S C H E D A E I N F O R M A T I C A E
VOLUME 19 2010

Use of Artificial Neural Networks in Forecasting of Financial Time Series of High Frequencies with Stock Exchange Quotations as an Example

MARCIN WASIK[1]
[1]Institute of Computer Science, Jagiellonian University,
Prof. Stanisława Łojasiewicza 6, 30-348 Cracow, Poland
e-mail: *wasik3@op.pl*

1. Introduction

A field of science such as artificial neural networks (ANN) arouses more and more interest all over the world, not only among academics and researchers. Since its beginning in the 1940'ties it has experienced its ups and downs. From the huge fascination in the early stage to nearly falling into oblivion after Minsky's book [10]. Then in the 80'ties interest in ANN experienced a rapid growth which surprised everyone and resulted in a dramatic rise and demand for information about ANN. This interest is also a result of a very dynamic development of computer science in the recent years as well as increasing need for iterative computational models, to which most definitely we can include ANN.

Generally speaking, we can say that artificial neural networks is an advanced technology used as a statistical data modelling tool. It is a tool used in decision making processes. ANN is so successful thanks to its two important characteristics: ability of approximation any given function and ability to generalize gained knowledge. It allows us to find important correlations between processes, which observations are used as input data. It does not require to have any knowledge of analytical form of the described

model. Another big advantage is also ability to conduct computation in parallel resulting in possibility to process huge amounts of data.

An attempt to describe a data processing system in a form of an artificial neural network and resulting practical applications in the modern world of economy is the purpose of this paper. High frequency time series forecasting on the example of trends of the indices on the Warsaw Stock Exchange with the use of artificial neural networks is the main issue of this dissertation. In the following chapters efficacy results of selected types of artificial neural networks in forecasting trends of the index under consideration are shown, together with a debriefing.

2. Examples of implementing artificial neural networks in financial markets

Empirical researches on implementation of artificial neural networks and selected econometric models in the capital market have been conducted for a long time. In the article [16] results of research on implementation of neural networks to forecast directions of short-term trends and to predict price changes in ten-days time horizon are presented. Indicators used for technical analyse were used as the input data, and in the case of share price forecast also some fundamental data was used. The tested type of a neural network was a perceptron with one hidden layer, trained using the back propagation method. Also some empirical research of the size of hidden layer took place, issues of selection and preliminary data processing of the size of training data set and the length of the training process were analysed in brief. The obtained results were very promising. In the case of share price prediction – 68% of forecasts were correct regarding direction. Additionally it was proved that efficacy of this strategy for the Greek market (similarly to the Polish market, it is a developing market with a relatively low effectiveness) was much higher than for the significantly more developed German market.

Another example of using artificial neural networks is connected with choosing the right investment portfolio [5]. The future annual profit of investment in a particular company was chosen as a criterion for selecting right shares. The training data set comprises eighty eight input variables, which include twenty eight fundamental factors (linked with fourteen chosen indicators of a given firm), twenty five values showing of a given firm in its market and thirty five values created basing on seven macroeconomic indicators. Data from 231 firms was used and obtained results were very promising. If we talk about dividing shares into two categories (making profit and making loss), in about 66,4% cases, networks made right decisions, in 26,2% decisions were incorrect and in 7,4% cases the networks were not able to make a decision.

In the paper [13] Polish market forecasting of short term contracts on index WIG20 is presented. As the input data, nineteen values describing dynamism of changes of index WIG20 and chosen contract, were used. The testing and training data was taken from the two-year period (from 2nd February 1998 to 31st January 2000). Models of the network of the type – perceptron were used. They generated significant levels of profit between 50% to 78% in the testing period of around a hundred market sessions. By the way, some comparisons of efficiency between different types of neural networks and neural methods with the classic linear regression were made. They outcome was that a non-linear network of the perceptron type is much more efficient for modelling phenomena under consideration.

In the book [8] an innovative idea of decomposition of time series into a few separate components of different types can be found. Time series used, include absolute changes of WIG in the period between 23rd April 1991 and 3rd November 1997, what covers 1187 values. Authors have carried out two experiments. In both cases one way multi layer networks trained with the use of the Marquardt–Levenberg method were used, the output variable was described by values of six previous series. In the first experiment an original unprocessed series was used. On a few different networks thirty iterations of the Marquardt–Levenberg algorithm were conducted, then the best one was trained in the same way. The obtained network was able to forecast right direction of changes in 61% of cases for tested data. In the second experiment, the previously mentioned decomposition of time series was used. For this purpose the discrete wavelet transform was used in order to separate components of different frequencies and to place them in time. A filter function of the type Daubechies4 was used. Wavelet analysis separated ten components. Each of them was independently trained, analogously to the first experiment, although partial models differed in the size of hidden layers and what follows in the number of connections. The value of the original series was obtained through adding up the theoretical value of each component. In this case the model was able to forecast right direction of changes in 86% of cases, what gives us a very good forecasting result.

3. Forecasting future values of WIG20

3.1. Introduction

In this part of the paper results of the research obtained by forecasting indexes of the largest companies with the use of artificial neural networks are briefly described. In all experiments, data from the period between 1st August

2002 and 31st October 2007 (what gives us 1322 Stock Exchange quotation) was used. As we can see in Fig. 1 it was a period slightly longer than the longest bull market in the history of the Warsaw Stock Exchange.

