A Case Study of Time Series Forecasting with Backpropagation Networks

Nico Schlitter
Group Knowledge Management and Discovery
Otto-von-Guericke-University Magdeburg
Nico.Schlitter@iti.cs.uni-magdeburg.de

This article describes the use of backpropagation networks to predict economic time series. In this study short-term prediction models for the Dow Jones Average Industrial Index and the German Stock Index are generated. The article presents the prediction process and suggests some modifications for the classical training algorithm. Furthermore several methods to select relevant network inputs and their preprocessing are examined. After a short introduction of quality measures the prediction results and their implications are presented.

1 Introduction

Predicting a stock price is more a process rather than a single action. "'Neural network training is an art. Trading based on neural network outputs, or trading strategy is also an art."' [YT01] Therefore it is important to discuss which part of this generation process could be improved. In this article some methods of increasing the prediction quality and to decrease the essential efforts at the same time are suggested. Section 2 shows the main target of the stock price prediction. Section 3 describes each part of the process and presents the improvements. The required quality measures are presented in section 3.5. In the following section 4 the results are visualized and finally a short summary and a future outlook are given.

2 Main target

The time series $\overline{O}$ that we want to predict, is given by daily stock price differences. Consequently the generated models predict the daily interest of $\overline{Z}$:

$$\overline{O}(i) = \frac{\overline{Z}(i) - \overline{Z}(i-1)}{\overline{Z}(i-1)} \quad (1)$$

For some applications the exact stock price is not interesting because in such cases the direction of the price movement is (much) more important. The trend is defined as:

$$\overline{O}(i) = \begin{cases} +1 & \overline{Z}(i-1) < \overline{Z}(i) \\ 0 & \overline{Z}(i-1) = \overline{Z}(i) \\ -1 & \overline{Z}(i-1) > \overline{Z}(i) \end{cases} \tag{2}$$

Independent from the choice of $\overline{O}$ the target of our efforts is to find a function F that maps a set of unknown external variables I_k to the value $\overline{O}(i)$. Probably it is impossible to determine the exact function. Therefore a non explicable rest R remains:

$$\overline{O}(i) = F(I_1, I_2, \cdots, I_n) + R \tag{3}$$

The success of this calculations lies in finding the function F and a fixed number of variables I_k. Fortunately it is possible to find dependencies between $\overline{O}$ and other stock time series by using historical data. This is done in the prediction process.

3 The prediction process

The set of procedures which end in a successful prediction model is denoted as *prediction process*. It consists of data preprocessing, data analysis, data selection, model generation, and model testing. In the following, each phase of this closed loop process is described in detail.

3.1 Data preprocessing

The original market stock prices are given by a set of synchronized time series $\overline{X}(i)$:

$$\overline{X}_k(i) = \left(\overline{X}_k(1), \overline{X}_k(2), \cdots, \overline{X}_k(n) \right) \in R^n \tag{4}$$

In this regard *synchronized* means that the values $\overline{X}_p(l)$ and $\overline{X}_q(l)$ describe the same day in two different time series. In the following, the number of values in a time series $\overline{X}$ is defined as $|\overline{X}_k|$).
The data preprocessing is realized by a set of transformations $f_j : R^n \mapsto R^n$. Each f_j maps an original time series $\overline{X}$ to a transformed time series $\overline{T}$.
A large number of the dependencies which are used for the prediction are time phased. Therefore it is necessary to regard them oder: 'to take them into account'. The function MV moves the time index as shown in (5) and realizes in this way a time lag structure.

$$\begin{aligned} \overline{T} &= MV(\overline{X}, n) \qquad n \in N^+ \\ \overline{T}(i) &= \overline{X}(i-n) \end{aligned} \tag{5}$$

To take account of the economic background some special functions are used. The following important economic indicators (cf. [Wil90]) are used in this study:

- relative strength index (RSI)
- MACD-trigger
- trend oscillator (TO)
- over bought over sold indicator (OBOS)
- volatility (VOL)

Using these kinds of indicators seems to be reasonable, because many traders conform to them and so they influence the stock price. Beside the economic indicators there are some other helpful transformations. They simplify the network training by providing for instance the following useful information:

- difference to the moving average (DifToMA)
- difference to the minimum (DifToMin)

