Forecasting Sales Using Neural Networks

Frank M. Thiesing and Oliver Vornberger

Department of Mathematics and Computer Science
University of Osnabrück
D-49069 Osnabrück, Germany
E-Mail: frank@informatik.uni-osnabrueck.de

Abstract. In this paper, neural networks trained with the back-propagation algorithm are applied to predict the future values of time series that consist of the weekly demand on items in a supermarket. The influencing indicators of prices, advertising campaigns and holidays are taken into consideration. The design and implementation of a neural network forecasting system is described that has been installed as a prototype in the headquarters of a German supermarket company to support the management in the process of determining the expected sale figures. The performance of the networks is evaluated by comparing them to two prediction techniques used in the supermarket now. The comparison shows that neural nets outperform the conventional techniques with regard to the prediction quality.

1 Introduction

A central problem in science is predicting the future of temporal sequences. Examples range from forecasting the weather to anticipating currency exchange rates. The desire to know the future is often the driving force behind the search for laws in science and economics.

In recent years many sophisticated statistical methods have been developed and applied to forecasting problems [1], however, there are two major drawbacks to these methods. First for each problem an individual statistical model has to be chosen that makes some assumptions about underlying trends. Second the power of deterministic data analysis can be exploited for single time series with some hidden regularity (though strange and hard to see but existent), however, this approach fails for multidimensional time series with mutual non-linear dependencies.

As an answer to the weakness of statistical methods in forecasting multidimensional time series an alternative approach gains increasing attraction: neural networks [2]. The practicability of using neural networks for economic forecasting has already been demonstrated in a variety of applications, such as stock market and currency exchange rate prediction, market analysis and forecasting time series of political economy [3, 4, 5, 6].

The approaches are based on the idea of training a feed-forward multi-layer network by a supervised training algorithm in order to generalize the mapping between the input and output data and to discover the implicit rules governing the movement of the time series and predict its continuation in the future. Most of the proposals deal with one or only few time series.

There are two main streams in the manner of presenting the data to the nets. In *explanatory* forecasting the values of several different but interesting economic indicators at time t are used as the components of the input vector during the training and the value at time $t+1$ of the time series to be predicted as the corresponding desired output. This approach is based on the assumption that the development of various phenomena in the economy are essential to study the behavior of the time series in discussion.

In *time series* prediction forecasting is realized by treating n successive past values of the time series in discussion ending up at time t as an input vector and the value at time $t+1$ as the corresponding desired output. This technique processes the times series in a sliding window of width n. The underlying assumption of this procedure is that any information that is necessary to predict the future behavior is hidden in the time series only.

In this paper, neural networks trained with the *back-propagation* algorithm [7] are applied to predict the future values of 20 time series that consist of the weekly demand on items in a German supermarket. An appropriate network architecture will be presented for a mixture of both explanatory and time series forecasting. Unlike many other neural prediction approaches described in the literature, we compare the forecasting quality of the neural network to two prediction techniques currently used in the supermarket. This comparison shows that our approach produces good results.

2 Time Series Considered

The times series used in this paper consist of the sales information of 20 items in a product group of a supermarket. The information about the number of items sold and the sales revenue are on a weekly basis starting in September 1994. There are important influences on the sales that should be taken into consideration: advertising campaigns sometimes combined with temporary price reductions; holidays shorten the opening hours; the season has an effect on the sales of the considered items.

We take the sales information, prices and advertising campaigns from the cash registers and the marketing team of the supermarket. The holidays are calculated. For the season information we use the time series of the turnover sum in DM of all items of this product group as an indicator. Its behavior over a term of 19 months is shown in figure 1.

We use feed-forward multilayer perceptron (MLP) networks with one hidden layer together with the back-propagation training method. In order to predict the future sales the past information of n recent weeks is given in the input layer. The only result in the output layer is the sale for the next week.

Due to the purchasing system used in the supermarket there is a gap of one week between the newest sale value and the forecasted week. In addition the pricing information, advertising campaigns and holidays are already known for

