ACTA UNIVERSITATIS LODZIENSIS
FOLIA OECONOMICA 225, 2009

*Tomasz Kozdraj**

USING ARTIFICIAL NEURAL NETWORKS TO PREDICT STOCK PRICES

Abstract

Artificial neural networks constitute one of the most developed conception of artificial intelligence. They are based on pragmatic mathematical theories adopted to tasks resolution. A wide range of their applications also includes financial investments issues. The reason for NN's popularity is mainly connected with their ability to solve complex or not well recognized computational tasks, efficiency in finding solutions as well as the possibility of learning based on patterns or without them. They find applications particularly in forecasting stock prices on financial markets.

The paper presents the problem of using artificial neural networks to predict stock prices on the example of the Warsaw Stock Exchange. It considers the general framework of neural networks, their potential and limitations as well as problems faced by researcher meets while using neural networks in prediction process.

Key words: neural networks, financial markets, financial forecasting.

1. Introduction

Artificial neural networks are one of the most developed branches of artificial intelligence. A wide range of their applications also includes economic issues.

The main reason for its popularity can be seen as the ability to solve complex or not well recognized computational tasks, efficiency in finding solutions, ability to generalization, as well as the capability of learning based on patterns or without them.

---

* PhD., Chair of Statistical Methods, University of Łódź.

Despite working of artificial neural networks imitates neural models, most of neural network models are based on strictly mathematical conceptions which do not have much in common with real neurophysical basis (see N e e l e k a n t a, D e G r o f f, 1994; H u, H w a n g, 2002).

The first fundamental modelling of neural nets was proposed in 1943 by McCulloch and Pitts in terms of a computational model of „nervous activity". The McCulloch-Pitts neuron was a binary device and each neuron had a fixed threshold logic. This model led the works of John von Neumann, Marvin Minsky, Frank Rosenblatt, and many others.

Figure 1 represents a scheme of well known model of an artificial neuron. Such a neuron consists of $n+1$ weighted inputs, transformation unit and one output.

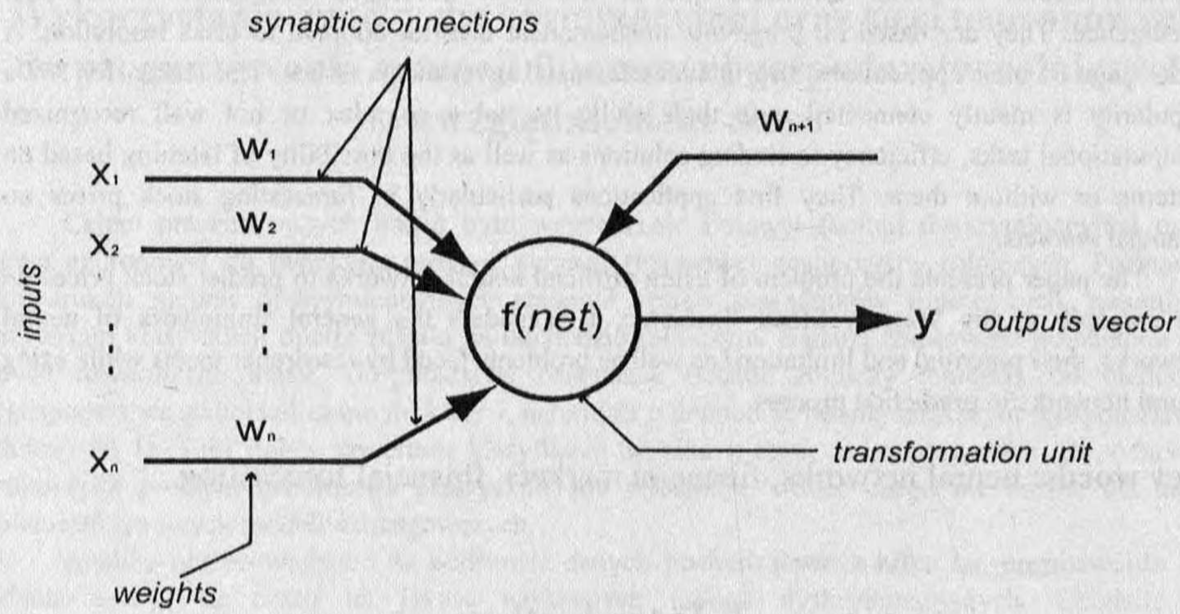

Fig. 1. Sample neuron scheme

The output value of the single neuron is described by the following equation:

$$y = f(net) = f\left(\mathbf{w}^T \mathbf{x}\right) \tag{1}$$

where w is a vector of weights (weighted synaptic connections), x is a vector of input signal values and function f is called the activation function or neuron function.

