, das das Abverkaufsverhalten des i-ten Artikels lernen soll, eine Trainingsmenge mit folgenden Trainingspaaren (vgl. Bild 2):

$$\left(INPUT^t, sal_i^t\right) \text{ für } n \leq t \leq T$$

Zur Prognose des noch unbekannten Abverkaufs von Artikel i in der zukünftigen Woche $T+1$ wird das trainierte i-te Netz mit folgendem Eingabevektor abgefragt:

$$INPUT^{T+1}$$

Als Ausgabewert des Netzes wird der Wert sal_i^{T+1} erwartet, der zurücktransformiert werden muß zum Wert für den prognostizierten Abverkauf von Artikel i in der Woche $T+1$.

4 Ergebnisse

Zur Bestimmung der geeigneten Konfiguration des MLP-Netzwerks wurden zahlreiche Experimente unternommen. Folgende Parameter wurden variiert:

1. Größe des Zeitfensters: $n = 2$ bzw. $n = 3$
2. Zahl der verborgenen Neuronen: ein Sechstel bzw. ein Zwölftel der Anzahl der Eingangsneuronen
3. Lernrate und Momentumterm

Wir verwenden in der Eingangsschicht der Netze die Zeitreihen von 53 Artikeln. Die Topologie wird beschrieben durch `(#Eingangsneuronen:#Verdeckte Neuronen:#Ausgangsneuronen)`. Entsprechend der Anwendung für die Absatzprognose im Supermarkt werden zur Bestimmung der Prognosegüte die gegebenen Daten aufgeteilt: Als Trainingsmenge werden die Daten der Wochen 36/1994 bis 24/1995 verwendet. Um die Generalisierungsfähigkeit des Netzes überprüfen zu können, verwenden wir die 25. Woche 1995 als Testmenge. Damit ergeben sich 39 Trainingspaare für $n = 2$ bzw. 38 für $n = 3$ sowie jeweils ein nicht trainiertes Paar als Testmenge.

Durch zahlreiche Experimente haben sich eine Lernrate von 0.25 und ein Momentumterm von Null als gute Parameterwahl herausgestellt. Beim Vergleich der betrachteten Netztopologien zeigten sich die größeren Netze überlegen: Die Generalisierungsfähigkeit mit einem Sechstel der Anzahl der Eingangsneuronen in der verdeckten Schicht erweist sich hier als besser.

Bild 3 stellt den Fehlerverlauf mit dieser Parametereinstellung während des Trainings von 1000 Epochen für den in Bild 1 gezeigten Artikel dar. Aufgetragen sind die Fehlerverläufe für Trainings- und Testmenge für zwei Netze mit $n = 2$ bzw. $n = 3$. Es zeigt sich, daß beide Netze den Fehler auf der Trainingsmenge schnell reduzieren, während auf der Testmenge das Netz mit zwei Wochen Vergangenheit den kleineren Prognosefehler liefert.

Eine konkrete Prognose für die 25. Woche 1995 mit dem Netz 424:70:1 kann in Bild 1 als gepunktete Line abgelesen werden: Der Prognosefehler ist kleiner als ein Stück.

Tabelle 1: Trainingszeiten unterschiedlicher Netze auf einer SPARC 20-50MHz

Netz Konfiguration	424:35:1	424:70:1	583:50:1	583:100:1
n	2	2	3	3
# Trainingspaare	39	39	38	38
1000 Epochen	489 Sek	1018 Sek	907 Sek	1815 Sek

Im Vergleich zu früheren Prognosen zeigen die aktuellen Ergebnisse eine Verbesserung der Prognosequalität begründet durch die längere Historie seit September 1994. Die Rechenzeit zum Training der Netze ist Tabelle 1 zu entnehmen.

