Neurocomputing 2 (1990) 17 – 27
Elsevier

Stock price prediction using neural networks: A project report

E. Schöneburg

Expert Informatik GmbH, Roenneberg Str. 5A, D-1000 Berlin (West) 41, FRG

Abstract. We analyzed the possibility of predicting stock prices on a short-term, day-to-day basis with the help of neural networks by studying three important German stocks chosen at random (BASF, COMMERZBANK, MERCEDES). We examined the use of PERCEPTRON, ADALINE, MADALINE and BACK-PROPAGATION networks.

The results were encouraging. Within short prediction time spans (10 days), we achieved a very hight degree of accuracy of up to 90%. With a BACK-PROPAGATION network we carried out an absolute-value prediction. The network was thereby able to recognize on its own an obvious heuristic and showed a behaviour similar to the exponential smoothing algorithm.

The results we achieved led us to expect that neural networks could considerably improve the prognosis of stock prices (and more generally, the prognosis of semi-chaotic time series) in the future.

Nevertheless considerable improvements are needed in the theory of neural networks, as practicable methods to support the design of neural networks for specific applications are not available yet.

1. Introduction

In the course of the past two years, the international financial world has suffered two serious stock exchange collapses. Surprisingly however, these crashes did not have the devastating effects on the international economy which could have been expected from a historical point of view. A certain separation of the stock exchange from the economic situation seems to have emerged, the stock exchange no longer being the direct barometer of economic development in some areas.

This partial disappearance of the narrow cause and time relationship between the stock exchange and the actual economic fluctuations makes the prediction of stock prices even more difficult than it had been until now. In addition to this, stock trading has become able and forced to react to political and other relevant events faster and faster due to the ever increasing use of data processing technology and to the improvement of communication systems.

Banks, financial institutions, large-scale investors and stock brokers therefore now more than ever face the problem of having to buy and resell at a profit a maximum number of stocks within the shortest possible time. In this regard, a time span of only a few hours between buying and selling stocks is not unusual.

Under these conditions, what are the possibilities for predicting stock prices with the help of data processing and computer programs?

Conventional (i.e., nonadaptive) prediction programs are working best in situations where the real course of stock prices can be smoothened for the prediction by means of statistical procedures without loss of prediction relevance. In the case of medium-term investment, this is realistic and useful, for instance, for private investors who usually keep their stocks for a number of weeks or months. Large-scale investors and brokers can use such procedures on a limited scale only. They make their profits, for the most part, not from a maximum price increase of fewer stocks over a longer period of time, as is the case for private in-

0925-2312/90/$03.50

vestors, but rather as a rule from a relatively smal' price increase of a great number of stocks in an extremely short time.

Under these conditions, conventional prediction methods can only be used with a great amount of effort. The price fluctuations are for the most part too minimal to be successfully recognized and predicted with the statistical procedures which smoothened the course of stock prices for prediction purposes. Statistical smoothing usually disregards just that factor which is most important for predictions in short periods of time: small upward or downward price fluctuations.

A further and very important disadvantage of conventional nonadaptive program techniques is that for the analysis of the data, all relevant influencing factors must already be known in advance. Conventional programs must "know" all parameters and influencing factors, i.e., they must contain the parameters along with their assessments and weightedness a priori in their program code. The designer of this type of program must therefore know and determine exactly which influencing factors he thinks are important and to what extent the wants to consider them.

In the case of the economic development and intertwinement of today's world, such as we are witnessing with the upheavals in Eastern Europe, it is practically impossible to recognize all the relevant influencing factors and explicitly formulate them. The fundamental analysis is basically restricted by the incredible abundance and complexity of information to be considered. No one can determine today what relevant factors will affect the world economy in the next few years. Too many factors are unsure.

As a result, it is highly improbable that a conventional program should be able to correctly recognize and predict, even with only approximate accuracy, the trends implicitly contained in the economic data of the future. This would only be possible if the program were constantly adapted to the current situation, which is as a rule very costly and difficult, and which would probably take more time than the life-span of the trends themselves.

(We do not criticize adaptive algorithms for statistical prediction since most of them are using ADALINE-like neural network algorithms introduced by Widrow and Hoff; see [4], [1, p. 199 ff.] and [2].)