That way the network does not have to realize such simple functions by adapting the internal weights.
For using artificial neural networks it is necessary to normalize the input data. Different methods are examined to normalize the time series. The min-max-normalization scales the data into a fixed interval e.g. $[0, 1]$ or $[-1, 1]$.

$$\overline{T}(i) = \frac{\overline{X}(i) - min(\overline{X})}{max(\overline{X}) - min(\overline{X})} \tag{6}$$

The z-normalization ensures that the data will have an average of zero and a standard deviation of one. Additionally, the outliers do not determine the interval range:

$$\overline{T}(i) = \frac{\overline{X}(i) - aver(\overline{X})}{std(\overline{X})} \tag{7}$$

3.2 Data analysis

The aim of the data analysis is to detect which of the transformed time series $\overline{T}_k$ are related to the objective time series $\overline{O}$. To determine the scale of the dependencies four different test algorithms are used.

Correlation analysis The *correlation analysis* (r) is able to find linear dependencies.

$$r_k = \frac{\sum_i \left(\overline{T}_k(i) - aver(\overline{T}_k)\right) \cdot \left(\overline{O}(i) - aver(\overline{O})\right)}{\sqrt{\sum_i \left(\overline{T}_k(i) - aver(\overline{T}_k)\right)^2 \cdot \sum_i \left(\overline{O}(i) - aver(\overline{O})\right)^2}} \tag{8}$$

For $r > 0$ a linear dependency between the objective time series $\overline{O}$ and another potential influential time series $\overline{T}_k$ exists. For $r < 0$ one can assume an indirect linear dependency. Consequently the value $|r|$ measures direct and indirect dependencies.

Delta-test The *δ-test* (I) in [PP93] is based on the continuity of functions and is able to discover linear and nonlinear relations.

Average mutual information The *average mutual information* (AMI) in [Sha48] is also enabled to discover linear and nonlinear relations. It measures the mutual information between two variables a and b. This mutual information is a measure for the prediction precision of the value b if the value a is known.

Sign parity The *sign parity* (S) describes how often the sign of $\overline{T}_k(i)$ matches the sign of $\overline{O}(i)$:

$$S_k = -0.5 + \frac{1}{|\overline{O}|} \sum_i c \tag{9}$$

$$c = \begin{cases} 1 & sign\left(\overline{O}(i)\right) = sign\left(\overline{T}_k(i)\right) \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Because of the subtraction of 0.5 the value $|S|$ can also be used to measure the direct and indirect dependencies.
Table 1 shows a partial list of results from these tests for circa 20 million pairs of time series. High values point to a strong relation between $\overline{O}$ and $\overline{T}_k$. In this case the value $\overline{T}_k(i)$ seems to be adequate to predict the value $\overline{O}(i+1)$. In order find valid dependencies

Table 1: A part of the described test results

Transf. time series $\overline{T}_k$	**Year**	$\lvert\overline{X}\rvert$	$\lvert r\rvert$	I	AMI	$\lvert S\rvert$
MV(TO(852362,10,15),2)	2004	130	0.054	0.079	7.8	0.007
MV(VOL(852362,5),5)	2003	22	0.014	0.133	3.4	0.035
MV(DifToMA(200455,3),4)	2004	260	0.194	-0.024	0.5	0.190
MV(Rendite$_d$(200455,5),1)	2003	780	0.053	0.035	2.1	0.030
MV(VOL(854717,3),6)	2004	260	0.018	0.094	10.4	0.034
MV(DifToMin(855689,5),6)	2002	520	0.024	0.003	0.6	0.130
MV(OBOS(856584,5),9)	2003	780	0.026	0.054	5.7	0.232
MV(TO(850229,10,15),2)	2004	65	0.054	0.079	7.8	0.007

three different points in time and eight different sizes of time series are used. In that way it is possible to analyze if the prediction quality depends on the size of the training set. Section 4 describes this issue in detail.

3.3 Data selection

The results in table 1 can be used to select the required number of variables in (3). The most relevant time series are examined by two different methods as follows.

High dependency Based on the test results only transformed time series are used with a high influence on the objective time series $\overline{O}$. That way each test method determines twelve influential time series $\overline{I}_k$.

High dependency and less standard deviation In addition to the high influence the time stability seems to be important for a successful prediction. To measure this stability the standard deviation of eight overlapped time periods consisting of 22, 65, 130, 195, 260, 520, 780 and 1560 days are used as shown in figure 1.

