

# Designing of Neural Networks for Financial Market Forecasting

Antoshchuk Svitlana  
Sc.D., Professor  
Odesa National  
Polytechnic University,  
Ukraine, Odesa  
asonpnu@gmail.com

Teslenko Pavlo  
Ph.D., Associated  
professor  
Odesa National  
Polytechnic  
University, Ukraine,  
Odesa  
p\_a\_t@ukr.net

Sytnyk Volodymyr  
Ph.D., Associated  
professor  
Ukraine, Odesa  
vladas@ua.fm

Sherstiuk Olha  
Ph.D., Senior Lecturer,  
Odesa National  
Maritime University,  
Ukraine, Odesa  
olusha972@gmail.com

*The paper discusses methods for forecasting financial markets. Their advantages and disadvantages are presented. A genetic approach for the formation of the structure and training of the neural network is proposed. The method of forming a neural network based on a genetic algorithm is given. The effectiveness of the proposed methodology is presented as a result of the task of forecasting the stock market.*

**Keywords:** information system, neural network, financial market forecasting, genetic algorithm.

**Introduction.** The development of effective forecasting information systems is an urgent task for both theory and practice in various fields. In particular, in the field of economics and finance, the need for forecasting can be explained by the high degree of variability in the development of economic systems, which occur under conditions of uncertainty, instability and risk [1]. The background for this is that there are many factors, such as trends of globalization, complication of economic interrelations, growth rates of national markets [2, 3], etc.

In accordance with [4], there are currently more than 100 classes of models. All forecasting methods are divided into two groups: intuitive and formalized. Intuitive forecasting is used when the object of forecasting is either too simple or, on the contrary, so complicated that it is impossible to analytically take into account the influence of external factors. Intuitive forecasting methods do not provide for the development of forecasting models and reflect individual judgments of specialists (experts) about the prospects for the process development. In [5], the application of expert systems is presented, including the use of fuzzy logic.

**Main part.** Formalized forecasting methods [6] use statistical and structural models. In statistical models, the functional relationship between future and actual values of the time series, as well as external factors, is set analytically [7]. The statistical models include the following groups:

- regression models;
- autoregressive models;
- models of exponential smoothing.

In structural models, the functional relationship between future and actual values of the time series, as well as external factors, is set structurally. Structural models include the following groups:

- neural network models;
- models based on Markov chains;
- models based on classification regression trees.

Firm statistical assumptions about the properties of time series restrict the use of mathematical statistics, the theory of random processes to predict financial markets, due to the fact that many real processes are non-linear [8] and have either a chaotic or quasi-periodic or mixed basis.

In this situation, neural networks (NN) can serve as an adequate apparatus for solving problems of diagnostics and forecasting, and, as very promising, radial base structures, characterized by high learning speed and universal approximating capabilities, should be noted.

The aim of the work is to develop a methodology for the formation of neural networks for the financial market analysis.

For forecasting systems based on NN, a heterogeneous network consisting of hidden layers with a non-linear activation function of neural elements and an output linear neuron shows the best quality. The disadvantage of most nonlinear activation functions is that their output values are limited by the  $[0,1]$  or  $[-1,1]$  segment. This leads to the need to scale the data, if they do not belong to the above ranges of values.

Various network learning algorithms and their modifications are used for the network training [5]. The back-propagation error algorithm is of the greatest interest, since it is an effective tool for training in multi-layer forward-propagation neural networks.

Training in the method of back propagation of an error is reduced to the selection of the values of the forward-directed neural network weights, based on the principles of the fastest descent. One of the main drawbacks of this classic algorithm is the possible hit in the local minima of the cost function.

The analysis of multilayer neural networks and their learning algorithms revealed a number of shortcomings and emerging problems [7]:

1. Uncertainty in the choice of the number of layers and the number of neural elements in a layer;
2. Slow convergence of the gradient method with a constant learning step;
3. The difficulty of choosing the appropriate learning rate. Since a small learning rate leads to the NN rolling up to a local minimum, and a high learning rate can lead to the omission of the global minimum and make the learning process divergent;
4. The impossibility of determining the local and global minimum points, since the gradient method does not distinguish them;
5. The effect of random initialization of NN weighting factors on the search for the minimum of the root-mean-square error function.