1.1. The alternative: Neural networks

In the current situation, neural networks offer a genuine alternative to conventional nonadaptive prediction techniques. They have two great advantages over conventional methods: (1) they can recognize "on their own" implicit dependencies and relationships in data, and (2) they can "learn" to adapt their behavior (their prediction) quickly and without complication to changed conditions. These important capabilities of neural networks solve, at least in principle, some of the problems mentioned above.

The following results show that it is possible to make surprisingly good predictions, even with relatively simple and almost "old-fashioned" types of neural networks. These results are part of a comprehensive study, which we carried out in preparation for a contract to develop neural networks for stock price prediction for a major German bank (see [3]).

2. The results

For the purpose of the following study, we have assumed that stock prices do not represent completely chaotic time series. This is assumed from time to time. We have been working on the basis of the axiom that stock prices show, at least in part, certain tendencies and trends which may be very difficult to recognize but nevertheless exist. We have also presupposed that price predictions—even in a very restricted form—are possible and useful using only methods of technical stock analysis. We have until now not considered fundamental analytical methods.

We have analyzed several classical network types in terms of their suitability for stock price prediction. In particular, we have examined ADALINE, MADALINE, PERCEPTRON and BACK-PROPAGATION networks. Our aim was mainly a short-term rise-fall prediction for the next day (in the case of the ADALINE, MADALINE and PERCEPTRON networks) and an absolute-value prediction (BACK-PROPAGATION network) for the next day.

As data for training the networks, we used the prices of three randomly chosen major German shares (BASF, COMMERZBANK and MERCEDES) over a period of 42 days (from 9 February 1989 until 14 April 1989). After the 42 training days, predictions were made for up to a maximum of 56 days.

The input vector had a breadth of 40 elements for most of the networks (the BACK-PROPAGATION network had 10 input elements), i.e., for each learning step 40 (10) data items were fed in parallel into the networks' input layer. These are the input information items we used:

- *K* = the current day stock price,
- *VV* = the absolute variation of the price in relation to the previous day,
- *RV* = the direction of this variation (rise, fall),
- *RG* = the direction of the variation from two days previously,
- *G* = major (>1% of stock price) variations in relation to the previous day,
- the prices of the last 10 days in the case of BACK-PROPAGATION.

In our studies, the element *K* has for the most part not been directly necessary for prediction, as it is only the relative variation in prices, and not the absolute price value, which is of significance for price predicition.

Figure 1 shows some of the linearizations of input data we examined and their distribution on the input vector. The predictions were strongly dependent on the linearization chosen, as shown by the result charts below.

2.1. *The ADALINE network*

Training was over a period of 42 days, predicting over 19. A comparison of the dependencies between various linearizations and the learning cycles (between 2000 and 3000 presentations of data to be learned were "shown" to the network) is presented in Table 1.

Our best result, with the BASF stock (linearization (c), 2500 learning steps), was an accuracy rate of 79% for a rise-fall prediction for the following day. The COMMERZBANK stock, with linearization (d) and also 2500 learning steps, was predicted

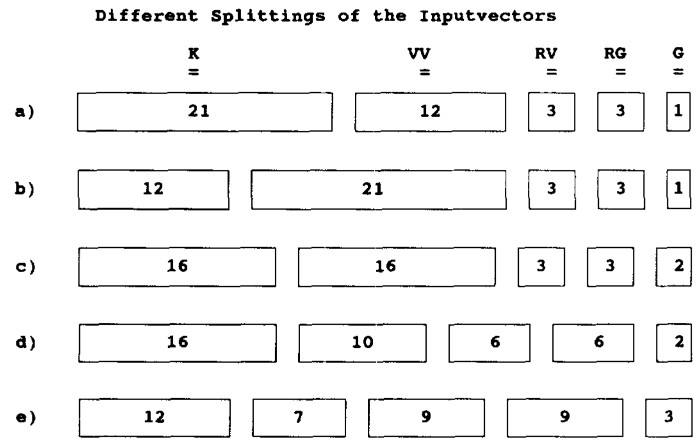

Fig. 1. Different splitting of the input vectors.

with a maximum accuracy of 74%. In contrast, the results for the MERCEDES stock were rather modest. The maximum accuracy achieved here with linearizations (a) and (d) and 2000 learning steps, and with linearization (a) and 3000 learning steps, was only 58%.

We found the dependency of the networks' prediction capacity on the planned prediction period and on the distance in time from the actual date to be of particular interest. The results are shown in Table 2, Fig. 2, Table 3, Figs. 3 and 4.