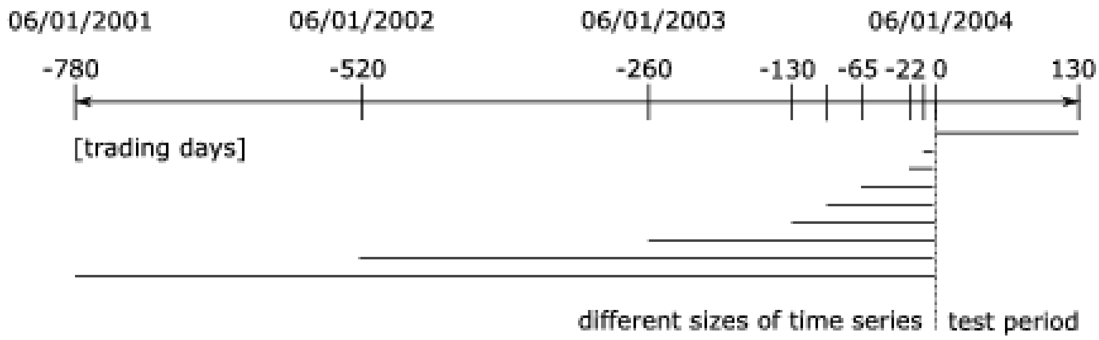

Figure 1: Overlapping time periods for the calculation of the standard deviation

In the following the use of the correlation test will be described in detail. Beside the correlation coefficient the other three methods are used in the same way to select twelve time series each. The standard deviation of the correlation coefficient is defined as:

$$\sigma_k = \sqrt{\frac{1}{8} \cdot \sum\nolimits_{i=1}^{8} (|r_k^i| - \overline{r}_k)^2} \tag{11}$$

Finally the set of time series $\{I_1, \ldots, I_{12}\}$ with a maximal correlation coefficient and a minimal standard deviation is used:

$$\sum\nolimits_{k=1}^{12} \sigma_k \longrightarrow minimize \tag{12}$$

$$\sum\nolimits_{k=1}^{12} r_k \longrightarrow maximize \tag{13}$$

3.4 Model generation

After the selection of the input time series it is possible to create a prediction model. For this end we use modified backpropagation networks. The aim of the network training is to approximate the function F in (3) by minimization of the *mean squared error* (MSE) between the real values $\overline{O}(i)$ and the predicted values $\overline{P}(i)$.

$$MSE = \frac{1}{2 \cdot |\overline{O}|} \sum_{i}^{|\overline{O}|} \left(\overline{O}(i) - \overline{P}(i)\right)^2 \tag{14}$$

Usually the standard algorithm in [RHW86] is used to adapt the weights in the backpropagation network:

$$W_{ij}^L(t+1) = W_{ij}^L(t) + \Delta W_{ij}^L(t) \tag{15}$$

$$\Delta W_{ij}^L(t) = -\gamma(t) \cdot N_i^L(t) \cdot \delta_j^{L+1}(t) \tag{16}$$

where

$W_{ij}^L(t)$ is the weight between neuron i in layer L and neuron j in layer $L+1$ at time t
$\Delta W_{ij}^L(t)$ is the change of the weight $W_{ij}^L(t)$ at time t
$\gamma(t)$ is the learning rate at time t
$N_i^L(t)$ is the input of the neuron i in layer L at time t
$\delta_j^L(t)$ is the error signal of neuron j in layer L at time t

In this study about three million backpropagation networks were generated to predict four several time series:

- the daily interest of Dow Jones Index (DJ)
- the daily interest of German Stock Index (DAX)
- the daily trend of Dow Jones Index
- the daily trend of German Stock Index

Eight different time series sizes (cf. figure 1) and three different points in time[1] are used to examine the relation between prediction quality and the size of the training set. Furthermore the network parameters e.g. the numbers of the network layers or the neurons, were controlled by standard evolution strategies [Rec94]. The necessary calculation power was provided by the homogeneous Linux cluster *CLiC* consisting of 512 nodes. The network training is supplemented by the following three modifications to improve the classic training algorithm.