Neuron scheme can be also presented in a different form (see Fig. 2).

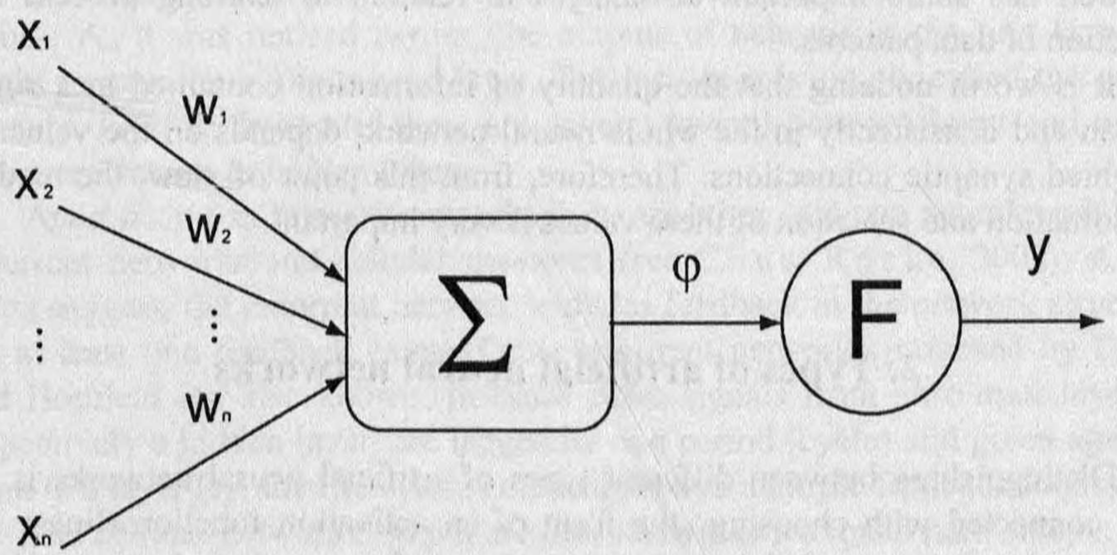

Fig. 2. An alternative neuron scheme

In that case the neuron model consists of two blocks: a summation block Σ and an activation block *F*. The activation function can be expressed in a linear or nonlinear form. Among proposed activation functions for different artificial neural networks, the most common are: step, sigmoid, Gaussian and others (see Tadeusiewicz, 1993; Hu, Hwang, 2002). In particular:

- The linear function $f(x) = ax + b$
- Sigmoid: $f(x) = \frac{1}{1 + e^{(-\beta x)}}$
- Hyperbolical tangent: $f(x) = \tanh\left(\frac{\beta x}{2}\right) = \frac{1 - e^{(-\beta x)}}{1 + e^{(-\beta x)}}$
- Inverted tangent: $f(x) = \frac{2}{\pi}\tan^{-1}\left(\frac{\beta x}{2}\right)$
- The threshold function (step function): $f(x) = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \end{cases}$
- Gaussian function: $f(x) = \exp\left[\frac{-(x-\mu)^2}{\sigma^2}\right]$ for given parameters μ and σ
- Sinusoidal function: $f(x) = \sin(\beta x)$

Most sources suggest using nonlinear activation functions. Particularly, they recommend sigmoid and tangent based functions. The usage of nonlinear function has some important advantages in relation to learning process and detection of data patterns.

It is worth noticing that the quantity of information contained in a single neuron and consistently in the whole neural network, depends on the values of weighted synaptic connections. Therefore, from this point of view, the method of estimation and selection of these values is very important.