Figures 2–4 are to be interpreted in the follow-

Table 1
Prediction accuracy (%) dependent on linearization and number of learning cycles with the ADALINE network

	BASF			COMMERZBANK			MERCEDES		
	2000	2500	3000	2000	2500	3000	2000	2500	3000
Linearization (a)	58	58	58	47	68	68	58	53	58
Linearization (b)	47	68	63	63	58	58	42	53	53
Linearization (c)	63	79	74	68	63	63	53	53	53
Linearization (d)	68	68	68	63	74	68	58	53	53
Linearization (c)	47	63	63	58	63	63	53	53	53

Table 2
Prediction accuracy (%) dependent on linearization and prediction time for BASF stock with the ADALINE network

	Prediction period in days						
	10	20	30	40	50	58	
(a)	70,0	55.0	46,7	40,0	42,0	41,4	49,2
(b)	70,0	65,0	56,7	50,0	50,0	48,3	56,7
(c)	80,0	70,0	60,0	55,0	60,0	56,9	63,7
(d)	80,0	75,0	70,0	57,5	58,0	55,2	66,0
(e)	70,0	60,0	50,0	40,0	38,0	37,9	49,3
	74,0	65,0	56,7	48,5	49,6	47,9	57,0

Table 3
Prediction accuracy (%) dependent on linearization and prediction time for COMMERZBANK stock with the ADALINE network

	Prediction period in days						
	10	20	30	40	50	58	
(a)	60,0	65,0	63,3	52,5	54,0	51,7	57,8
(b)	80,0	65,0	63,3	60,0	62,0	58,6	64,8
(c)	70,0	65,0	56,7	47,5	54,0	53,4	57,8
(d)	90,0	75,0	60,0	52,5	58,0	56,9	65,4
(e)	70,0	65,0	60,0	52,5	56,0	53,4	59,5
	74,0	67,0	60,7	53,0	56,8	54,8	61,1

ing way: the thin line represents the training period, the bold line the prediction period and the dotted line the average prediction succes in % of the several 10 day periods. The thin line should be placed in front of the bold line. It is included to allow a comparison between the training data and the recall data. The straight line represents the overall trend in prediction accuracy.)

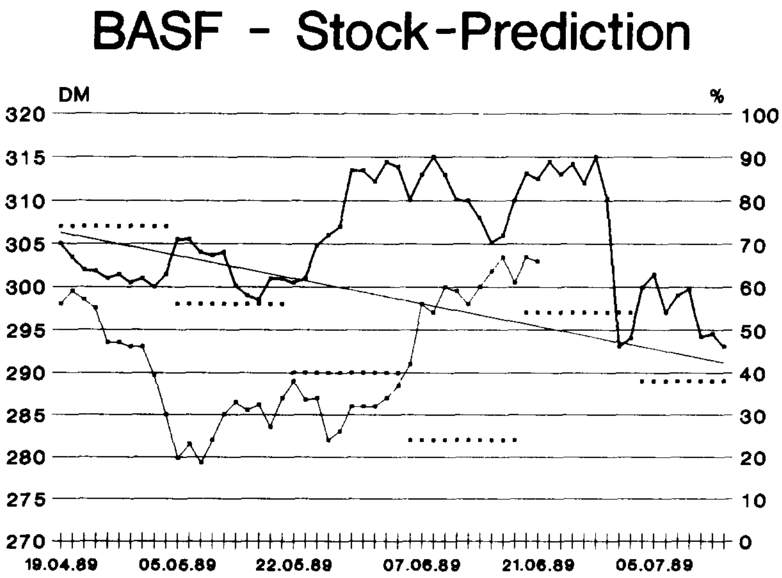

Fig. 2. BASF stock prediction with the ADALINE network: Training (thin line), prediction (bold line), average prediction (dotted line), and overall trend (straight line).