[1] 06/01/2002, 06/01/2003 and 06/01/2004

Backpropagation with momentum term Rumelhart et al. [RHW86] suggest to add some part of the last weight change to the actual weight change. Therefore (16) is modified:

$$\Delta W_{ij}^{L}(t) = \gamma(t) \cdot N_i^L(t) \cdot \delta(t)_j^{L+1} \overbrace{\cdot(\alpha - 1) + \alpha \cdot \Delta W_{ij}^{L}(t-1)}^{\text{weighted momentum term}} \tag{17}$$

Delta-Bar-Delta This modification is based on different learning rates γ_{ij}^{S} for each weight. Additionally a learning rate control exists that changes the learning rate depending on the neurons' error signal. [Jac88]

$$\Delta W_{ij}^{L}(t) = -\gamma_{ij}^{L}(t) \cdot N_i^L(t) \cdot \delta_j^{L+1}(t) \tag{18}$$

$$\gamma_{ij}^{L}(t+1) = \gamma_{ij}^{L}(t) + \Delta\gamma_{ij}^{L}(t) \tag{19}$$

$$\Delta\gamma_{ij}^{L}(t) = \begin{cases} \kappa & \text{for } \overline{\delta}_j^{L+1}(t-1) \cdot \delta_j^{L+1}(t) > 0 \\ -\phi \cdot \gamma_{ij}^{L}(t) & \text{for } \overline{\delta}_j^{L+1}(t-1) \cdot \delta_j^{L+1}(t) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

$$\overline{\delta}_j^{L}(t) = (1-\theta) \cdot \delta_j^L + \theta \cdot \overline{\delta}_j^{L}(t-1) \tag{21}$$

where

$\gamma_{ij}^{L}(t)$ is the learning rate at time t
$\Delta\gamma_{ij}^{L}(t)$ is the change of the learning rate at time t
κ is the linear increment of the learning rate
ϕ is the proportion of decrease of the current learning rate
$\overline{\delta}_j^{L}(t)$ is the exponential average of the current and past error signals
θ is the base of this exponential average and t the exponent

Equation (20) enables a linear increment and exponential decrease of the learning rate. Thus a fast learning rate adaption is guaranteed. In this study the learning constants (e.g. κ, ϕ, γ) are controlled by Rechenbergs' evolution strategies.

Delta-Bar-Delta with exponential growth Furthermore Jacobs' algorithm is adapted to allow an exponential increment of the learning rate. Therefore (20) changes to:

$$\Delta\gamma_{ij}^{L}(t) = \begin{cases} +\kappa \cdot \gamma_{ij}^{L}(t) & \text{for } \overline{\delta}_j^{L+1}(t-1) \cdot \delta_j^{L+1}(t) > 0 \\ -\phi \cdot \gamma_{ij}^{L}(t) & \text{for } \overline{\delta}_j^{L+1}(t-1) \cdot \delta_j^{L+1}(t) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

where

κ is the proportion of increase of the current learning rate

This modification allows an exponential growth of the learning rate. So the learning rate can be quickly adapted to the current error surface.
Independent from the training algorithm a separate test set is used to determine the ideal training stop. This test set is not involved in the training process of the networks and contains 50 percent of the available historical data. Using a time series size of 520 days the training and test set consist of 260 vectors with the dimension of 48.
To prevent overfitting it is necessary to stop the network training phase. In this purpose the MSE on the test set is used. During the training process the network with the smallest MSE is saved. After a fixed number of training cycles (e.g. 1000) or a significant rising of the MSE (e.g. 20% beyond the smallest MSE) the saved network with the smallest error is recovered.

3.5 Model testing

After model generating some ways to compare oder: "'of how to compare"' the different models are required. Contrary to the network training the testing orients toward the practical exertion: The error rate, Theil's inequality coefficient, and the profit gain are used to test the models on a separate validation set. This set consists of 130 trading days (half a year) and is not used in the training period. During the validation phase the prediction models are not adapted.

Error rate In the validation period measurements were done to check how often the signs of the real value $\overline{O}^V(i)$ differs from the predicted value $\overline{P}^V(i)$. The proportion of false predictions is termed as *error rate* (ER).

$$ER = \frac{1}{130} \cdot \sum_{i=1}^{130} \begin{cases} 1 & \text{for } sign\left(\overline{O}^V(i)\right) \neq sign\left(\overline{P}^V(i)\right) \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

In an acceptable model the number of false trend predictions should be remarkably less than 50 percent. The aim is to minimize this error rate.