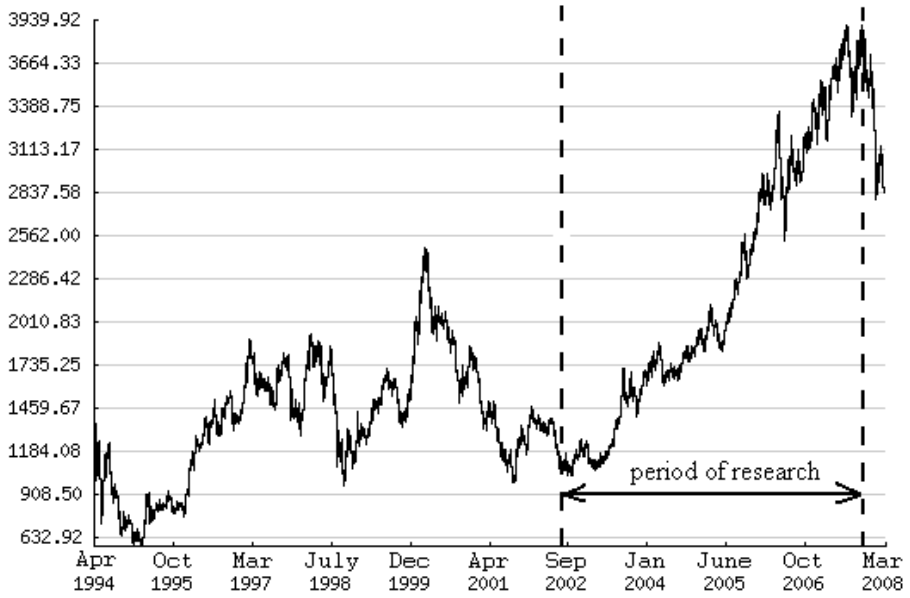

Fig. 1. WIG20

The above chart could suggest a very large domination of increases over decreases, however in the period under investigation, they account for 'only' 53% sessions. So, the model to show any usefulness should achieve minimum the level of 53% correct direction forecasts. Whilst it is generally accepted that forecasting at the level of 60% allows us to think about making profits on the stock exchange.

In order to simulate and test all kinds of neural models the "Statistica Neural Networks" software version 7.1. was used. Three types of networks have been tested – a multi layer perceptron, RBF network and General Regression Neural Network (GRNN). There exist many measures of absolute and relative errors which could be used to assess obtained results, however output time series in many experiments was significantly transformed and the only fully comparable value is a variable determinative a properly forecasted direction of the change of the index (DIR). This value is treated as the most important measure of model's usefulness in practice, as what really interests most stock investors (and most definitely all speculators) is information about changes of discussed values during the next session. An average from five independent experiments and a so-called "committee" are

given as results for each model. A "committee" can be defined as a network with averaged results on the level of each session.

In the following chapters assumptions and results obtained in each experiment are presented.

3.2. First experiments

The research was started with the most common type of neural networks – a multi layer perceptron (MLP). Capabilities of this "classic" network in the area of forecasting financial time series and influence of different methods of teaching on the results, were tested on the examples of 4 small architectures. "Statistica Neural Networks" contains several implemented methods of teaching. Networks described below were tested with the use of four most common methods of teaching: back propagation, back propagation trained with the use of the conjugate gradient method, Quasi-Newton method and Marquardt–Levenberg method. The main aim of this experiment was to test whether any significant statistical differences in results of WIG20 forecasting appear when different methods of teaching are used and whether adding hidden layers markedly improves results.

First experiments results are shown in Tab. 1. The first conclusion after analysing Tab. 1 is the fact that initial results are not very promising. Similar results can be obtained by simply tossing a coin. Efficacy at the level of 50% tells us nothing and has no practical value. Taking into consideration the above results we can list out three main conclusions or, in point of fact, three problematic questions:

- are MPL networks not suitable for forecasting financial time series?
- is forecasting stock exchange not possible basing only on past values?
- is forecasting horizon too narrow (here maximum eleven)?

In the following sections, where more advanced models will be tested, answers for these and many more questions can be found.

As for the above results, it is a difficult task to present any clear rules and correlations basing only on them. Maximum discrepancy between the results of our research is 3%, and taking into consideration that these are averages from only five tests (where the standard deviation was sometimes higher than 3) it is impossible to choose a training algorithm which gives significantly better results than the other ones[1].

[1] There are many papers that try to prove the existence of "better" and "worse" methods for training neural networks. For example in the paper [7], superiority of the Marquardt–Levenberg method over other training algorithms was proved. However research done in that article was of different character to the one presented in this dissertation. There the task of a network was to

Tab. 1. Efficacy of small networks of the perceptron type with the use of different training algorithms