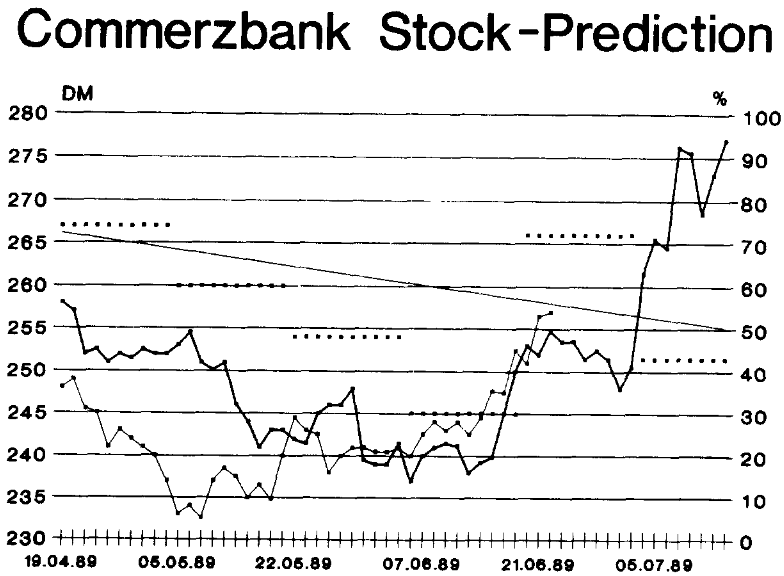

Fig. 3. COMMERZBANK stock prediction with the ADALINE network: Training (thin line), prediction (bold line), average prediction (dotted line), and overall trend (straight line).

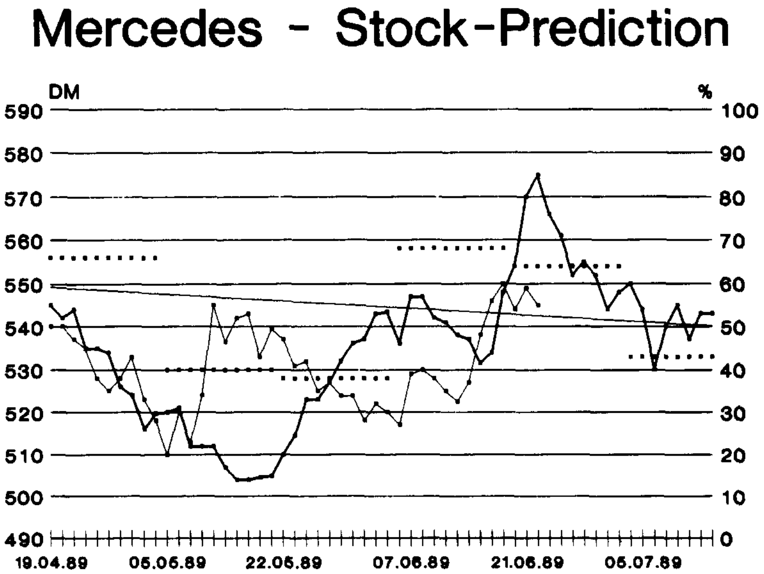

Fig. 4. MERCEDES stock prediction with the ADALINE network: Training (thin line), prediction (bold line), average prediction (dotted line, and overall trend (straight line).

It must be noted that, although the success rate drops along an almost linear pattern as the time distance to the actual date increases (which was to be expected), it was nevertheless possible to make very good predictions for some ten-day periods even though the time distance to the actual date was very large. This is probably because the network can make good predictions for periods which have occurred in a similar form in the past and for which the future then was similar to the future to be predicted. In these cases the network was able to recognize similar historical situations relatively well.

2.2. The MADALINE networks

A MADALINE network can be seen as a combination of ADALINE networks, in which there are ADALINE neurons in the intermediate layer of the network which generate a result by using the majority function in the output layer. For a MADALINE network, therefore, it is necessary to determine how many ADALINE processors are to be used in the intermediate layer. We tested MADALINE networks with 3 to 21 ADALINE elements (due to combinatorial considerations, we only used odd numbers of ADALINE PE's). At first, we worked with a fixed linearization (a) and put the network through 4000 learning steps.

The results of these tests (based on a rise-fall prediction for 19 days) can be seen in Table 4. For all three stocks, 17 ADALINE processors in the intermediate layer were a local optimum (as we guess). The relationship between the number of ADALINE processors needed, the basic linearization and the prediction time are presented in a chart (Table 5), using the COMMERZBANK stock as an example. We have, however, not yet analyzed these interrelationships mathematically, as they are quite complex.

2.3. The PERCEPTRON network

The PERCEPTRON network showed the worst prediction capacity of all, a fact which we had expected in view of the well-known limitations of this type of network. The best result was 68% accuracy, but it was surprisingly enough achieved very often for the MERCEDES stock, which was not very successfully predicted by the other network types.