Theil's inequality coefficient *Theil's inequality coefficient* [The67] allows a comparison between the current prediction model and the *naive prediction.* This special benchmark method predicts the trend or change of the previous day for the next day.

$$U = \frac{\sqrt{\frac{1}{130} \cdot \sum_{i=1}^{130} \left(\overline{P}^V(i) - \overline{O}^V(i)\right)^2}}{\sqrt{\frac{1}{130} \cdot \sum_{i=1}^{130} \left(\overline{O}^V(i)\right)^2}} \tag{24}$$

The interpretation is simple: The smaller Theil's inequality coefficient the better the prediction quality is. A perfect model shows an inequality coefficient of $U = 0$. When $U > 1$ the naive prediction is better compared to the proposed prediction model.

Profit gain The stock trading is simulated to estimate the model's practical implications. The reached profit is compared to the profit of the by-and-hold strategy. This simple trading strategy measures the profit which results from a stock buying on the first and a stock selling on the last day of the validation period. A model is considered as useful if the simulated profit exceeds the buy-and-hold profit. The difference between both is called *profit gain* (PG):

$$\mathrm{PG} = \underbrace{\left(\frac{\text{final capital}}{\text{start capital}}\right)^{\left(\frac{260}{130}\right)}}_{\text{simulated profit p.a.}} - \underbrace{\left(\sum_{i=1}^{130} \overline{O}^V(i)\right)^{\left(\frac{260}{130}\right)}}_{\text{buy-and-hold profit p.a.}} \tag{25}$$

To allow an easy comparison between different prediction models the profit gain in (25) is calculated on an annual basis (260 trading days) by including compound interest effects - consequently the reached interest is measured per annum.
The profit gain is maximized by using a suitable trading strategy and a good prediction model. The start capital amounts 10,000 Euro and during the simulation we do not take account of any trading fees. Depending on a daily prediction this proposed trading strategy provides buy- or sell signals. In case of positive predictions the stock is bought; otherwise it is sold.

4 Results

The prediction results are presented in Table 2. Beside the training and test period one can recognize the buy-and-hold profit (B&H), the annually profit gain (PG) and the reached annually total profit. For each objective time series and each prediction period the results of the best network are shown in a row. The selection of the best network is based on the precision of the test set. The displayed results come from the validation set. It is almost always possible to reach a positive profit gain. That means, the generated

Table 2: Complete prediction results

(a) DAX daily trend prediction

prediction period	optimal training period	B&H	PG	Profit
06/01/ - 11/29/02	65: 27.02.02 - 06/01/02	-51,8 %	73,4	21,6 %
06/01/ - 11/28/03	65: 26.02.03 - 06/01/03	50,6 %	1,7	52,3 %
06/01/ - 11/30/04	65: 01.03.04 - 06/01/04	12,7 %	13,7	26,4 %

(b) DAX daily interest prediction

prediction period	optimal training period	B&H	PG	Profit
06/01/ - 11/29/02	22: 05/02/02 - 06/01/02	-51,8 %	26,6	-25,2 %
06/01/ - 11/28/03	65: 02/26/03 - 06/01/03	50,6 %	3,7	54,3 %
06/01/ - 11/30/04	22: 05/03/04 - 06/01/04	12,7 %	-24,1	-11,4 %

(c) Dow Jones daily trend prediction

prediction period	optimal training period	B&H	PG	Profit
06/01/ - 11/29/02	65: 02/28/02 - 06/01/02	-18,9 %	16,2	-2,7 %
06/01/ - 11/28/03	65: 02/27/03 - 06/01/03	22,9 %	2,6	25,5 %
06/01/ - 11/30/04	65: 03/01/04 - 06/01/04	5,5 %	4,6	10,1 %

(d) Dow Jones daily interest prediction

prediction period	optimal training period	B&H	PG	Profit
06/01/ - 11/29/02	22: 05/01/02 - 06/01/02	-18,9 %	1,8	-17,1 %
06/01/ - 11/28/03	65: 02/27/03 - 06/01/03	22,9 %	2,6	25,5 %
06/01/ - 11/30/04	520: 05/10/02 - 06/01/04	5,5 %	14,3	19,8 %

models are able to outperform the general market. The prediction of the DAX in the year 2004 is the only exception.