2. Types of artificial neural networks

Distinguishing between different types of artificial neural networks is not only connected with choosing the form of an activation function (linear and nonlinear neural networks) but also with types of connections between single neurons. The type of connections between processing neurons (links) as well as other factors determines significantly the topology of neural network and the applied learning algorithm.

While there are numerous different artificial neural network architectures that have been studied by the researcher, the most successful applications in economic problems have been feedforward networks. These are networks in which there are no feedbacks and links between two neurons in both directions.

They are created as a result of clusterization of single neurons in a form of layers and then network layers are linked with any one another. This means that the outputs of neurons in one layer are inputs to neurons in the next layer. Therefore links between neurons in the same layer and backward connections cannot exist.

Figure 3 shows an example of a multilayer neural network.

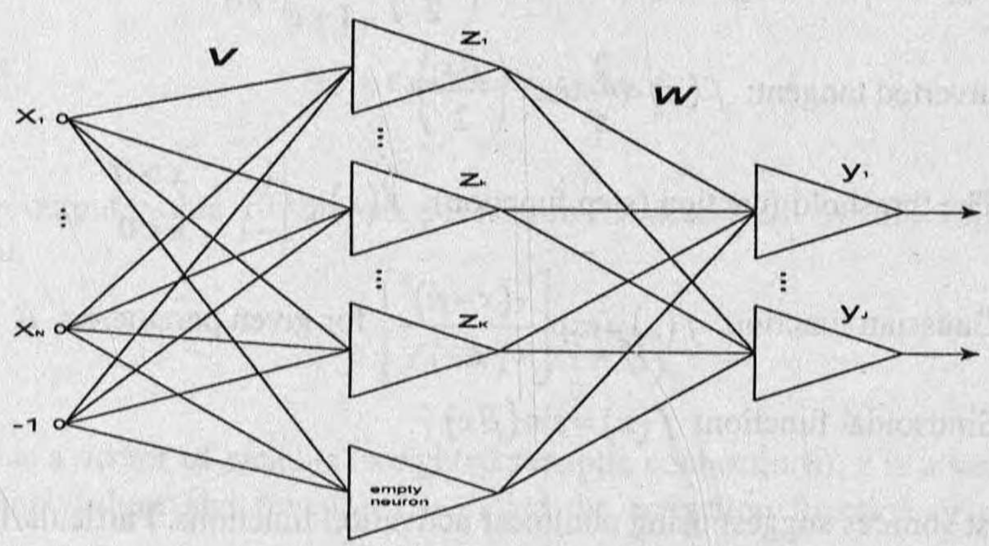

Fig. 3. Graphical representation of a sample multilayer artificial neural network

The first layer (input layer) consists of nodes that simply accept the input values. As it was noticed earlier, the outputs of neurons in the first layer are inputs to neurons in the second layer. The last layer is simply called the output layer. A layer or layers (if there are several layers) between input and output layers are known as hidden layers.

Apart from feedforward networks there are other common neural models i.e. recurrent networks and cellular networks (see Chua, Roska, 2002). As the name suggest, the recurrent network includes feedback in the network structure, i.e. at least one feedback exists. Other recurrent networks proposed by Elman and Hopfield are also known. In these cases signals from an output layer, or respectively a hidden layer, are lagged by one period (cycle) and given again as input to a layer (symmetric neural connections and multiple feedback loops).

The cellular networks, which are also known as Kohonen networks or Self-Organising Maps (SOM) are recognised to be the most complicated and advanced structures among neural networks models (see Kohonen, 1995; Kohonen, Deboeck, 1998). They apply unsupervised learning algorithm particularly in data mining, image processing, visualisation and pattern recognition. As a basic description one can say that high-dimensional data is transformed there into a low-dimensional output space.

3. Designing and learning of feedforward neural networks

As it was noticed earlier multilayer neural networks form the most common type of artificial neural networks. In most cases in forecasting issues they realize goals better than other types of neural nets.

Network designing process is strictly connected with problem formulation. In other words, which and how many variables a decision-maker should and is able to use as inputs (this determine the dimension of input layer) and in what form the answer should be obtained (the number of outputs). Another problem is to determine the number of hidden layers and their neurons. It is believed that artificial neural networks with one hidden layer should be able to solve most research problems. There are no well known problems which could require networks with more than three hidden layers. However, still does not exist a good formal formula describing the number of hidden neurons.