Network description		Training data			Testing data		
		DIR (%)	DIR-increases (%)	DIR-decreases (%)	**DIR (%)**	DIR-increases (%)	DIR-decreases (%)
Architecture: 6-6-1	average	51,8	54	48,8	**50,2**	51	47,6
Algorithm: back propagation	*committee*	*51,9*	*54,2*	*46,5*	***49,3***	*49,6*	*48,7*
Architecture: 6-6-1	average	52,1	55,7	47,4	**50,5**	49,4	51,5
Algorithm: backpropag.+conjug.grad.	*committee*	*52,6*	*56*	*47,9*	***51,5***	*50,3*	*53,1*
Architecture: 6-6-1	average	54,8	56	52,1	**50,8**	51,6	49,2
Algorithm: Quasi-Newton	*committee*	*54,4*	*56,3*	*50*	***54***	*51,4*	*61,3*
Architecture: 6-6-1	average	54,6	55,7	52,2	**51**	53,3	47,6
Algorithm: Marquardt–Levenberg	*committee*	*54,8*	*56,7*	*50,6*	***53,7***	*51,7*	*57,7*
Architecture: 6-12-6-1	average	51,5	56	48,4	**49,6**	49	49,5
Algorithm: back propagation	*committee*	*50*	*54,2*	*45,5*	***46,5***	*44,9*	*47,9*
Architecture: 6-12-6-1	average	52,9	55,7	48	**49,8**	48,9	51,3
Algorithm: backpropag.+conjug.grad.	*committee*	*52,6*	*55,3*	*47,4*	***49,9***	*48,9*	*51,4*
Architecture: 6-12-6-1	average	54,2	56,8	49,8	**49,1**	48,3	50,3
Algorithm: Quasi-Newton	*committee*	*55,6*	*57,6*	*51,9*	***49,5***	*48,7*	*50,9*
Architecture: 6-12-6-1	average	52,5	55,5	47,6	**48**	47,2	49
Algorithm: Marquardt–Levenberg	*committee*	*53,4*	*55,9*	*48,6*	***45,8***	*45,7*	*45,9*
Architecture: 11-11-1	average	51,9	56,3	48,1	**47**	44,8	48,3
Algorithm: back propagation	*committee*	*50,1*	*54,7*	*46,2*	***46,5***	*44,9*	*47,9*
Architecture: 11-11-1	average	53	56	48,5	**46,8**	46,2	47,5
Algorithm: back	*committee*	*53,1*	*55,9*	*48,5*	***44,8***	*44,8*	*44,9*
Architecture: 11-11-1	average	54,6	57,5	50,4	**48,6**	47,7	49,8
Algorithm: Quasi-Newton	*committee*	*55,1*	*58*	*51,1*	***46,8***	*46,2*	*47,7*
Architecture: 11-11-1	average	54,8	57,4	50,8	**47,1**	46,6	47,8
Algorithm: Marquardt–Levenberg	*committee*	*55,4*	*57,6*	*51,6*	***42,8***	*43,2*	*42,2*
Architecture: 11-22-11-1	average	53,1	55,4	49,7	**49,3**	48,4	51,1
Algorithm: back propagation	*committee*	*54,8*	*56,1*	*51,2*	***49,8***	*49,1*	*51,9*
Architecture: 11-22-11-1	average	52,4	55,5	47,7	**49,8**	48,8	51,2
Algorithm: back	*committee*	*52,8*	*55,6*	*48,1*	***50,5***	*49,4*	*52*
Architecture: 11-22-11-1	average	54,3	57,2	50,1	**47,1**	46,3	48
Algorithm: Quasi-Newton	*committee*	*53,5*	*56,5*	*49,1*	***47,8***	*47*	*48,9*
Architecture: 11-22-11-1	average	52,5	55,8	48	**47,3**	46,5	48,2
Algorithm: Marquardt–Levenberg	*committee*	*52,3*	*55,7*	*47,6*	***50,8***	*49,7*	*52,2*

compute results of function $z = f(x, y)$ with x and y given. The function was complex, but its form was known. Here we deal with financial time series, where it is very difficult to find any correlations.

However, it is worth mentioning that the classic method of teaching – back propagation offered us one of the best results. But, when we look at the results of networks on the training data we can notice that back propagation gives the worst results. It does not, however, puts this method in an unfavourable position. We can assume, basing on this, that this method is to the smallest degree affected by the process of learning by heart input vectors and that it best generalizes the obtained knowledge. Considering the above arguments and the fact that it is the fastest method of the four chosen, back propagation was used in the following sections to test bigger and more advanced architectures, where the teaching speed plays a key role.

It was surprising to find out that the smallest (6-6-1) of tested architectures gave the best results. So it is only natural that we can find it in several papers, e.g. [8]. The use of committees also did not improve results as expected. In only a few cases we can notice 2–3% increase of efficacy. In most of the cases we not only did not benefit from the use of committees but even suffered from a few percent decrease in results. This decrease is most visible on the testing data. On the training data, committees usually did not worsen results.

Additionally, the standard deviation was calculated for each experiment. In order to have a better clarity, the results are given in a separate table.

Tab. 2. The standard deviation of errors

Network description	Standard deviation DIR on training data	Standard deviation DIR on testing data
6-6-1 (back propagation)	1,4	3,4
6-6-1 (back propagation + gradients)	0,4	2,3
6-6-1 (Quasi-Newton method)	0,8	2,1
6-6-1 (Marquardt–Levenberg method)	0,8	3,5
6-12-6-1 (back propagation)	2,8	2,3
6-12-6-1 (back propagation + gradients)	1,2	1,6
6-12-6-1 (Quasi-Newton method)	0,6	1,9
6-12-6-1 (Marquardt–Levenberg method)	0,6	1
11-11-1 (back propagation)	1,9	1,2
11-11-1 (back propagation + gradients)	0,8	2,4
11-11-1 (Quasi-Newton method)	1,2	2,4
11-11-1 (Marquardt–Levenberg method)	1	2,6
11-22-11-1 (back propagation)	2,1	0,9
11-22-11-1 (back propagation + gradients)	1,3	2,9
11-22-11-1 (Quasi-Newton method)	1,6	1,6
11-22-11-1 (Marquardt–Levenberg method)	1,1	3

In our case considerations concerning standard deviations of network results are not the most important so we will not discuss it in much detail. Nonetheless, we have to mention two, one might say obvious, conclusions:

- standard deviation on testing are larger than on the training data,
- small networks with one hidden layer are characterized by bigger variability of results.