An alternative to the method of back propagation of errors can serve as genetic algorithms. Genetic algorithms are used to solve optimization problems using the evolution method, i.e. by selecting the most appropriate one from a variety of solutions. They differ from traditional optimization methods in the following properties [9]:

1. Process not the parameter values of the problem, but their coded form.
2. Search for solutions based on a certain population.
3. Use only the objective function, and not its derivative.
4. Algorithms are stochastic.

The purpose of the training is to minimize the cost function  $E(n)=(e_k^2(n))/2$ , where  $e_k(n) = d_k(n) - y_k(n)$  is the error,  $d_k(n)$  is the desired output of the neural network,  $y_k(n)$  is the real output of the neural network,  $n$  is the iteration number.

The parameters of the problem are the weights that determine the point of the search space and, therefore, represent a possible solution.

If the weights take real values from the interval  $[-1,1]$ , then each chromosome will be a combination of 9 binary sequences (genotypes) encoding specific weights. The corresponding phenotypes are represented by the sets of the corresponding real numbers from the interval  $[-1,1]$ . The length of chromosomes depends on the problem situation.

If it is required to find a solution with accuracy up to  $q=2$  significant decimal figures for each weight, then the interval  $[a, b]$  should be divided into  $(b - a) \cdot 10^q$  identical subintervals. This means applying discretization in increments of  $r=10^{-q}$ . The smallest positive integer  $m_i$  satisfying the inequality  $(b - a) \cdot 10^q \leq 2^{m_i} - 1$  determines the necessary and sufficient length of the binary sequence required to encode a number from an interval  $[a, b]$  in increments of  $r$ . As a result, the length of the binary coding sequence is 8 bits.

When coding real numbers, an integer is taken as the value of the gene that determines the number of the interval (The Gray code is used). As the phenotype value, the number that is the middle of this interval is taken.

The initial chromosome population is randomly assigned. When calculating the neural network output values, you can use the logistic activation function with the learning speed parameter  $\eta=1$ .

The stage of selection of parental chromosomes for creating a new population performs the greatest role in the successful functioning of the algorithm. The most effective is the tournament method. The essence of the method is as follows. All individuals of the population are divided into subgroups of 2-3 individuals each. The choice of parents is made randomly with a probability less than 1.

In the classical genetic algorithm, two main genetic operators are used: the crossover operator and the mutation operator. The probability of crossing is large enough (usually  $0,5 \leq p_c \leq 1$ ), the probability of mutation is established very small (most often  $0 \leq p_m \leq 0,1$ ). This means that interbreeding in the classical algorithm is almost always performed, while mutation is quite rare.

The crossover operator acts as follows:

- from the population with probability  $p_c$  two individuals are selected, which are included in the composition of the temporary parent population;
- the point of crossing  $l_k$  is determined (also randomly);
- the concatenation of the part of the first and second parents.

The mutation operator with probability  $p_m$  changes the value of a gene in the chromosome to the opposite. The probability of mutation can be calculated by randomly choosing a number from the interval  $[0, 1]$  for each gene and selecting for this operation those genes for which the played number is less than or equal to the value  $p_m$ .

The chromosomes obtained by applying genetic operators to the chromosomes of the temporary parent population are included in the new population. It becomes the current population for this iteration of the genetic algorithm. At each iteration the value of the fitness function for all chromosomes of this population is calculated, after which the condition of the algorithm stop is checked. As such a condition, either a restriction on the maximum number of epochs of the algorithm functioning is applied, or the algorithm convergence is determined by comparing the population's fitness function values at several epochs. When this parameter is stabilized, the algorithm stops.

The genetic algorithm was used in the study of neural network structures to forecast the stock prices of public JSC (joint-stock company) “Lukoil”.

The studies were conducted for the 2017-year time series. Fig. 1, 2 shows the results of the search for optimal neural networks. Studies were conducted for MLP type networks (three and four-layer network). It is known that the search for the type of neural network and structure is a rather time-consuming procedure, so the intermediate task of determining the initial “prototype” was solved, after which it was determined that the further structure of the NN was further refined.

The results of calculations showed that for the stock price of JSC “Lukoil” the optimal structures are neural networks with the following indicators: type –MLP; three-layer structure: 3 input neurons – 8 neurons of the hidden layer – 1 output neuron (Fig. 2), or MLP; three-layer structure: 1 input neuron – 11 neurons of the hidden layer – 1 output neuron (Fig. 1).