One of the main reasons why neural networks have proven so attractive is that they are, in a sense, capable of "learning". The use of such anthropomorphic language might be considered controversial, but in a mathematical approach to neural networks, "learning" simply means changing the weights of the network in response to some input data. When "successful" or "convergent" learning is possible, there is no need to program the network explicitly to perform a particular task. In other words, one does not need to know in advance how to set the weights. The neural network adjusts its weights according to a learning algorithm, in response to some classified training examples, with the state of the network converging to the "correct" one. In this sense, the neural network "learns from experience".

In any model of supervised learning, it is assumed that there is some "target function" (error function), which is the function to be learned (minimized). The target function is to be thought of as the "correct" function the network should compute. It simplifies matters greatly if one assumes that there is a correct function, and that this function can be computed by the network with some set of weight assignments.

For large networks and long training series, the learning process can be time-consuming. Moreover, it rarely happens that the first networks are built correctly. It is usually effect of cut-and-try method.

The most popular supervised learning method is backpropagation method. The algorithm was processed in 1974 and can be described as an efficient way to calculate the partial derivatives of the network error function with respect to the weights. According to backpropagation rule a weight update from iteration n to $n+1$ may have the following form (see for example White, 1989; Tadeusiewicz, 1993; Żurada, Barski, Jędrych, 1996; Domański, 1998):

$$w_{ji}^{m}(n+1) = w_{ji}^{m}(n) + \Delta w_{ji}^{m}(n) \tag{2}$$

where w_{ji}^{m} are weights between neurone i and j in m:th layer, Δw_{ji}^{m} are weight corrections. A familiar way of determining the search of the weight values is to apply the gradient descent method which is a relatively simple rule. The major drawback though is that learning easily is caught in a local minima. To avoid this hazard some modifications were introduced. According to this rule the correction of single weight should be of the form:

$$\Delta w_{ji}^{m}(m) = \eta \delta_{j}^{m-1} + \alpha \Delta w_{ji}^{m}(n-1) \tag{3}$$

where w_{ji}^{m} are weights between i:th and j:th neurone in m:th layer, are networks errors, η is defined as a learning rate (or step length), α is a momentum parameter and δ_{j}^{m} is value of derivate from error function in m:th layer.

It is worth noticing that error values in each hidden layer are calculated before weight updating in a given layer. The sequence order in updating process is not important. In practice in a given learning process the size of the training set is usually small. In fact the training set is divided into two sets: training and verifying data set.

The crucial fact is to determine the value of learning rate η and momentum parameter α. The learning step parameter has a great influence on stability and quickness of the whole process. If the learning rate parameter is too small then the learning process is really slow. If it is set on high level then the algorithm may be not stable and could not reach the optimal point in the set of whole weights. The value η is usually from set of [0.05; 0.25]. The correct value of momentum parameter, which is set experimentally, make it possible to find the optimum solution.

In other learning approaches i.e. nonsupervised learning rules, there is a vector **x** on the network input. On the basis of the former learning process the network makes some classifications. According to that rule the output vector **y** can represent for example the most typical object in a given class, degree of similarity to average vector in a given class, type of class, etc.

The nonsupervised learning process is usually based on Hebb and Oj algorithms (see Żurada, Barski, Jędrych, 1996).

4. Forecasting stock prices with neural networks

In conventional time series analysis instructions and rules are central. A mathematical formula defines the dynamics. One can pick a model that is assumed to be applicable for the present task, e.g. the well-known Auto Regressive Moving Average (ARMA) model. However neural networks do not perform according to preset rules. When displayed to data the network gains experience, learns from regularities in the past and sets its own rules. Data are not described explicitly in mathematical terms. Neural networks are unique in that sense.

Neural networks have several advantages. The most important one is the ability to learn from data and thus the potential to generalise, i.e. produce an acceptable output for previously unseen input data (important in prediction tasks). This even holds (to a certain extent) when input series contain low-quality or missing data. Another valuable quality is the non-linear nature of a neural network.