Smaller networks with one hidden layer are more affected by different fluctuations. To stabilize results we can either add another hidden layers or enlarge a network.

In the next section, results of MLP networks with a much larger forecasting horizon (up to 200) will be presented.

3.3. MLP Networks and large forecasting horizons

In the previous section abilities of small MLP networks on the financial market were presented. The forecasting horizon was rather small – maximum eleven. It could have been a cause of not good results, that is why for the next stage of research we will use up to 200 past values. However, it is related with the use of much bigger architectures and what follows, much longer time needed for teaching such networks.

All of the networks were trained with the use of the back propagation method with momentum (1000 iterations), and then trained with the use of the conjugate gradient method (500 iterations). The coefficient of teaching was 0,6 at the beginning and was gradually decreasing with successive iterations to 0,01[2], and the component of momentum was 0,3. Numerous experiments showed that networks with two hidden layers give the best results. One and three hidden layers gave notably worse results. In the case of two hidden layers, a rule was used, which stated that the first layer had a number of the input data and the second was a half of the first. Results are presented in Tab. 3.

Extending the forecasting horizon brought the expected improvement of results. Networks systematically improved their forecasting abilities up to the moment when 100 past values were used. Over this number, excess of information became visible and networks were less efficient. To summarize, forecasting horizon of 100 was optimal for MLP networks.

[2] This value of the coefficient of teaching and its changes during the teaching process were mentioned in [19].

Tab. 3. Perceptron and large forecasting horizons

Network description		Training data			Testing data		
		DIR (%)	DIR-increases (%)	DIR-decreases (%)	DIR (%)	DIR-increases (%)	DIR-decreases (%)
25-25-13-1	average	52,3	56	47,8	**48,4**	47,5	49,4
	committee	*52,9*	*56,5*	*48,4*	***49***	*48,1*	*50*
50-50-25-1	average	52,9	56,8	48,4	**53**	52	54,1
	committee	*52,1*	*55,9*	*47,5*	***56,6***	*55,6*	*57,5*
75-75-38-1	average	53,7	57,9	49,3	**56,2**	54,6	57,9
	committee	*55,5*	*59,4*	*51,2*	***60,4***	*58,7*	*62*
100-100-50-1	average	56,9	61,4	52,1	**60,3**	57,5	63,4
	committee	*59,7*	*63,7*	*55,2*	***62,3***	*58,6*	*67,2*
140-140-70-1	average	57,9	62,6	52,6	**57,3**	55,1	60
	committee	*59,9*	*64,3*	*54,8*	***59,8***	*56,8*	*64*
150-150-75-1	average	57,9	62,7	52,5	**58,2**	55,6	61,3
	committee	*62,5*	*66,5*	*57,6*	***61,1***	*58,1*	*65*
200-200-100-1	average	59,9	65	54,5	**58,9**	56,5	61,8
	committee	*62*	*66,4*	*56,9*	***61,8***	*58,7*	*66*

Looking at the above results we come to very interesting conclusions for which it is difficult to find a rational explanation. First of all, in four out of seven cases, forecasts on the testing data were better than on the training data. Another interesting thing is that in all tests on the training data forecasts of increases were better than the forecasts of decreases. However, on the testing data, we can see the opposite situation. Differences were relatively significant.

Efficacy of networks at the level of 60% provides us with some optimism as these results offer basis for practical applications as well as for future experiments – see Tab. 3.

3.4. Input data transformation

All of the previous experiments took place with the use of raw data. In this section, results of networks trained on the previously processed data will be presented. We can find many methods of preprocessing data in literature. Here, some of the most common ones will be discussed:

- absolute changes of time series values ($x_t - x_{t-1}$),
- absolute changes with correction[3] ($x_t - x_{t-1}$),
- percentage changes ($x_t - x_{t-1}$)/ x_t,
- percentage changes with correction[4] ($x_t - x_{t-1}$)/ x_t,
- thresholds –1, 1 (index decrease marked as –1, increase as 1).

New possibilities were tested on a small MLP network with 6-6-1 architecture (six input data, one hidden layer). Results are presented in Tab. 4.