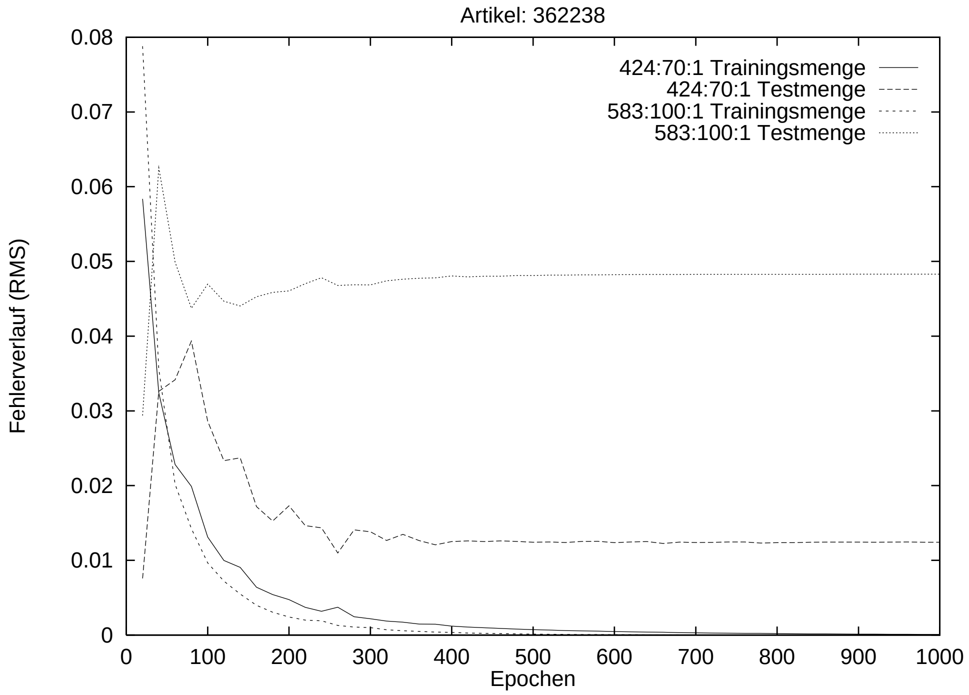

Bild 3: root mean square error während des Trainings

5 Parallelisierung

Zur Verringerung der Trainingszeiten des Backpropagation-Algorithmus ist dieser auf verschiedene Arten parallelisiert worden. Neben einer Parallelisierung des Batch-Learning, bei der die Trainingsmenge verteilt trainiert wird und die Gewichts- und Schwellwertkorrekturen nach jeder Epoche unter den Prozessoren ausgetauscht werden, sind zwei Parallelisierungen des Online-Trainings durchgeführt worden. Hierbei wird schichtweise im MLP-Netzwerk die Zuständigkeit für die Neuronen unter den Prozessoren verteilt, so daß die Matrix-Vektor-Operationen parallel durchgeführt werden können. Die eine Variante nach Morgan et.al. [5] verwendet einen kommunikationsintensiven Algorithmus, während die Parallelisierung nach Yoon et.al. [6] den Kommunikationsaufwand zu Lasten von Speicher- und Rechenbedarf reduziert.

Die parallelen Algorithmen sind mit Hilfe der Message Passing Systeme Parix und PVM auf Parsytec Parallelrechnern implementiert worden. Bild 4 stellt für die Implementation auf einem Parsytec GC mit PowerPC Prozessoren die zeitliche Beschleunigung des Trainings durch die Parallelisierung (Speedup) dar. Dabei zeigt sich die gute Parallelisierbarkeit des Batch-Learning. Bei der Parallelisierung des Online-Trainings ist der Ansatz von Yoon dem von Morgan leicht überlegen, bei großen Prozessorzahlen skalieren beide Parallelisierungen wegen des hohen Kommunikationsaufwands nicht mehr.