**Conclusions.** The neural network structures obtained were used to forecast the value of LUKOIL stocks (Fig. 3, 4).

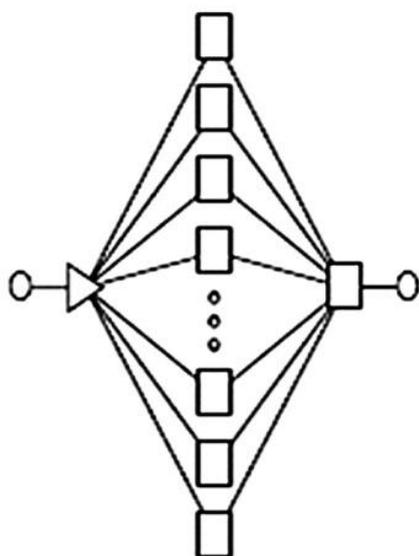


Fig. 1. MLP 1:11:1

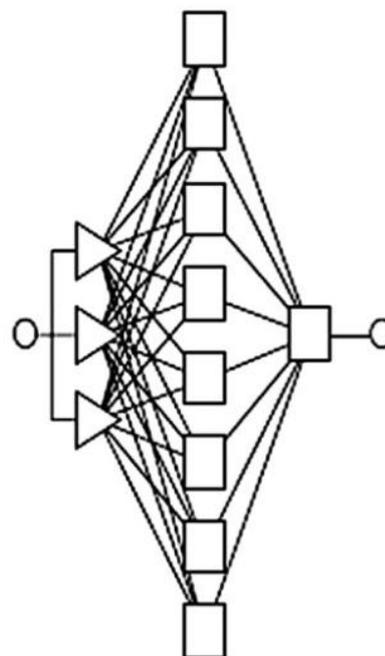


Fig. 2. MLP 3:8:1

The forecasting accuracy was 73% for LUKOIL stocks, which is a high result and exceeds the forecasting results by other methods.