Tab. 4. Use of data transformation on the example of perceptron type 6-6-1

Network description		Training data			Testing data		
		DIR (%)	DIR-increases (%)	DIR-decreases (%)	**DIR (%)**	DIR-increases (%)	DIR-decreases (%)
Unchanged series	average	52,1	55,7	47,4	**50,5**	49,4	51,5
	Committee	*52,6*	*56*	*47,9*	***51,5***	*50,3*	*53,1*
Absolute changes	average	55,3	55	55,8	**53,7**	53,5	55
	committee	*55,9*	*55,5*	*57,3*	***54,5***	*53,9*	*57,8*
Absolute changes with correction	average	56	55,7	57,4	**52,3**	53,2	48,6
	committee	*57,2*	*56,3*	*61,6*	***53,2***	*53,6*	*51*
% changes	average	54,8	55,3	52,5	**49,3**	48,7	51,9
	committee	*55,1*	*55,5*	*53,6*	***49,3***	*48,7*	*51,7*
% changes with correction	average	56,6	56,3	58,3	**51**	49,7	57,7
	committee	*57*	*56,3*	*60,5*	***51,2***	*49,8*	*60*
Thresholds	average	55,1	58,6	52,7	**51,5**	59,5	45,8
	committee	*57,2*	*59,8*	*54,9*	***51,3***	*58,2*	*45,3*

Only the use of percentage change worsened results. Another four methods of preprocessing the input data helped to achieve better results than unprocessed series. The best solution was the use of absolute changes which gave 3% improvement. Nonetheless, the obtained results are still not good enough for any practical use. The use of committees also bettered results, but not as significantly as in the previous experiments (Tab. 4).

[3] If the change was over 3% (what is an unusual situation) then this value was set to 3%.
[4] The same situation as above.

The results obtained in this section give hope that with the use of large forecasting horizons and absolute changes, it will be possible to get results better than those presented in Tab. 4. Such a combination is shown in Tab. 5. All details of the process are the same as those described in the previous section.

This time, the results of our experiments are very disappointing. As for the testing data, efficacy of networks was only around 50% and using committees did not improve the obtained results. The best result was achieved when the forecasting horizon was 140, however the result was still lower than for the 6-6-1 network. Very good results on the training data are a curiosity. We can see that generalization of networks decreases as we use absolute changes. Networks in this case are negatively affected by the process of learning by heart of the input data. With the use of raw values described processes seem to be much more beneficial.

Tab. 5. Input data transformation and perceptrons with large forecasting horizons

Network description		Training data			Testing data		
		DIR (%)	DIR-increases (%)	DIR-decreases (%)	**DIR (%)**	DIR-increases (%)	DIR-decreases (%)
25-25-13-1	average	58,3	58,7	57,7	**52,1**	53,2	49,8
	committee	*61,6*	*60,7*	*63,5*	***51,9***	*52,8*	*49,4*
50-50-25-1	average	61,9	62,2	61,4	**49,6**	51,1	46,8
	committee	*65,3*	*64,1*	*67,8*	***50,2***	*51,5*	*47,3*
75-75-38-1	average	64	64,5	63,4	**48,2**	49,9	45,2
	committee	*68,3*	*66,6*	*71,8*	***47,4***	*48,2*	*44,1*
100-100-50-1	average	67,5	68,3	66,7	**49**	50,7	46,7
	committee	*71,2*	*70,2*	*72,9*	***48,6***	*50,3*	*45,5*
140-140-70-1	average	69,7	69,8	69,5	**52,8**	54,1	50,6
	committee	*76,3*	*72,9*	*83,1*	***52,6***	*53,6*	*50,6*
150-150-75-1	average	70,8	70	72,2	**51,1**	52,9	48
	committee	*77,1*	*73,2*	*85,2*	***49,8***	*51,7*	*46,3*
200-200-100-1	average	72	71,8	72,5	**51,5**	53	49,4
	committee	*78,6*	*75,6*	*83,9*	***51,4***	*52,6*	*48,8*

In order to finally determine usefulness of the previously mentioned series transformations, we carry out another series of experiments, showing results

of networks with the use of more preprocessing methods on the example of a 140-140-70-1 architecture. Technical details of the training process stay the same as in the whole section. Results are presented in Tab. 6.

Tab. 6 confirms the fact that using input data transformation does not improve network efficacy. This time the best results were achieved with the use of absolute changes, but they are still lower by about 4% compared to the case with raw series.

Tab. 6. Use of data transformation on perceptron type 140-140-70-1

Network description		Training data			Testing data		
		DIR (%)	DIR-increases (%)	DIR-decreases (%)	**DIR (%)**	DIR-increases (%)	DIR-decreases (%)
Unchanged series	average	57,9	62,6	52,6	**57,3**	55,1	60
	committee	*59,9*	*64,3*	*54,8*	***59,8***	*56,8*	*64*
Absolute changes	average	69,7	69,8	69,5	**52,8**	54,1	50,6
	committee	*76,3*	*72,9*	*83,1*	***52,6***	*53,6*	*50,6*
Absolute changes with correction	average	76,8	75,9	78,3	**53**	55,9	48,8
	committee	*82,8*	*78,9*	*89,3*	***57,4***	*59*	*54,6*
% changes	average	72,4	72,7	72	**51,8**	52	51,4
	committee	*81*	*78,9*	*84,2*	***51,1***	*51,6*	*50,5*
% changes with correction	average	84,3	84,3	84,2	**50,1**	49,2	51,1
	committee	*86,9*	*86,6*	*87,2*	***47,4***	*47,1*	*48*
Thresholds	average	89,2	89	89,4	**49,7**	57,9	39,1
	committee	*89,9*	*89,8*	*90*	***49***	*57*	*38*

To summarize, input data transformations which were used in this section, do not improve network efficacy at all. All of the used methods of preprocessing not only, did not improve the obtained results but they also worsened them by a few percent (of course, taking into consideration directions of changes of index WIG20). Most likely, transformations of original series, described in this section, lead to loss of important information that helps to forecast, and enlarging the forecasting horizon only increases the level of data dredging and lowers the possibility of training data generalization. Experiments described in this section tend to support this argument. Small architectures MLP, e.g. 6-6-1 (Tab. 4) proved in many cases to be better than large ones: 140 140 70-1 (Tab. 6).

