Transactions on Information and Communications Technologies vol 19 © 1998 WIT Press, www.witpress.com, ISSN 1743-3517

Forecasting of the German stock index DAX with neural networks: Using daily data for experiments with input variable reduction and a modified error function

Wolfgang Siegler
HYPO-BANK AG
Research Department
80278 München
E-Mail: wolfgang.siegler@handel.hypo.de
Elmar Steurer
DAIMLER-BENZ AG
Forschung und Technik - F3S/E
89013 Ulm
E-Mail:elmar.steurer@dbag.ulm.daimlerbenz.com

Abstract

Using neural networks for the prediction of economic time series still involves many problems. Examples for using neural networks in financial market applications are de Groot (1993), Baun (1997) and Burgess (1996). In these studies neural networks were successfully applied. Intensive work has been done regarding data transformation and the selection of an appropriate topology for neural networks. By using daily data of the German stock index DAX this study shows that:
1) Principal Component Analysis is not an appropriate technique for input variable reduction.
2) The Usage of a modified mean squared error as error function leads to significantly better results.

1 Introduction

Investigations of Principal Component Analysis (PCA) as a technique for the reduction of the number of input variables were done by de Groot (1993), Utans et al. (1997) and by Thomason (1996a, 1996b). Utans et al. investigate four exchange rates with hourly data and nine exchange rates with daily data. By using the Hurst exponent they conclude that PCA leads to improved forecasting models. Thomason uses PCA for the reduction of neural network input variables. He forecasts daily closing prices of the S&P 500 Index. His results indicate that PCA is appropriate for certain financial forecasting problems. Taken together, Utans' et al. and Thomason's results suggest that the usage of PCA for the prediction of the German stock index DAX is a reasonable method for input variable reduction. De Groot has a similar opinion:„... we prefer preprocessing of data leading to less parameters for the network rather than the opposite, which is common in the neural network literature (de Groot (1993), pg. 40).

A problem when coping with forecasting financial time series is connected with the noise inherent in financial data. This causes the neural network to tend to adapt to the noise instead of learning the important structures of the data. By using a modified mean squared error we try to dampen this noise.

2 Motivation

For training neural networks an error function has to be used. A commonly chosen function is the mean squared error (MSE). A disadvantage of this function is that point predictions which imply the correct signal (i.e. buy or sell) can lead to a higher error than forecasts giving a wrong signal. At the same time, when using a pure signal as output (i.e. either +1 or -1), weak or even coincidental historic price movements are extremely overemphasized during the training phase. Thus, in both cases the neural network cannot be expected to give good buy or sell signals. Therefore a compromise has to be made. In contrast to Baun (1997) we did not use a profit maximization function because with this, huge random price movements dominate the training of the neural network. Thereby the true functional relationships are biased. We tried to get a continuous signal function which is relatively insensitive to deviations lying apart from the buy/sell-threshold. This causes large errors when the deviations are placed around this threshold. Therefore we chose the 3rd root (shown in Figure 1) as transformation for the output:

$$output_{new} = \sqrt[3]{output_{old}}$$

Training time for neural networks is increasing exorbitantly with a growing number of input and hidden nodes. Thus, testing Principal Component Analysis (PCA) as a form of input variable reduction was another aim of our investigations. Figure 2 illustrates the usage of PCA as a pre-processing step in training neural networks. PCA is a linear method and this can put our method in question. Important nonlinearities inherent in the data may be removed by reducing the number of the principal components. Moreover, PCA is assuming that the variances are constant. This is not the case in financial time series.