Determining optimal learning coefficients was difficult. The final tests were carried out with learning coefficients $c1 = 0.01$ and $c2 = 0.05$, as they

Table 4
Prediction accuracy (%) with the MADALINE network

ADALINE PE's	BASF	COMMERZBANK	MERCEDES	
3	47	68	58	58
5	58	58	63	60
7	53	63	58	58
9	58	63	47	56
11	58	74	63	65
13	58	68	58	62
15	68	58	53	60
17	68	74	63	69
19	58	74	47	60
21	53	63	47	54
	58	66	56	60

Table 5
Relationship between number of ADALINE processors, basic linearization and prediction time for COMMERZBANK stock with the MADALINE network

COMMERZ-BANK		Prediction period in days						
		10	20	30	40	50	58	
(a)	11	80	70	63	60	58	57	65
(a)	17	70	75	63	63	56	57	64
(a)	19	80	70	60	63	56	57	64
(d)	7	70	65	60	55	56	57	61
(d)	11	80	65	53	50	56	57	60
		76	69	60	58	56	57	63

Table 6
Prediction accuracy (%) with the PERCEPTRON network

$c1 = 0.01$, $c2 = 0.05$	BASF			COMMERZBANK			MERCEDES		
	2500	3000	3500	2500	3000	3500	2500	3000	3500
Linearization (a)	58	58	58	47	47	47	63	63	63
Linearization (b)	42	42	42	53	58	63	58	42	47
Linearization (c)	58	58	58	47	47	47	68	63	63
Linearization (d)	47	47	47	58	58	58	68	68	68
Linearization (e)	68	52	52	42	42	42	63	68	68

achieved the best results. Linearizations (a) – (e) were used, and stock prices for 43 days learned 2500 to 3500 times. Afterwards, 19 predictions were assessed; the results are presented in Table 6.

2.4. The BACK-PROPAGATION network

To train the BACK-PROPAGATION network, the input data had to be transformed (scaled) to the real interval [0.1, 0.9] first. Finding the best topology for the BACK-PROPAGATION network required much work. The best results produced the topology with the corresponding transfer functions as shown in Fig. 5.

As learning rules, we used the cum-delta rule as well as the delta rule; however, only the connections to the output layer learned with the cum-delta rule. This led to slightly better results than with the exclusive use of one of the two rules. The summation function was the simple summation and the output function was direct. We used the following figures for the learning coefficients: $c1$ (learning rate) = 0.6 and $c2$ (momentum) = 0.9.

A partial result of our test series for 60 days of training and 40 recall figures (predictions) can be seen in Figs. 6 and 7. It can be seen that the predicted figures exhibit a time shift of almost one day.

There are two reasons for this time shift. Firstly, the network was trained wiht a "window" of 10 days being placed on the learning data to predict the eleventh day. The "window" was then shifted one day to the right and the new set of data was then learned.

Secondly, and interestingly, the network itself has apparently discovered the following heuristic: >> take for the next-day prediction the stock price of the last day and modify it only slightly <<. This is evident in the fact that the prediction varies only slightly from the previous day's stock price, and creates the impression of a time shift of the prediction in relation to the actual stock price.

This apparent time shift does also occur when using exponential smoothing of first or second order for prediction purposes instead of neural networks. The network therefore discovered the importance of the last day's stock price for the prediction which is the essence of the exponential smooting algorithm. It used the 10 day data window only to a minor extent as a support for the prediction.

The simple heuristic detected by the network often leads to quite good results, e.g. in weather forecasting, since as a rule a weather change is less probable than a steady weather pattern. This is also true for stock prices. The critical point with

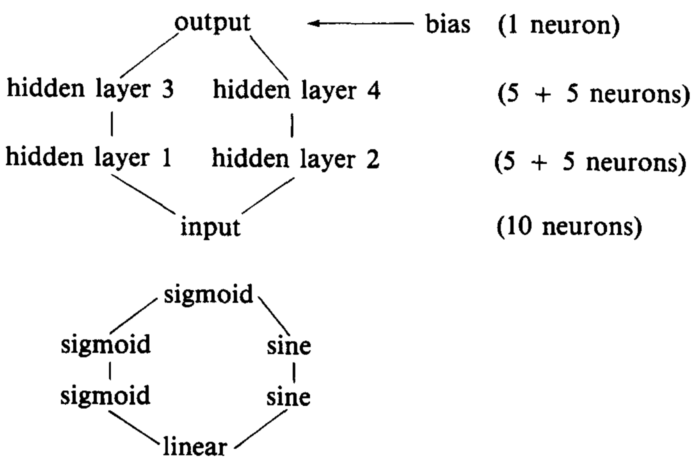

Fig. 5. Topology and transfer functions of the BACK-PROPAGATION network.