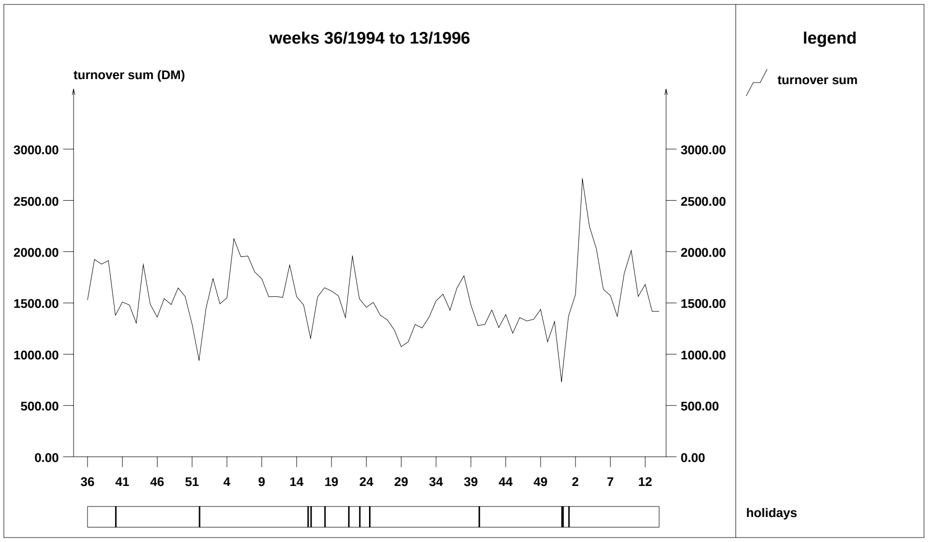

Fig. 1. turnover sum in DM of the product group September 1994 to March 1996

the future, when the forecast is calculated. This information is also given to the input layer as shown in figure 2.

3 Preprocessing the Input Data

An efficient preprocessing of the data is necessary to input it into the net. In general it is better to transform the raw time series data into indicators that represent the underlying information more explicitly. Due to the sigmoidal activation function of the back-propagation algorithm the sales information must be scaled to $]0,1[$. The scaling is necessary to support the back-propagation learning algorithm [8]. We tested several scalings (z_t) for the sale and the turnover time series $x = (x_t)$:

$$z_t = \frac{x_t - \min(x)}{\max(x) - \min(x)} \cdot 0.8 + 0.1 \qquad \text{resp.}$$
$$z_t = \frac{x_t - \mu}{c \cdot \sigma} + 0.5$$

where min and max are the minimum and maximum values of time series x and μ and σ are the average and the standard deviation. c is a factor to control the interval of the values.

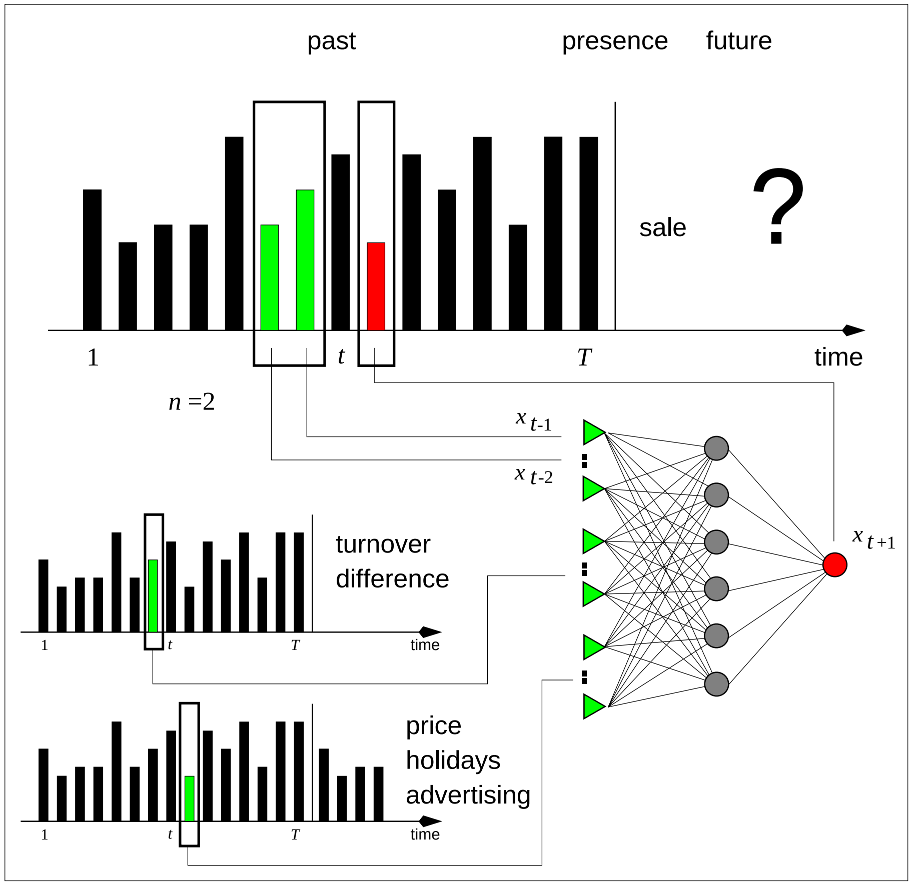

Fig. 2. input and output of the MLP

For the prices the most effecting indicator is the price change. So the prices are modeled as follows:

$$pri_t := \left\{ \begin{array}{rcl} 0.9 & : & \text{price increases} \\ 0.0 & : & \text{price keeps equal} \\ -0.9 & : & \text{price decreases} \end{array} \right\} \begin{array}{l} \text{within} \\ \text{week } t \end{array}$$