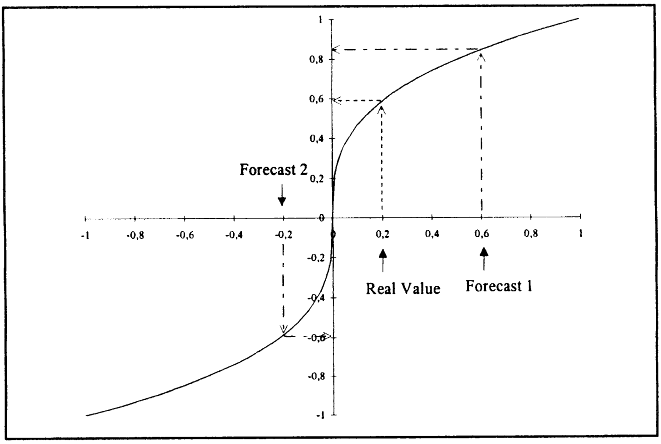

Fig. 1: 3rd root as output transformation

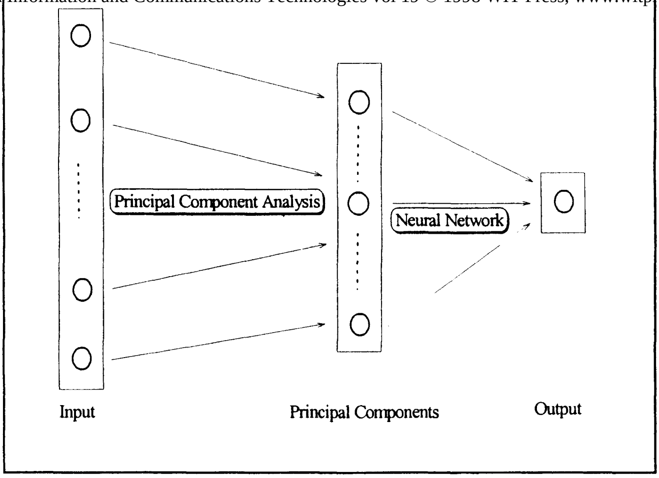

Fig. 2: Application of PCA for Input Variable Reduction

3 The data and the pre-processing

The dataset consists of 1000 daily observations starting on January 17, 1992 and ending on November 17, 1995. The whole period is subdivided into three periods: training period (first 800 observations), validation period (next 100 observations) and generalization period (last 100 observations). For our investigations we used daily closing prices of the German Stock Index DAX, the Dow Jones Industrial Average, the US-Dollar/DEM currency exchange rate, the 3-months interest rate FIBOR and the German Bond Index REX. All of them were used for inputs with time lags from 1 to 4 days. Moreover, we calculated Volatility, Relative Strength Index (over a period of 5 days) and Moving Averages (over a period of 5 and 20 days) from the DAX. An additional input was given by the Volume of the DAX measured by value. Added together, we got 25 inputs in total. They are listed in Table 1. The reason for the selection of the fundamental series was to capture the influence of the (suspected) most important factors. Time lags from 1 to 4 days should help the neural network to recognize complex patterns. By showing not only the DAX pattern but also other highly correlated series patterns to the neural network, we tried to dampen the influence of coincidental price movements. With this we wanted to achieve more stabilized results for the pattern recognition. The calculated series and the volume were used to represent technical influences.

Transactions on Information and Communications Technologies vol 19 © 1998 WIT Press, www.witpress.com, ISSN 1743-3517

Pre-processing of the data was done by calculating the differences of the logarithms of the original series. From the obtained series we subtracted the mean and then divided it by the standard deviation in order to receive a normalization of the variables (Refenes (1992)):

$$x_{new} = \frac{x_{old} - \mu}{\sigma}$$

	Lag 1	Lag 2	Lag 3	Lag 4
DAX	x	x	x	x
Dow Jones Index	x	x	x	x
US-Dollar/DEM	x	x	x	x
FIBOR	x	x	x	x
REX	x	x	x	x
Volatility 5	x			
Volume	x			
Relative Strength Index 5	x			
Moving Average 5	x			
Moving Average 20	x			

Table 1: Input Variables

The output series was additionally transformed as described in section 2. When the 3rd root is regarded as such, it is necessary for a comparison to use the 'regular' output in addition.