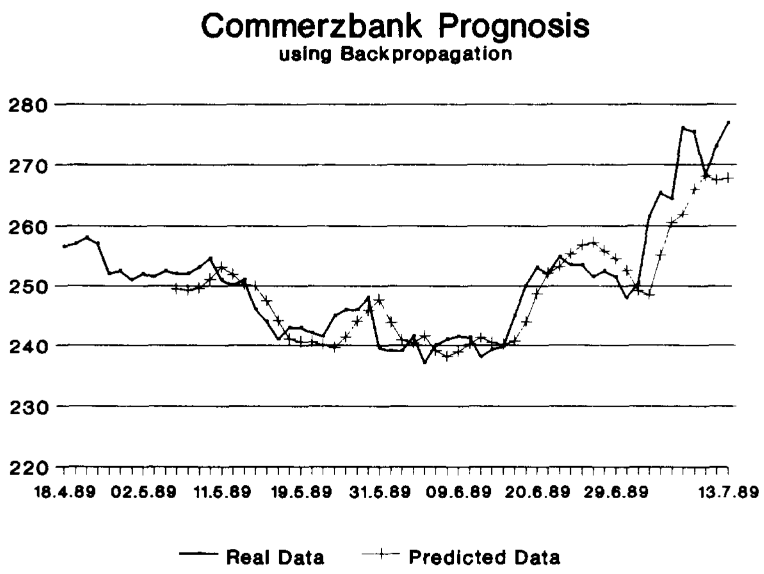

Fig. 6. COMMERZBANK stock prediction with the BACK-PROPAGATION network.

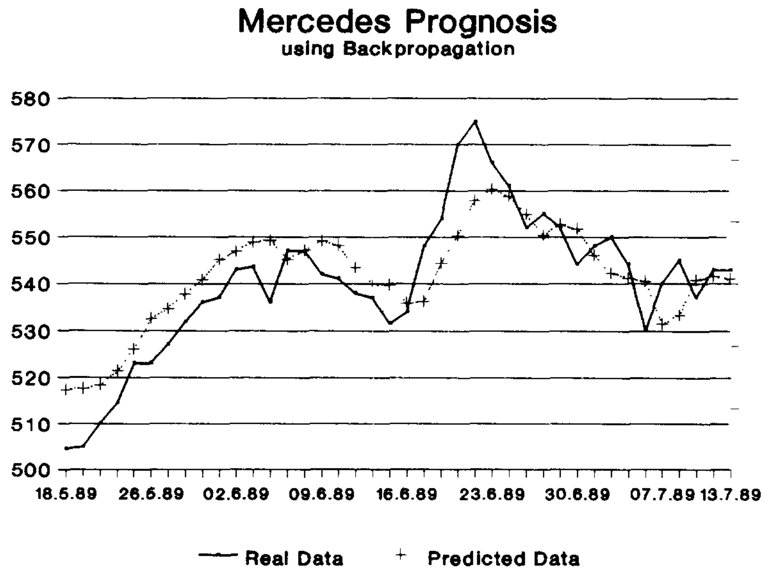

Fig. 7. MERCEDES stock prediction with the BACK-PROPAGATION network.

this simple heuristic is, however, the >> slight variation << from the previous day's price as a recommendation for the following day. The quality with which this >> slight variation << is recognized and characterized is the quality with which the prediction can be improved compared with the obvious heuristic.

3. Further tests

The results presented above were encouraging and have led us to further tests. We are currently concentrating on the following areas:

(1) The selection of other, perhaps more suitable input information for the networks (e.g. variation

quotients from previous day's prices as input or information about stock price behavior on the basis of reference stocks from the same field of business—key word: field dependencies, etc.). We are also currently examining the possibility of designing neural networks as constraint-satisfaction networks for fundamental analytical purposes.

We have estimated that the amount of work necessary to design such a network would be about the same as the amount of work required for the acquisition of knowledge about stock behavior to be used in an expert system for this task. We believe that this would be very time-consuming and expensive.

(2) The study of further network types in terms of their suitability for price predictions (among others, counter propagation, Hopfield, Kohonen, ARTx). The first analyses with Hopfield networks required considerable effort for a suitable transformation and preparation of the input and the definition of an energy function. Other tests using feature extraction with Kohonen-like networks have not yet produced results worthy to mention.