Some of the networks and the trading strategies obtain a positive interest even though the market moves downward. But this is only possible by ignoring the trading fees. Otherwise the numerous trades and the resultant fees would depreciate the profit.

Probably, this problem can be solved by using a better trading strategy which regards the accruing fees of each potential sale. This way it is possible to reduce the number of sales and consequently the amount of money spent on trading fees. Furthermore the change of the prediction horizon (e.g. weekly instead of daily predictions) is another option to decrease the number of trades, because the number of alternated trading signals would be reduced.

4.1 Size of training set

Also the small size of the training set in table 2 attracts to attention. Figure 2 shows the quality of the daily interest prediction of the DAX depending on the size of the training set. The best results are achieved with a relatively small training set: By using 65 days Theil's inequality coefficient and the error rate are minimal and the profit gain reaches a maximum.

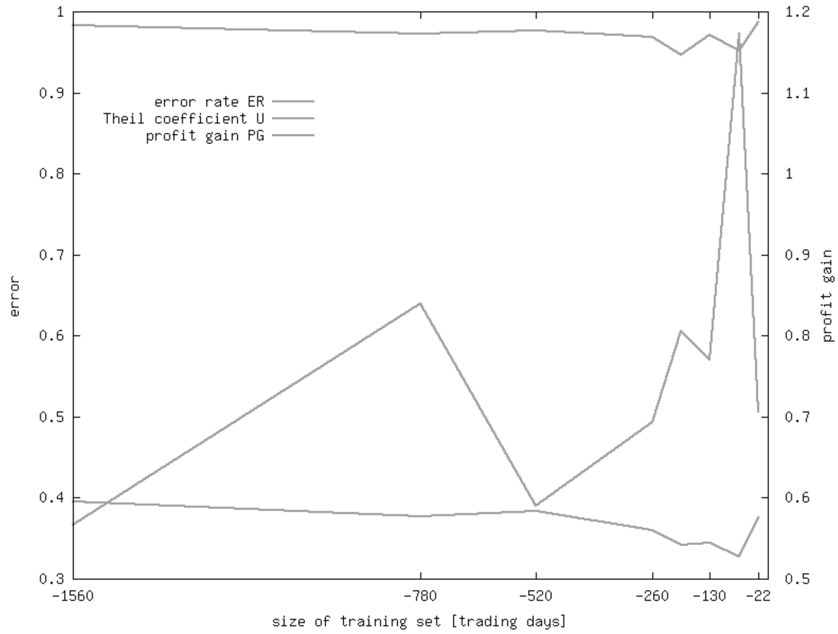

Figure 2: Trend prediction quality DAX 06/01/2003

Against the expectation the optimal training period is relatively short. With 65 days it seems to be capacious enough to include necessary information but small enough to exclude timeworn rules and information. With regards to the optimal training period nearly all experiments show similar results (cf. table 2).

4.2 Training algorithm

The training algorithm has only a small influence on the prediction quality in this study. But it certainly influences the convergence of the network training. The quotient of complexity and prediction precision is used to show this fact. The complexity is measured by the training cycles (TC), the precision by the profit gain (PG), the reciprocal error rate ($\frac{1}{ER}$), or the reciprocal Theil inequality coefficient ($\frac{1}{U}$):

$$PG_{rel} = \frac{TC}{PG}, \quad ER_{rel} = TC \cdot ER, \quad U_{rel} = TC \cdot U \qquad (26)$$

The smaller the complexity precision proportion the better model's precision and complexity are. Figure 3 shows the advantage of *Delta-Bar-Delta with exponential growth.* In the relevant range between 65 and 260 days it has almost always the best complexity precision proportion.