3.5. Financial time series forecasting with the use of RBF networks

In the previous section we obtained MLP networks efficacy at the level of 60%. It is a maximal result that was obtained with the use of multi layer perceptrons and with original series values in the training process. In this section RBF network's ability in the area of financial time series forecasting will be presented.

In the first series of experiments original series were used and RBF networks, characterized by the input layer of 25 to 200 neurons and variable number of neurons in one hidden layer, were tested. Results are presented in Tab. 7.

Tab. 7. Efficacy of RBF networks

Network description		Training data			Testing data		
		DIR (%)	DIR-increases (%)	DIR-decreases (%)	**DIR (%)**	DIR-increases (%)	DIR-decreases (%)
25-25-1	average	50,6	55,1	46,4	**53,4**	52,7	54
	committee	*51*	*55,4*	*46,7*	***52***	*51,2*	*52,6*
25-50-1	average	52,8	56,7	48,4	**48,7**	47,9	49,7
	committee	*53,5*	*57,2*	*49,1*	***49,3***	*48,4*	*50,4*
25-100-1	average	58	61,2	54	**43,7**	43,2	44,2
	committee	*59,2*	*62,2*	*55,4*	***38,9***	*39,4*	*38,3*
50-50-1	average	52,4	56,5	47,9	**55,4**	54,5	56,2
	committee	*53*	*57,1*	*48,6*	***54,1***	*53,1*	*55,2*
50-100-1	average	54,7	58,6	50,3	**53,4**	52,4	54,3
	committee	*56,6*	*60,1*	*52,5*	***52,1***	*51,1*	*53*
50-200-1	average	60,7	63,7	57	**48,1**	47,2	49
	committee	*64*	*66,3*	*61*	***47,6***	*47*	*48,4*
100-100-1	average	57,4	61,8	52,7	**57,9**	55,2	61,1
	committee	*58,8*	*62,3*	*54,3*	***59,8***	*57*	*62,9*
100-200-1	average	64	67,3	60	**60,3**	57,2	64,1
	committee	*67,1*	*70*	*63,6*	***60,5***	*57,4*	*64,3*
100-300-1	average	69,6	72,4	66,2	**60,7**	57,9	63,8
	committee	*74,1*	*76,5*	*71,2*	***61,9***	*59,1*	*65,2*
100-400-1	average	74,7	76,9	72,1	**58**	55,2	61,4
	committee	*77,5*	*78,9*	*75,8*	***61,6***	*58,7*	*64,9*

140-280-1	average	70,5	73,6	66,8	**63,8**	61,2	66,8
	committee	*74,6*	*76,8*	*71,9*	***67***	*64*	*70,6*
140-400-1	average	76,5	78,3	74,1	**64,9**	61,6	69,2
	committee	*81,5*	*81,9*	*81*	***66,7***	*63,1*	*71,4*
140-600-1	average	84,1	85,4	82,4	**63**	60,2	66,6
	committee	*86,1*	*86,8*	*85,2*	***64,5***	*61,4*	*68,3*
140-800-1	average	92,3	93,3	91,1	**64**	61,2	67,3
	committee	*93,7*	*94,4*	*92,8*	***63,8***	*61,1*	*66,9*
150-300-1	average	73,2	76,1	69,7	**66,3**	63,3	70
	committee	*77,9*	*79,9*	*75,3*	***70,5***	*67,3*	*74,2*
150-450-1	average	79,3	81,3	76,8	**65,4**	62,5	68,7
	committee	*83,7*	*84,4*	*82,7*	***66,9***	*63,8*	*70,6*
150-600-1	average	84,6	86,5	82,2	**65,8**	62,9	69,2
	committee	*86,9*	*88,4*	*84,9*	***65,5***	*62,6*	*68,8*
150-800-1	average	92,9	94,1	91,5	**63,4**	60,2	67,6
	committee	*95,2*	*96,3*	*93,8*	***62,5***	*59,5*	*66,4*
200-200-1	average	69,2	72,7	65	**64,7**	61,8	68,3
	committee	*73*	*76,2*	*69,3*	***65,3***	*61,6*	*70,3*
200-400-1	average	79,3	81,1	77	**68,2**	65	72,3
	committee	*82,8*	*83,9*	*81,4*	***70,2***	*66,7*	*74,6*
200-600-1	average	87,4	88,7	85,9	**69,2**	65,8	73,4
	committee	*89,9*	*90,6*	*89,1*	***68,3***	*64,4*	*73,5*
200-800-1	average	91,3	92,2	90,3	**68**	64,6	72,5
	committee	*93,3*	*93,7*	*92,8*	***69,5***	*65,3*	*75*

It is easy to see that RBF networks have advantage over MLP networks in financial time series forecasting. With the forecasting horizon between 25 and 100 the advantage is insignificant, but above this level, the efficacy of networks increased by almost 10% with 200 past series values used as the input data. The best networks and their committees were characterized by efficacy at the level of almost 70% which is a great achievement. The forecasting horizon equal 200 seems to be optimal in the case of RBF networks as experiments on larger networks did not bring any improvements of the results.