The principal motivation for the neural network approach in stock prediction is twofold: (see Vanstone, Finnie, Tan, 2005; Kozdraj, 2005):

- stock data is highly complex and hard to model, therefore a non-linear model is beneficial
- a large set of interacting input series is often required to explain a specific stock, which suits neural networks

It is also possible to approach the prediction task from the angle of economics. Sources suggest the following viewpoint: Each single neuron represents a market participant's decision process. Hence a neural network represents interacting decisions among all participants in the market.

Thus a neural network is a complete description of the financial market in itself. This viewpoint gives an attractive mixture of the mathematical theory of neural networks and economics.

In the cinducted research two networks with different structures were applied to prediction process. Their topology was selected experimentally. This means that other structures were considered e.g. with one hidden layer, larger number of input variables, different number of neurons in layers. Furthermore, other input variables (e.g. difference in maximum and minimum market price, exponential moving averages and others) were considered. However, including these variables did not increase the quality of results.

All networks were created and implemented in Borland Delphi programming environment (Object Pascal).

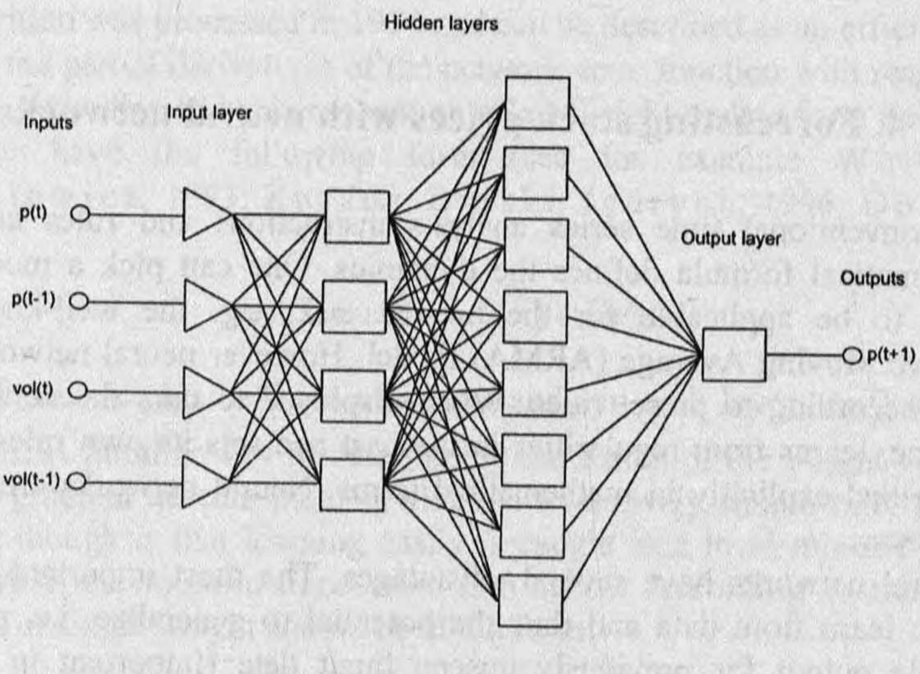

Fig. 4. The scheme of the first forecasting neural network

According to Figure 4 it could be noticed that the first neural network consisted of four layers, and to be more precise, the input layer, two hidden layers and one output layer. The input layer's task is to transform signals, the hidden layers have to make some "inference" through activation and deactivation of appropriate neurons. The last layer has to transform signal into price forecast.

Four variables were found on the network input, that is price in current and previous period and volume in current and previous period. Due to different fluctuation range and to prevent from inappropriate network working, the input variables were normalized to [0, 1] a range according to the following formula:

$$\hat{x} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4}$$

where x is an input value.

Using the original data could lead to incorrect estimation results and applied sigmoid activation function (neuron function) would reach only marginal values. That could result in ineffective learning.

Accordingly to the input values normalization, the output values had to be normalized as follows:

$$y = \hat{x} \cdot (\max(x) - \min(x)) + \min(x) \tag{5}$$

The second neural network consisted of five layers. One input layer, where inputs were selected in the same way as in the first network, three hidden layers with twenty neurons and output layer. In all neurons sigmoid activation function was applied.