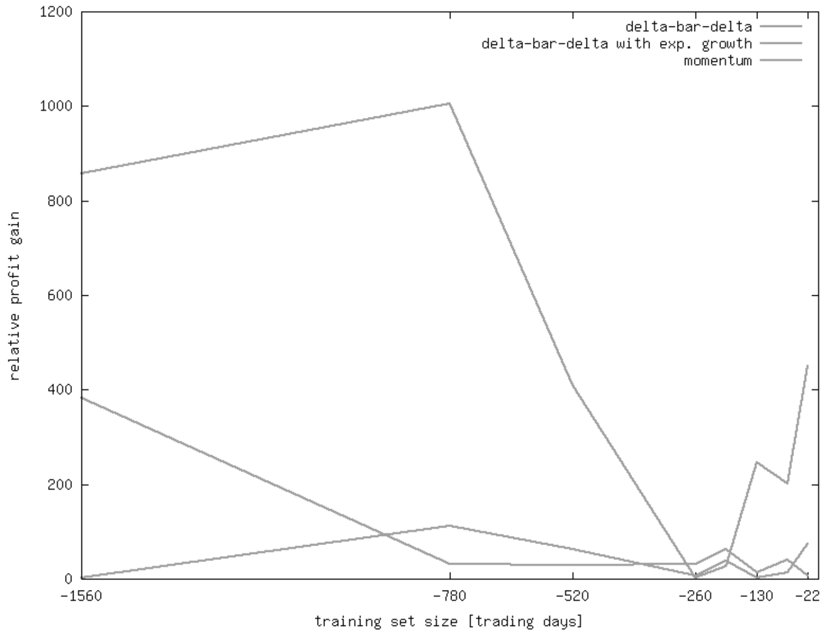

Figure 3: Prediction quality DAX trend prediction

4.3 Normalization

Similarly to the training algorithm the kind of normalization is not important for the prediction quality. However like the training algorithm it influences the time of convergence. Figure 4 shows the relative profit gain (PG_{rel}) for the three analyzed normalization methods.

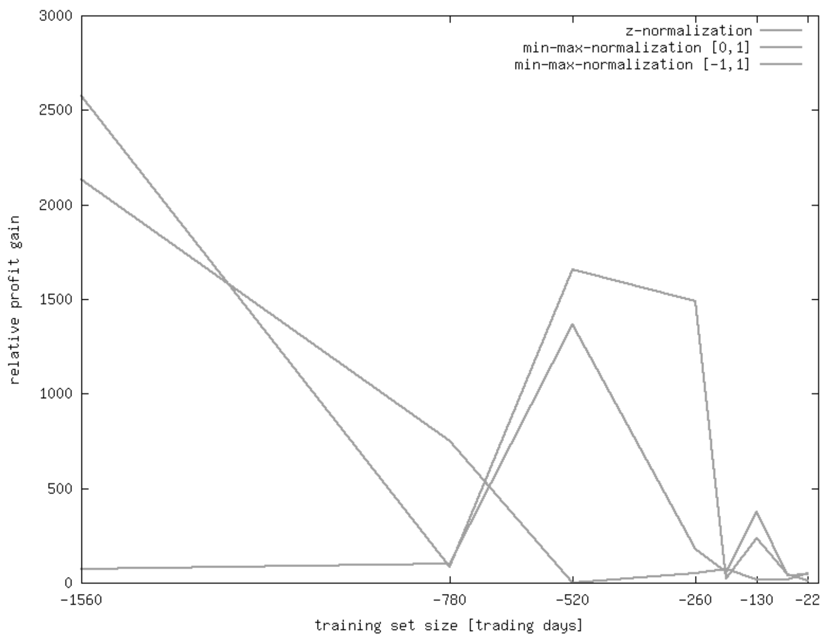

Figure 4: Prediction quality DJ daily interest prediction

The advantage of the z-normalization is clearly recognizable, because in the important range between 22 and 260 days it has a better relative profit gain than both other methods. By using the z-normalization it is possible to reduce the necessary calculation power and enhance the prediction precision at the same time.

4.4 Data selection

Finally the influence of data selection methods is analyzed. The differences between the two examined algorithms are shown in figure 5.