The type of the network under consideration gave best results when the hidden layer was 2–3 times bigger than the input layer. The bigger layer led to a situation, when the network started to learn by heart the input data and decreased generalization of results. The used committees in this case gave us

1–2% increase of results. Only it the case of two smaller architectures, the efficacy of networks decreased.

RBF networks were also tested with the use of the preprocessed input data. Tests were done on a few architectures and the results on the 100-200-1 network are presented in Tab. 8.

Tab. 8. RBF networks and input data transformation

Network description		Training data			Testing data		
		DIR (%)	DIR-increases (%)	DIR-decreases (%)	**DIR (%)**	DIR-increases (%)	DIR-decreases (%)
Unchanged series	average	64	67,3	60	**60,3**	57,2	64,1
	committee	*67,1*	*70*	*63,6*	***60,5***	*57,4*	*64,3*
Absolute changes	average	63,7	63	65,1	**47,9**	49,8	43,7
	committee	*71,1*	*69,7*	*73,5*	***46,4***	*48,7*	*41,2*
Absolute changes with correction	average	62,4	61,5	64,9	**53,1**	54,7	48,8
	committee	*72,7*	*69,5*	*79*	***54,6***	*55,3*	*52,3*
% changes	average	62,4	62	63,1	**49,6**	49,8	49,1
	committee	*72*	*70*	*75,6*	***49,8***	*50*	*49,2*
% changes with correction	average	63,2	59,6	88,1	**48,4**	49,5	47,7
	committee	*76,5*	*69,7*	*90*	***48,2***	*49,3*	*47,6*
Thresholds	average	61,6	57,4	90,2	**52,3**	54,2	49,8
	committee	*74,9*	*67,4*	*92,3*	***51,4***	*53,6*	*49,5*

Similarly as in the case of the perceptron, efficacy of RBF networks trained on the preprocessed data is much worse than in the case of original series. Even though in this case the best solution was the use of absolute changes with correction, still the result offers very little forecasting value. Committees in this case worsened the obtained results.

To summarize, the use of RBF networks helped us to increase efficacy of artificial neural networks by almost 10%. Results at the level of 70% correctly forecasted directions of change are a considerable result and offer much forecasting value. Their advantage over the multi layer perceptron is visible only when we consider large architectures and the forecasting horizon equal or larger than 100. It seems that RBF networks have better ability to analyse high frequency data such as financial time series and especially WIG20 index quotation.

3.6. Use of GRNN networks in time series forecasting

General regression neural networks are a specific type of neural networks, mostly due their structure and the way they work. GRNN networks have a pre-set architecture and are always composed of four layers. First radial – hidden layer has as many neurons as the number of training data cases. For each of them, one neuron is available which stretches out over his case a "bell" of a suitable Gaussian function. Training data cases, which are "copied" to neurons of the hidden layer estimate network answers also for testing data. Weights, which depend on the distance between point tested and particular training points, set once at the beginning are not modified in the training process. That is why training of GRNN networks is so fast. Second hidden layer has one neuron more than the input layer, so in our case two. One of them calculates a weighted sum from the previous layer and the second sums up weighted coefficients.

Tab. 9. Efficacy of GRNN networks

Forecasting horizon	Training data			Testing data		
	DIR (%)	DIR-increases (%)	DIR-decreases (%)	**DIR (%)**	DIR-increases (%)	DIR-decreases (%)
5	49,4	53,6	44,8	**60**	58,2	62
10	49,5	53,8	45,3	**60,9**	59,7	62
15	49,4	53,8	45,2	**60,1**	59,1	61,1
25	50,5	54,8	46,1	**57,4**	56,2	58,6
50	52,7	56,8	48,3	**57,9**	56,3	60
100	59	63	54,4	**64,1**	60,9	67,7
140	64,6	68,2	60,2	**68,5**	64,7	73,3
150	66,5	69,7	62,5	**67,3**	63,3	72,6
200	69,8	72,8	66	**69,5**	65,3	75

The network output is a quotient of those values, that is, weighted average of first hidden layer outputs. Considering that effects of GRNN networks work on a given data set are always the same, it makes no sense to use average and committees. Efficacy of this type of networks is presented in Tab. 9.

As we can see in the above table GRNN networks achieve as good results as RBF networks and in the case of small architectures they are even better. With a maximum forecasting horizon they also reached efficacy of almost 70%.

Similarly as in the case of other networks, also here in tests on the training data, forecasts of increases were better than the forecasts of decreases. However on the testing data, we can see the opposite situation. A strange situation or even a phenomenon is a fact that network efficacy on the training data was slightly worse than on the testing data. This was especially visible in the case of small architectures.

GRNN networks were also tested in order to check usefulness of series transformation but once again the outcome was disappointing and there is not point in presenting these results.

4. Concluding remarks

In Section 3 three main types of artificial neural networks were tested: a multi layer perceptron (MLP), radial basis function (RBF) and finally general regression neural networks (GRNN).