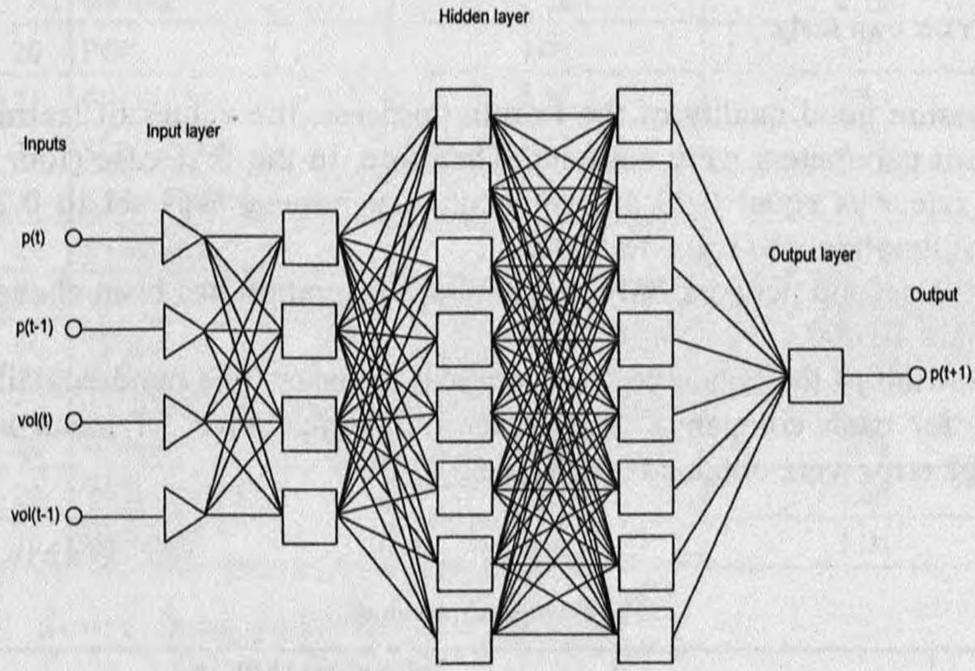

Fig. 5. The scheme of the second forecasting neural network.

Both networks were learned with modified backpropagation algorithm with momentum and learning rate parameters (see equation 3).

To verify the results of forecasts with applied networks, stocks of thirty companies were randomly selected and data fitness (daily forecasts and empirical data) were studied. For each selected stock one hundred different time periods (250 daily quotations) were examined.

Table 1 presents the list of selected companies.

Table 1

List of selected stocks

No.	Company	No.	Company	No.	Company
1	01 NFI	11	KABLE HOLDING	21	PROKOM
2	12 NFI	12	KGHM	22	SOKOŁÓW
3	AGORA	13	KOMPAP	23	STALEXPORT
4	BSK	14	KREDYT BANK	24	STRZELEC
5	BUDIMEX	15	MENNICA	25	TIM
6	COMPUTERLAND	16	MILMET	26	TP S.A.
7	DĘBICA	17	NETIA	27	TU EUROPA
8	GRAJEWO	18	OPTIMUS	28	WARTA
9	IRENA	19	ORBIS	29	WÓLCZANKA
10	JELFA	20	PGF	30	ŻYWIEC

Source: own study.

To assure good quality of the learning process, the values of learning and momentum parameters were constant. Therefore, in the first case (four layers) the step rate was equal 0.90 and momentum parameter was set to 0.21. The number of iteration was equal to 5000.

For the second network only the number of iteration has been changed and was equal to 10 000.

As a result of the conducted research and analyses (one hundred daily price forecasts for each company) the values of average level of mean absolute percentage error were obtained (see Table 2).

Table 2

The average error values

No.	Company	Average MAPE [%]	
		Neural net 1	Neural net 2
1	01 NFI	2.81	3.66
2	12 NFI	4.69	4.71
3	AGORA	2.83	2.87
4	BSK	1.05	1.23
5	BUDIMEX	1.91	4.66
6	COMPUTERLAND	1.36	2.14

Table 2 (contd.)