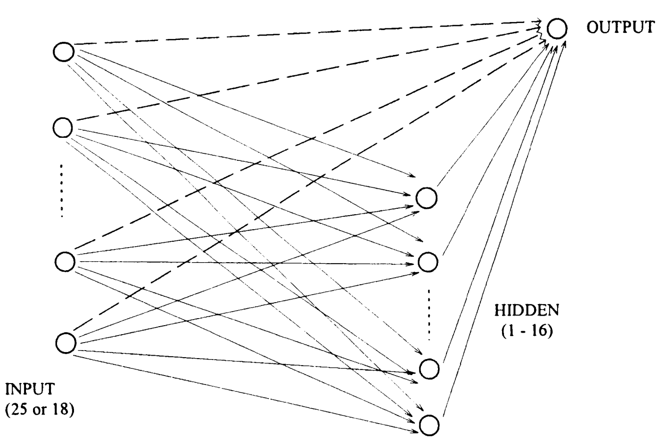

Fig. 3: Network Topologies

4 Results

Investigations have been made with the software SNNS (1995). The chosen network was a feed-forward network with one hidden layer and with input-output (i.e. linear) connections. The output layer consists of one node. The applied learning algorithm was Rprop (Resilient backpropagation). The number of hidden nodes was varied from 1 to 16. It was the only distinction between the different networks. Their topologies are shown in Figure 3. We used the stop training method to determine the network weights (see for more details e. g. de Groot (1993)). For each network topology (i.e. number of hidden nodes) we started a series of 30 training runs and calculated the mean over their MSEs. The topology with the smallest mean was classified as 'best network.' So the 'best network' consists of 30 independent networks representing a portfolio of traders (Burgess (1996)). The reasons for taking the average of 30 networks was to stabilize the results of the networks. They are usually very sensitive to the (randomly chosen) weight initialization. This fact leads to overfitting effects caused by the neural Network. Thus, the averaging method used in this study can be considered as an alternative to tackle overfitting. Averaging of the neural network forecasts is based on the idea of the combination of predictions. But up to date little has been published on techniques for combining models. Early researchers in the statistical field (e. g. Bates / Granger (1969), Dickinson (1973)) had noted the strong performance of averages of forecasts. Furthermore a good overview of the published literature to this feature is given in Clemen (1989). The basic idea is to combine different uncorrelated forecasts to an average forecast: „Thus we have nothing to lose by combining and much to gain." (Diebold (1988b), pg. 77). The link between the statistically motivated approach and the neural network methodology was drawn by de Groot (1993). He proposed the combination of different neural networks in terms of topology or input variables to achieve a set of different model values. These outcomes are combinated by averaging to avoid the overfitting of one single neural network: "Now the strategy has to be a combination of models which are as uncorrelated as possible." (de Groot (1993), pg. 46).

Transactions on Information and Communications Technologies vol 19 © 1998 WIT Press, www.witpress.com, ISSN 1743-3517

4.1 Input variable reduction by using PCA

In order to achieve a comparable basis we decorrelated the 25 input series by applying PCA. The received 25 eigenseries were called model A. We found that 18 components explained more than 90% of the variance, as illustrated in Figure 4. This effect was disappointing because one could have expected less variables to be responsible for this amount of variability. Moreover, the amount of explained variance of 'neighboring' components decreased slowly. Nonetheless, we trained this second model (model B) with the first 18 eigenseries.

The performance of the two models was measured by the profit earned from their 'best network' by going long and short the DAX over the generalization period (100 days). The topology with one hidden node turned out to be the best for each model. The results can be seen in Table 2. Model A made a profit of 13.02%, while model B gained only 5.69%. The difference of 7.33% between the two models suggests that even a small reduction in the number of input variables leads to a large decrease in performance.