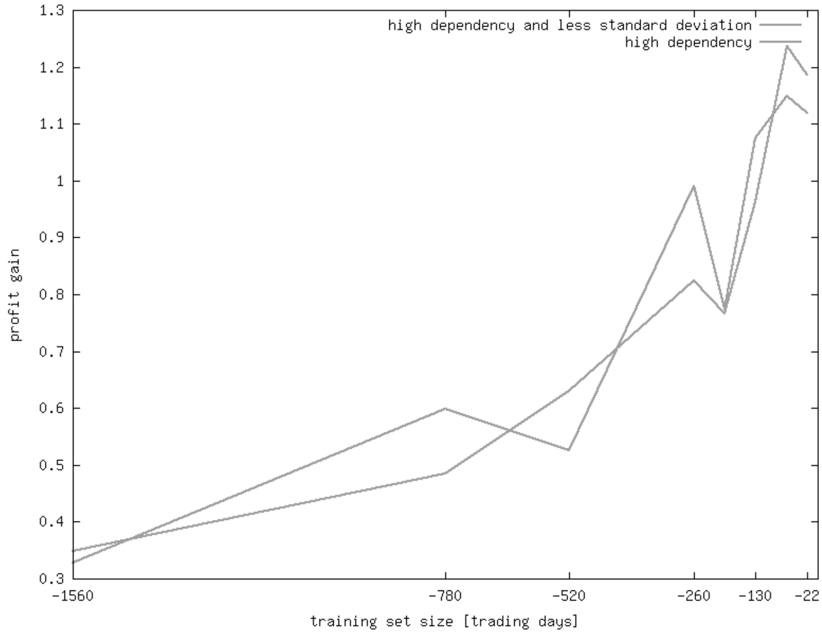

Figure 5: Prediction quality DJ daily interest prediction

It is not possible to recognize a significant advantage for one of both. The costlier method of regarding time stability is not worthwhile. On that score it is necessary to develop another more suitable data selection method.

5 Conclusion

The prediction models reach nearly always a positive profit gain. That means it is succeeded to generate models that outperform the general market.
Furthermore the results show some interesting implications. So the correct size of the training set is more important for the prediction quality than expected. This fact has not been taken into account in other papers. Based on the results a training period between 65 and 130 trading days is suggested to reach a better prediction quality. Furthermore the normalization and the choice of the training algorithm influence the quality indirectly because of the training convergence. The Use of the Delta-Bar-Delta algorithm with exponential growth and the z-normalization reduce the time for training. The saved time can be spend to create more prediction models by using optimization methods, as e.g. evolution strategies, to adjust the number of layers, neurons, or activation functions.

6 Further Research

Due to complexity of these calculations it is almost impossible to find the ideal size of the training set. Therefore it might be useful weighted training set. Older stock prices should be less involved than more actual data. To use such a method it is necessary to adapt the training algorithm and the quality measures. The future research should regard this supposition.
Data selection and preprocessing are the most important parts of the prediction process.

Based on the *efficient market hypothesis* [Fam70] it is necessary to include other information in addition to raw stock price time series. The analysis of business news could contribute in a promising way. Kroha's research on this field [KBY05, KBYK06] should be combined with traditional forecast methods to improve the prediction quality.

References

[Fam70] E.F. Fama. Efficient Capital Markets. *The Journal of Finance*, XLVI:1575–1617, 1970.

[Jac88] Robert A. Jacobs. Increased Rates of Convergence Through Lerning Rate Adaption. *Neural Networks*, 1:295–307, 1988.

[KBY05] Petr Kroha and R. Baeza-Yates. A case study: News classification based on dexa'2005, workshop on theory and applications of knowledge management takma'2005. In *Proceedings of 16th International Conference*, pages 428–432, Copenhagen, August 2005. IEEE Computer Society.

[KBYK06] Petr Kroha, R. Baeza-Yates, and Bjrn Krellner. Text mining of business news for forecasting. In *Proceedings of Seventh International Workshop on Theory and Applications of Knowledge Managment*, September 2006. to be published.

[PP93] Hong Pi and Carsten Peterson. Finding the Embedding Dimension and Variable Dependencies in Time Series. *Neural Computation*, 6:509–520, 3 1993.

[Rec94] Ingo Rechenberg. *Evolutionsstrategie '94*. Frommann Holzboog, 1994.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Interior Representation by Error Propagation. *Parallel Distributed Processing*, 1:318–362, 1986.

[Sha48] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, Juli, Oktober 1948.

[The67] Henri Theil. *Economics and information theory.* Rand-McNally, Chicago, 1967.

[Wil90] J. Wells Wilder. *New Concepts in Technical Trading Systems.* Dearborn Publishing, Inc., 1990.

[YT01] JingTao Yao and Chew Lim Tan. Guidelines for financial forecasting with neural networks. In *Proceedings of International Conference on Neural Information Processing*, pages 757–761, Shanghai, China, November 2001.