No.	Company	Average MAPE [%]	
		Neural net 1	Neural net 2
7	DĘBICA	3.13	4.50
8	GRAJEWO	2.96	2.31
9	IRENA	2.57	2.62
10	JELFA	1.82	2.34
11	KABLE HOLDING	4.38	4.74
12	KGHM	2.44	3.82
13	KOMPAP	2.43	3.95
14	KREDYT BANK	1.68	2.23
15	MENNICA	4.19	4.26
16	MILMET	4.20	4.33
17	NETIA	1.50	1.72
18	OPTIMUS	2.25	2.03
19	ORBIS	1.18	2.13
20	PGF	1.65	1.77
21	PROKOM	2.37	2.53
22	SOKOŁÓW	3.60	3.35
23	STALEXPORT	4.17	4.79
24	STRZELEC	2.15	2.33
25	TIM	9.00	11.05
26	TP S.A.	1.93	2.28
27	TU EUROPA	5.58	2.67
28	WARTA	3.38	2.74
29	WÓLCZANKA	4.80	5.53
30	ŻYWIEC	0.99	1.04

S o u r c e: own calculations.

The error values may suggest that forecasts exactitude can be accepted. In a single case there were some disturbances in forecast fitness. The reason for such a behaviour was probably connected directly with empirical data and network parameter setting (number of iteration and coefficient of learning). Still there is no direct rule allowing to get the optimum number of iteration and parameter setting. However, in most cases better results were obtained with the usage of the first network. This can be attributed to less complicated training process (fewer neurons and for that matter fewer weighted connections).

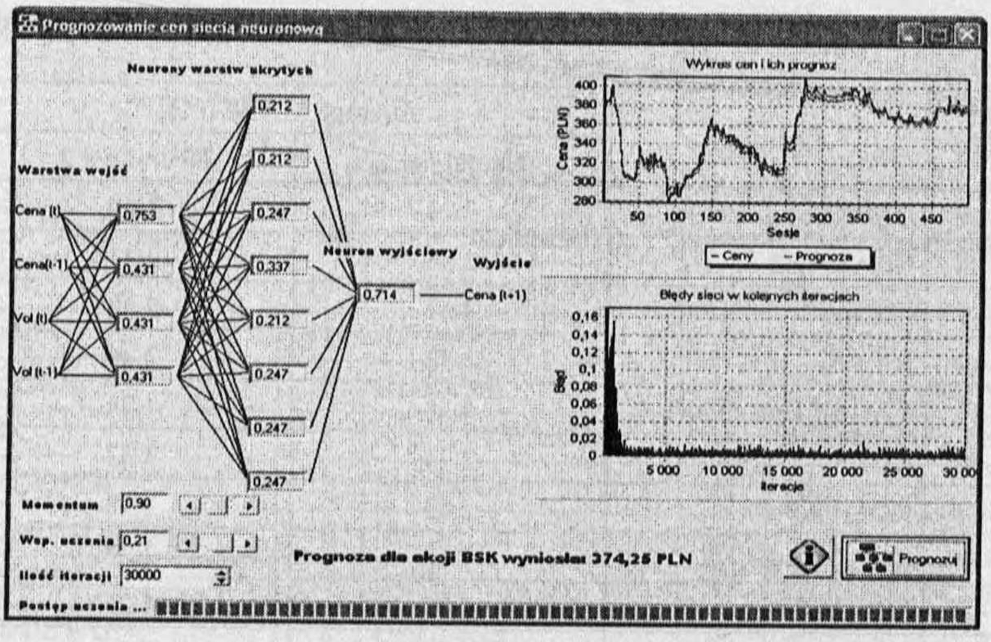

Fig. 6. Sample screenshot of implemented artificial neural network

5. Conclusions

The essence of rational forecasting is to build models of reality and observe their behaviour through a given time period. It is obvious that universal forecasting methods do not exist. As it was noticed artificial neural networks may be very useful when there is no appropriate forecasting model referring to market data. The most important advantages of neural networks lie in the ability of generalization and robustness, mapping of input and output and flexibility (a vast amount of problems can be solved). Another important feature is the fact that no assumptions of the model have to be made.