For both the time series of holidays and advertising campaigns we tested binary coding and linear aggregation to make them weekly. Their indicators are:

$$y_t := \left\{ \begin{array}{lcl} 0.9 & : & \text{if there is a holiday resp. advertising within week } t \\ 0.0 & : & \text{otherwise} \end{array} \right\} \text{ resp.}$$

$$y_t := \frac{\text{number of advertising resp. holidays within week } t}{6} \quad \begin{array}{c}\text{(normalized number}\\ \text{of special days}\\ \text{within week } t\text{)}\end{array}$$

4 Experimental Results

To determine the appropriate configuration of the feed-forward MLP network several parameters have been varied:

- modeling of input time series
- width of the sliding time window
- the number of hidden neurons
- interval of the initial random weights
- adapted training rate and momentum
- number and selection of validation patterns
- dealing with overfitting

To evaluate the efficiency of the neural network approach several tests have been performed. Table 1 compares the prediction error of a naive ("Naive") and a statistical prediction method ("MovAvg") to the successive prediction by neural networks ("Neural").
We reached good results for $n = 2$ recent values of the sale time series in the sliding window. The other inputs are, one neuron each: both the difference of the sale ($x'_t = x_t - x_{t-1}$) and the turnover of the whole group of items for the last week and the holiday, advertising and pricing information for the week to be predicted.
Thus, for each item a net with 7 input neurons and 4 hidden neurons is trained for a one week ahead forecast with a gap of one week. We reached better results with the binary scaling for the holiday and advertising time series and the σ-μ-scaling for sales. The learning rate of the back-propagation algorithm was set to 0.3 with a momentum of 0.1. The initial weights were chosen from [-0.5, 0.5] by chance. The training was validated by 12 patterns and the stopped at the minimum error.

4.1 Naive Prediction

The naive prediction method uses the last known value of the time series of sales as the forecast value for the future. In our terms: $\hat{x}_{t+1} := x_{t-1}$. This forecasting method is often used by the supermarket's personnel.

4.2 Statistical Prediction

The statistical method is currently being used by the supermarket's headquarters to forecast sales and to guide personnel responsible for purchasing. It calculates the moving average of a maximum of nine recent weeks, after these sale values have been filtered from exceptions and smoothed.

Table 1. prediction error: RMSE/Mean

Item	Mean	Neural	MovAvg	Naive
036252	7.79	1.037	1.217	1.181
078924	8.97	1.084	1.409	1.551
180689	9.92	0.973	1.243	1.231
215718	12.73	0.468	0.601	0.612
215732	12.97	0.325	0.402	0.395
215749	8.63	0.485	0.471	0.535
228558	17.40	0.481	0.709	0.923
229104	63.88	0.431	0.591	0.649
289573	8.29	0.992	1.154	1.325
304962	15.35	0.971	1.267	1.523
341110	6.79	0.672	0.815	0.755
341127	5.79	0.953	1.114	1.293
362238	10.88	1.221	1.573	2.138
372206	9.69	3.119	3.253	4.513
392785	19.79	0.513	0.624	0.534
399883	11.47	0.407	0.530	0.444
468978	23.99	0.263	0.318	0.387
468985	17.19	0.942	1.135	1.420
567411	3.95	0.586	0.569	0.623
852234	7.75	0.891	1.125	1.211
Average	—	0.841	1.006	1.162

4.3 Comparison of Prediction Techniques

To measure the error the root mean squared error (RMSE) is divided by the mean value ("Mean") of the time series. The results are calculated for the successive prediction of the 22 weeks 44/1995 to 13/1996. In this weeks there are influences of many campaigns and Christmas holidays.

Based on the information in table 1 the naive approach is outperformed by the two other methods. For 18 of the 20 items the prediction by the neural network is better than the statistical prediction method.

A close inspection of the times series favored by the statistical approach shows that these are very noisy without any implicit rules that could be learned by the neural network. Especially item 567411 has an average weekly sale of less than 4 items.