First experiments dealt with perceptions and small forecasting horizons (maximum eleven). Also some training algorithms were tested. The best results were obtained with the use of the classical back propagation technique supported by the conjugate gradient method. However, all of the results were around 50% efficacy of forecasting direction of changes, so these results have no practical value. Slightly better efficacy was achieved by enlarging the forecasting horizon to maximum 200 historical values. Thanks to this operation the perceptron with two hidden layers and over 100 historical values used could predict changes of direction in about 60% of cases. When committees were used results were slightly over 60%. This level of results gives a basis to practical use in financial forecasting.

Next sections of the paper concerned transformation of the input data. The time series under consideration was first pre-processed and then used for neural networks training, also those of the perceptron type. However, it helped only in the case of a small 6-6-1 architecture, next attempts of enlarging the forecasting horizon did not increase the results and the level of 53% turned out to be maximal. It seems that series transformation led to loss of some information important for the forecasting process.

In the sequel efficacy of RBF networks was tested. In this case we can talk about very satisfying results. Efficacy at a very promising level of 70% was achieved. This result most definitely offers us practical value. At the same time some regularities and correlations regarding the size of radial and input layers in this type of networks became more visible.

The last section concerned research focused on GRNN networks. General regression neural networks also achieved a high level of efficacy and similarly to RBF networks reached the level of almost 70%.

Looking at all of the results presented in this dissertation we come into a number of interesting conclusions, which are described in detail in particular sections.

5. References

[1] Azoff E. M.; *Neural Network Time Series Forecasting of Financial Markets*, Wiley, New York 1994.

[2] Doman M., Doman R.; *Analiza dynamicznego polskiego rynku akcji na podstawie notowań indeksu WIG*, Zeszyty Naukowe AE w Poznaniu, 18, Prace z ekonometrii finansowej, Poznań 2002.

[3] Garsztka P., Łażewski M.; *Wykorzystanie sieci neuronowych do prognozowania szeregów finansowych o wysokiej częstotliwości, na przykładzie notowań akcji na GPW w Warszawie*, Zeszyty Naukowe AE w Poznaniu, 55, Prace z ekonometrii finansowej, Poznań 2005.

[4] Gately E.; *Sieci neuronowe. Prognozowanie finansowe i projektowanie systemów transakcyjnych*, WIG-Press, Warszawa 1999.

[5] Kryzanowski L., Galler M., Wright D.; *Using Artificial Neural Networks to Pick Stocks*, Financial Analysts Journal, 1993.

[6] Łażewski M., Matuszewski P., Zator K.; *Nieliniowe charakterystyki szeregów czasowych cen akcji z notowań ciągłych Giełdy Papierów Wartościowych w Warszawie*, Zeszyty Naukowe AE w Poznaniu, 18, Prace z ekonometrii finansowej, Poznań 2002.

[7] Lula P.; *Badania porównawcze wybranych metod uczenia jednokierunkowych sieci neuronowych*, Zeszyty Naukowe UEK w Krakowie, 3, Taksonomia, Kraków 1996.

[8] Lula P.; *Jednokierunkowe sieci neuronowe w modelowaniu zjawisk ekonomicznych*, Wydawnictwo Akademii Ekonomicznej w Krakowie, Kraków 1999.

[9] Lula P.; *Wstępna analiza danych w procesie modelowania przy użyciu sieci neuronowych*, Zeszyty Naukowe UEK w Krakowie, 493, Prace z zakresu informatyki i jej zastosowań, Kraków 1997.

[10] Minsky M., Papert S.; *Perceptrons*, MIT Press, Cambridge 1969.

[11] Morajda J.; *Analiza wpływu początkowych parametrów uczenia na efektywność sieci neuronowych w prognozowaniu finansowym*, Zeszyty Naukowe UEK w Krakowie, 641, Prace z zakresu informatyki i jej zastosowań, Kraków 2004.

[12] Morajda J.; *Czynniki efektywnego wykorzystania sieci neuronowych w procesie modelowania rynków finansowych*, Zeszyty Naukowe UEK w Krakowie, 604, Prace z zakresu informatyki i jej zastosowań, Kraków 2002.

[13] Morajda J.; *Neural networks as predictive models in financial futures trading*, Proceedings of the 5th Conference "Neural Networks and Soft Computing" in Zakopane 6–10.06.2000, Częstochowa 2000.

[14] Orzeszko W.; *Krótkoterminowe prognozowanie chaotycznych szeregów czasowych*, Przegląd Statystyczny 3, 2004.

[15] Refenes P.; *Neural Networks in the Capital Markets*, John Wiley & Sons, Sydney 1995.

[16] Siriopoulos C., Markellos R.N., Sirlantzis K.; *Application of Artificial Neural Networks in Emerging Financial Markets*, Neural Networks, 1996.

[17] Tadeusiewicz R.; *Możliwości i problemy wykorzystania sieci neuronowych w prognozowaniu procesów gospodarczych*, Zeszyty Naukowe UEK w Krakowie, 493, Prace z zakresu informatyki i jej zastosowań, Kraków 1997.

[18] Tadeusiewicz R.; *Sieci neuronowe*, Akademicka Oficyna Wydawnicza, Warszawa 1993.

[19] Tarczyński W.; *Analiza dyskryminacyjna na giełdzie papierów wartościowych*, Przegląd Statystyczny 1–2, 1996.

[20] Trippi R.R., DeSieno D.; *Trading Equity Index Futures with a Neural Network*, The Journal of Portfolio Management, 1992.

Received June 6, 2010