When considering downsides, the black-box-property first springs to one's mind. Relating one single outcome of a network to a specific internal decision is very difficult. Noisy data also reinforce the negative implications of establishing incorrect causalities, overtraining (or overfitting), which will harm generalisation. Finally, a certain degree of knowledge in current subject is required as it is not trivial to asses the relevance of chosen input series (some proposals are in research conducted by Medeiros and others (see Medeiros, Teräsvirta, Rech, 2005). Furthermore the dependence between network parameter setting and the outcomes quality is still significant, because wrong setting of coefficient of learning and momentum can result in inaccurate predictions.

The field of neural networks is very diverse and opportunities for future research exist in many aspects, including data preprocessing and representation, architecture selection, and application. The future of neural network in stock prices and, generally, times series forecasting seems to be before more complex network types that merge other technologies with neural networks, such as wavelet networks. Nevertheless, a theoretic foundation on which to build is absolutely necessary, as more complex networks still suffer from basic problems such as data preprocessing, architecture selection, and parameterization.

References

Chua L. O., Roska T. (2002), *Cellular neural networks and visual computing*, Cambridge University Press, Cambridge.

Domański Cz. (1998), *Statystyczne systemy ekspertowe*, Wydawnictwo Uniwersytetu Łódzkiego, Łódź.

Hu Y. H., Hwang J. N. (2002), *The handbook of neural network signal processing,* CRC Press.

Kohonen T. (1995), *Self-organising maps*, Springer-Verlag, Berlin.

Kononen T., Doboeck G. (ed.) (1998), *Visual explorations in finance with self-organizing maps*, The SOM Metodology, Springer-Verlag, Berlin.

Kozdraj T. (2004), *Statistical expert systems as a modern decision support tool*, Artificial Intelligence Studies, Siedlce.

Kozdraj T. (2005), *Statystyczne systemy ekspertowe w procesie decyzyjnym na przykładzie rynku kapitałowego* (praca doktorska), Uniwersytet Łódzki.

Medeiros M. C., Teräsvirta T., Rech G. T. (2005), *Building neural network models for time series:A statistical approach*, Working Paper Series in Economics and Finance (508), Stockholm School of Economics.

Neelekanta P. S., De Groff D. (1994), *Neural network modeling: Statistical mechanics and cybernetic prospectives*, CRC Press.

Tadeusiewicz R. (1993), *Sieci neuronowe*, WNT, Warszawa.

Vanstone B. J., Finnie G. R., Tan C. N. W. (2005), *Evaluating the application of neural networks and fundamental analysis in the australian stockmarket*, Computation Intelligence, Calgary.

White H. (1989), *Learning in artificial neural networks: A statistical perspective*, Neural Computation, 425–464.

Żuranda J. M, Barski M., Jędruch W. (1996), *Sztuczne sieci neuronowe*, PWN, Warszawa.

Tomasz Kozdraj

Zastosowanie sztucznych sieci neuronowych do prognozowania cen papierów wartościowych

Sztuczne sieci neuronowe stanowią jedną z najbardziej rozwiniętych gałęzi sztucznej inteligencji. Oparte są na pragmatycznych koncepcjach matematycznych dostosowywanych do rozwiązywanego zadania. Szeroki obszar zastosowań tych struktur obejmuje również zagadnienia szeroko rozumianych inwestycji finansowych. Przyczyn popularności należy upatrywać głównie

w możliwości rozwiązywania skomplikowanych lub niezbyt dobrze rozpoznanych problemów obliczeniowych, sprawności znajdowania rozwiązań oraz możliwości uczenia się na podstawie wzorców lub bez nich. W szczególności sztuczne sieci neuronowe znajdują swoje zastosowanie w problemach predykcji cen papierów wartościowych na rynkach finansowych.

Artykuł przedstawia problematykę zastosowania sieci neuronowych do prognozowania cen akcji na Giełdzie Papierów Wartościowych w Warszawie. Ukazuje ogólną koncepcję sieci neuronowych, ich możliwości, ograniczenia oraz problemy, jakie stają przed badaczem w momencie ich wykorzystania w procesie prognozowania.