Figure 3 shows the predicted values for item 468978 calculated by the statistical and neural approach. The price, advertising and holiday information is included in this figure.

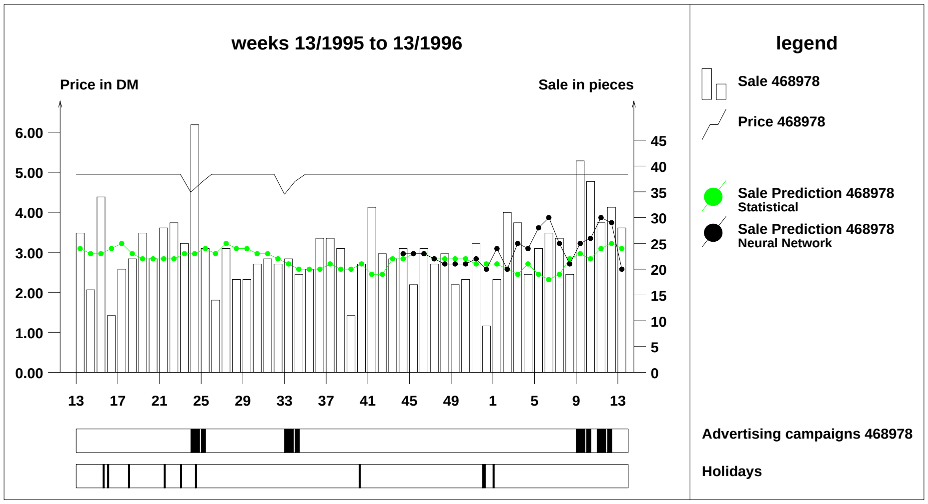

Fig. 3. sale prediction for an item by statistical and neural approach

5 Conclusions and Future Research

For a special group of items in a German supermarket neural nets have been trained to forecast future demands on the basis of the past data augmented with further influences like price changing, advertising campaigns and holiday season information. The experimental results show that neural nets outperform the naive and statistical approaches that are currently being used in the supermarket.

In contrast to many other neural prediction approaches our procedure preprocesses the data of all kinds of time series in the same manner and uses the same network architecture for the prediction of all 20 time series of sales. The parameter optimization is based on all of the time series instead of on one special item.

The program runs as a prototype and handles only a small subset of the supermarket's inventory. Future work will concentrate on the integration of our forecasting tool into the whole enterprise data flow process. Since a huge number of varying products have to be managed a selection process has to be installed that discriminates between steady time series suitable for conventional methods and chaotic candidates which will be processed by neural nets.

The prototype is part of an automatical forecasting system that is able to take the raw data, do the necessary preprocessing, train the nets and produce an appropriate forecast. The next steps will be the development of additional adaptive transformation techniques and methods to test the significance of inputs which can be used to reduce the complexity of the nets.

In addition other training algorithms for neural networks will be compaired. This will be done by the *Stuttgart Neural Network Simulator SNNS*[9].

References

1. Weigend A.S., Gershenfeld, N.A., *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, 1994.
2. Rojas, R., *Neural Nets*, Springer, 1996.
3. Schöneburg, E., "Stock Price Prediction Using Neural Networks: An Empirical Test," *Neurocomputing*, 2, 1, 1991.
4. Refenes A.N., Azema-Barac M., Chen L., Karoussos, S.A., "Currency Exchange Rate Prediction and Neural Network Design Strategies," *Neural Computing & Applications*, 1(1) pp. 46–58, 1993.
5. Chakraborty, K., Mehrotra, K., Mohan, C.K., Ranka, S., "Forecasting the Behaviour of Multivariate Time Series Using Neural Networks," *Neural Networks*, Vol. 5, pp. 961–970, 1992.
6. Freisleben, B., Ripper, K., "Economic Forecasting Using Neural Networks," *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Vol. 2, pp. 833–838, Perth, WA., 1995.
7. Vemuri, V.R., Rogers, R.D., *Artificial Neural Networks – Forecasting Time Series*, IEEE Computer Society Press, 1994.
8. Rehkugler, H., Zimmermann, H.G., *Neuronale Netze in der Ökonomie (in German; Neural Networks in Economics)*, Verlag Vahlen, München 1994.
9. *SNNS Stuttgart Neural Network Simulator User Manual, Version 4.1*, University of Stuttgart, Report No. 6/1995.

Bernd Reusch (ed.): Computational Intelligence, 5. Fuzzy Days, Dortmund, 28.–30.4.1997, pp. 321–328, Springer LNCS 1226, 1997