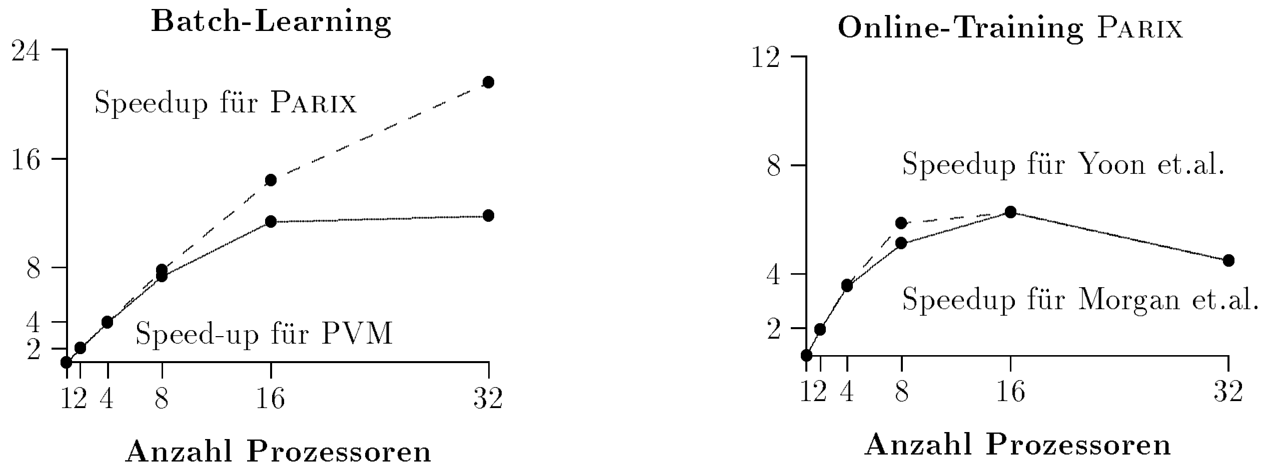

Bild 4: Speedups für Parsytec GC mit PowerPC Prozessoren

6 Zusammenfassung und Ausblick

Neuronale Netze sind geeignet zur Prognose von Absatzzahlen in Supermärkten. Die stark schwankenden und von vielen Einflußfaktoren abhängigen wöchentlichen Verkaufszahlen können von MLP-Netzen generalisiert und für das kommende Zeitintervall recht präzise vorhergesagt werden. Die dabei anfallenden Trainingszeiten sind durch geeignete Parallelisierungen verringert worden.

Ein erhebliches Potential zur notwendigen Verkürzung der Trainingszeiten birgt die Verkleinerung des Inputvektors. Wurden bisher die Zeitreihen aller verwandten Produkte, die möglicherweise Einfluß auf den Abverkauf eines Artikel haben konnten, herangezogen, so soll in Zukunft durch Korrelationsanalysen im Vorfeld erreicht werden, daß lediglich die signifikanten Daten zum Training ausgewählt werden.

Außerdem gibt es neben den bisher benutzten Zeitreihen offensichtlich noch weitere Einflußfaktoren. Erste Untersuchungen haben gezeigt, daß Informationen über Feiertage, Saison und Urlaub ebenfalls einbezogen werden sollten, um die Prognosegüte zu verbessern.

Diese Daten auszuwählen sowie in Zeitreihen geeignet zu modellieren und dabei die Zahl der Eingangsneuronen klein zu halten wird das Ziel der nächsten Anstrengungen sein. Als Ergebnis soll ein System entstehen, das vor Ort im Supermarkt die Disposition unterstützt und sich durch ständiges Training flexibel veränderten Absatzentwicklungen anpaßt.

Schriftum

[1] V.R. Vemuri, R.D. Rogers. *Artificial Neural Networks – Forecasting Time Series.* IEEE Computer Society Press 5120-05, 1994.

[2] U. Ahle. *Künstliche Neuronale Netze für die Prognose des Fernwärmebedarfs.* in B. Reusch (Hrsg.) *Fuzzy Logik – Theorie und Praxis*, S. 292-299, Springer Verlag 1994.

[3] H. Schreiber, S. Heine. *Einsatz von Neuro-Fuzzy-Technologien für die Prognose des Elektroenergieverbrauches an „besonderen" Tagen.* in B. Reusch (Hrsg.) *Fuzzy Logik – Theorie und Praxis*, S. 300-307, Springer Verlag 1994.

[4] D. Würtz, C. de Groot. *Wechselkursanalyse gestützt auf ein „neuronales" Informationsverarbeitungssystem.* Intelligente Software Technologie, Heft 2/1992, S. 44-45, Oldenbourg Verlag.

[5] N. Morgan, J. Beck, P. Kohn, J. Bilmes, E. Allman, J. Beer. *The Ring Array Processor: A Multiprocessor Peripheral for Connectionist Applications.* Journal of Parallel and Distributed Computing 14, S. 248-259, Academic Press Inc., 1992.

[6] H. Yoon, J.H. Nang, S.R. Maeng. *A distributed backpropagation algorithm of neural networks on distributed-memory multiprocessors.* Proceedings of the 3rd symposium on the Frontiers of Massively Parallel Computation, S. 358-363, IEEE 1990.