Nevertheless we presume that competitive self-organizing networks are most suitable for predictions. We do, however, intend to systematically examine other types of networks in terms of their suitability, as we are not sure how much complex networks are better as compared to simpler ones. It could be that the increase in prediction capacity of the more complex networks is relatively small, in which case the work involved would probably no longer be worth in practical use.

(3) The discovery of >> best << network types and topologies and the determination of relevant network parameters (e.g. the number of layers and neurons, learning coefficients, connections of layers, threshold values, learning rules and strategies, etc.).

(4) The improvement of the theoretical means at our disposal as aids in designing and analyzing networks.

(5) The development of distinct, problem-oriented network simulators in sequential and parallel >> C <<. For the simulations described above, some 5000 lines of C code were necessary. For the analysis of the performance of complex networks we are currently testing neurocomputers and transferring our simulator to transputer hardware.

(6) We have estimated the statistical relevance of our results. Since we had an accuracy of 63% using 285 predictions in total the probability of this outcome is as low as 1/100 000 which is not too bad. On the 10 to 20 days basis the results were even better as we had an average of 71,3% correct predictions on 300 predictions in total.

The statistical relevance of these results is yet to be confirmed. The test series were too short to permit statistical assessment. In 1990 we will carry out comprehensive statistical analyses over a continuous prediction period of 10 months.

(7) In order to become less dependent on the network types and the various parameters, we are currently exploring the possibility of determining the weighting between the neurons by means of a genetic algorithm and an evolution theoretical approach. We will be reporting on our progress in suitable time.

4. Problems

We cannot deny that we have been faced with complex problems in some of our attempts to determine the suitability of neural networks for prediction of stock prices; for some of these, there is no solution in sight, not even in theory.

Most of the time spent in the project was related to the difficulties of finding the >> best << network topology and setting the corresponding network parameters (see point (3) in Section 3). As a rule even slight parameter changes caused major variations in the behavior of almost all networks. There is no theory available yet which could be used as a guideline to find >> best << networks.

There is a real practical need for a theory of equivalence classes of neural networks and the development of a partial order on these classes defining the relation of complexity of networks and minimal elements (i.e., networks with the smallest number of neurons necessary to solve certain kinds of problems etc.). As classification theory in logic shows, this is possible even for complex mathematical theories. It should at least in principle be possible for subclasses of neural networks. Without any theory of this kind at hand one is almost always forced to proceed using a time-consuming and laborious trial-and-error strategy.

References

[1] M. Hüttner, *Prognoseverfahren und ihre Anwendung* (De Gruyter, Berlin, 1986).

[2] P. Mertens, ed., *Prognoserechnung* (Physica-Verlag, Würzburg, 1981).

[3] E. Schöneburg, N. Hansen, M. Gantert and M. Reiner, Kurzfristige Aktienkursprognose mit Neuronalen Netzen, Expert Informatik GmbH, Berlin, 1989.

[4] B. Widrow and M.E. Hoff, Adaptive switching circuits, in: Anderson and Rosenfeld, eds., *Neurocomputing* (MIT Press, Cambridge, MA, 1988) 126 ff.

Professor Eberhard Schöneburg, born in 1956, studied mathematics, computer science, physics and philosophy at the Berlin Free University and Technical University from 1979 to 1984. After the completion of his studies (degree: Dipl. Math.), he worked as software and systems engineer for SIEMENS AG in Berlin and Boca Raton, Fla, USA (1984 – 1986). At SIEMENS, his area of involvement consisted of the development of intelligent self-test and diagnosis modules for real time applications and the automatic generation of programs.

Professor Schöneburg accepted an offer to DORNIER to take over the direction of the department for new technologies, expert systems and computer security in 1986. In 1988 he founded Expert Informatik GmbH. This company specializes in expert systems applications in banks and in the production area (solvency analyses, production planning), and was the first company in Germany to be asked to develop neuronal networks for a major bank; it also rates among the leaders in the development of anti-computer-virus sytems.

With the chair of artificial intelligence at the Furtwangen/Schwarzwald Technical College, Professor Schöneburg's scientific interest has concentrated on applications of nonclassical logic in robot technology, in the prediction of semi-chaotic time series with neuronal networks (dissertation topic) and in expert systems security. Professor Schöneburg has just completed writing a German textbook on neural networks containing a software simulator in > C < (to appear in 1990).