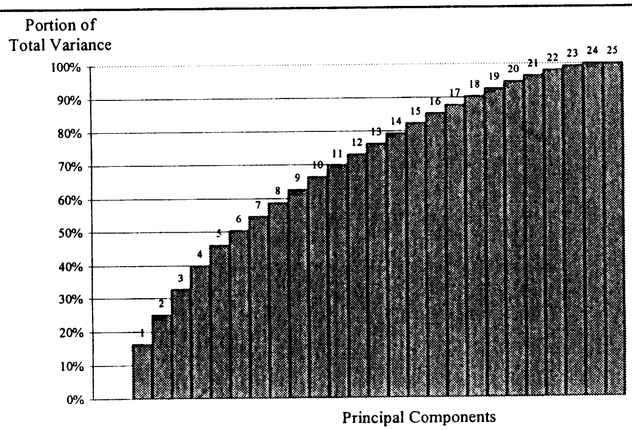

Fig. 4: Portion of Total Variance Explained by Number of Principal Components

4.2 Modified error function

Investigations and comparisons concerning the 'regular' and the modified error function were also made. We used the 25 input series without any decorrelation. The transformation for the output series mentioned in section 2, was applied on model C, whereas no such transformation was applied on model D ('regular' output).

For both models the performance was measured in the same way as it was done in section 4.1. Each 'best network' required only one hidden node. The profit of model C was 11.86% and of model D 7.82%. The difference in the profit of the two models is 4.04%. This suggests that the modified error function results in an improvement in performance.

	Profit
Model A	13.02 %
Model B	5.69 %
Model C	11.86 %
Model D	7.82 %

Table 2: Performance over the Generalization Period

5 Conclusion

In this study we measured the decrease in the performance of forecasting daily financial data. PCA was used for reducing the number of input variables. The achieved reduction in the number of input variables was disappointing with regard of the decrease in performance which was relatively huge. The results suggest that important information must have got lost in this process of input variable reduction. It further indicates that a big amount of useful information is hidden in the data as nonlinearities. That is why it can get lost by applying PCA which is a pure linear method.

Another aim of our investigations was the comparison of the 'regular' and our own modified error function. The results indicate that the modified function leads to a considerable increase in performance. This may be because the error function has threshold characteristics that do not overemphasize coincidental price movements outside of the threshold area.

References

[1] Bates, J. M. / Granger, C. W. J. (1969): The Combination of Forecasts, Operational Research Quarterly, Vol. 20, pg. 451 - 468

[2] Baun, S. (1997) Verteilungsprognose für den Deutschen Aktienindex, Deutscher Universitätsverlag, Gabler-Verlag, Wiesbaden

[3] Burgess, A. N. (1996) Modelling Conditional Co-Cointegration in Financial Time Series, Proceedings Unicom Seminar Intelligent Systems for Finance and Commerce, London

[4] Clemen, R. T. (1989): Combining Forecasts: A Review and Annotated Bibliography, International Journal of Forecasting, Vol. 5, pg. 559 - 583

[5] Diebold, F. X. (1988b): Serial Correlation and the Combination of Forecasts, Journal of Business & Economic Statistics, Vol. 6, No. 1, pg. 105 - 111

[6] Dickinson, J. P. (1973): Some Statistical Results in the Combination of Forecasts, Operational Research Quarterly, Vol. 24, pg. 253 - 260

[7] de Groot, C. de (1993): Nonlinear Time Series Analysis with Connectionist Networks, Dissertation ETH Zürich, No. 10038, IPS Report No. 93-03

[8] Refenes, A. N. / Azema-Barac, M. (1992): Neural Network Applications in Financial Asset Management, Department of Computer Science, University College London

[9] SNNS (1995), Stuttgart Neural Network Simulator, User Manual, Version 4.1, Report No. 6/95 University of Stuttgart 1995.

[10] Thomason, Mark R. (1996a), Principal Components Analysis for Neural Network Input Variable Reduction and Financial Forecasting (Part 1), in: NeuroVe$t Journal, Vol. 4, No. 1, pg. 29-32.

[11] Thomason, Mark R. (1996b), Principal Components Analysis for Neural Network Input Variable Reduction and Financial Forecasting (Part 2), in: NeuroVe$t Journal, Vol. 4, No. 2, pg. 30-34.

[12] Utans, J. / Holt, W. T. / Refenes, A. N. (1997), Principal Components Analysis for Modeling Multi-Currency Portfolios, Preliminary Draft, appears in: Weigend, A. S. / Abu-Mostafa, Y. / Refenes, A.-P. N. (eds.), Proceedings of the Fourth International Conference on Neural Networks in the Capital Markets, Singapore: World Scientific, 1997.