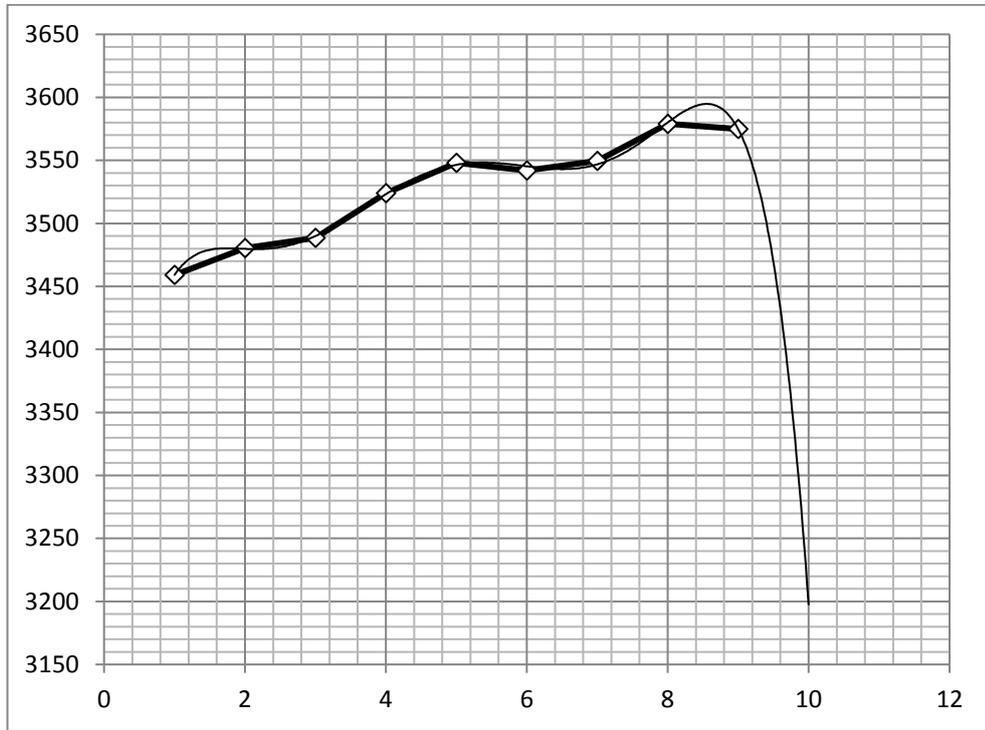


Fig. 3

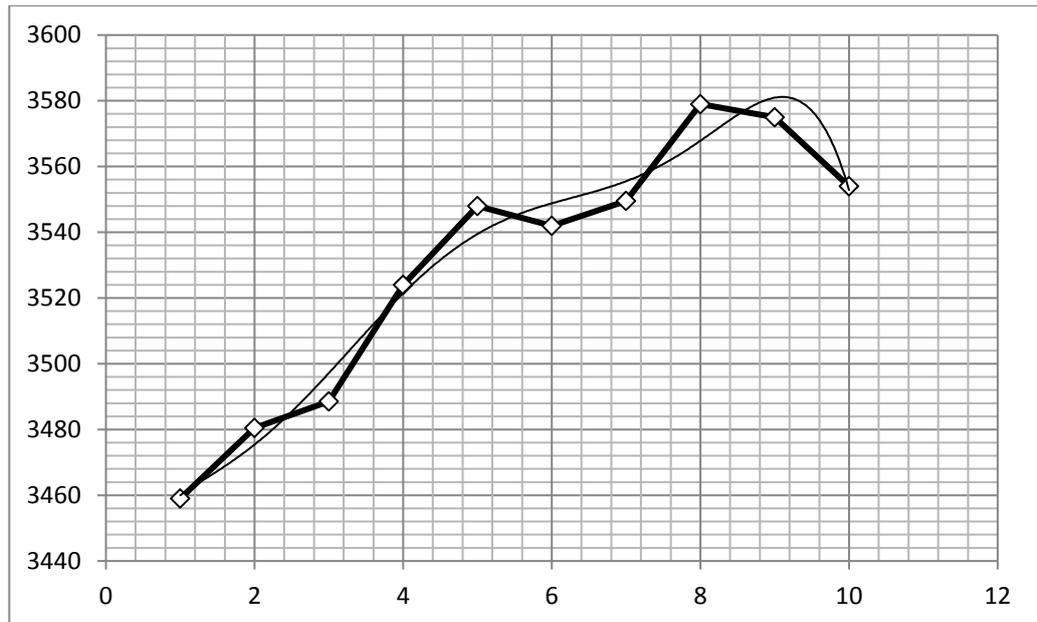


Fig. 4

In the future, the forecast results can be used in the management of project-oriented organizations [10], both in the management of individual portfolio projects and in the strategic management of the organization.

## References

1. Anatoliev A.A., Teslenko P.A., Chimshir V.I. (2015). Project-oriented orientation of the management processes of investment companies in the foreign exchange market. Bulletin of the National Technical University "KhPI": №1 (1110). pp. 80 – 84.
2. Kunwar Singh Vaisla, Ashutosh Kumar Bhatt. (2010). An Analysis of the Performance of Artificial Neural Network Technique for Stock Market Forecasting. International Journal on Computer Science and Engineering Vol. 02, No. 06, PP. 2104-2109
3. Niaki, S.T.A. & Hoseinzade, S. J Ind Eng Int (2013) 9: 1. Forecasting S&P 500 index using artificial neural networks and design of experiments. <https://doi.org/10.1186/2251-712X-9-1>
4. Tikhonov, E.E. (2006). Forecasting in market conditions. Nevinnomyssk.
5. Rutkovskaya, D., Pilinsky, M. & Rutkovsky, L. (2004). Neural networks, genetic algorithms and fuzzy systems. Moscow.
6. Sytnyk, V., & Georgalina, O. (2018). Theoretical and practical aspects of the development of modern science: the experience of countries of Europe and prospects for Ukraine: "Baltija Publishing", 524 p. DOI: [dx.doi.org/10.30525/978-9934-571-30-5](https://doi.org/10.30525/978-9934-571-30-5).
7. Tymchenko B., Antoshchuk S. (2019) Race from Pixels: Evolving Neural Network Controller for Vision-Based Car Driving. ICDSIAI 2018. Advances in Intelligent Systems and Computing, vol 836. Springer, Cham
8. Fernandez-Rodriguez, F., Sosvilla-Rivero, S. & Andrada-Felix, J. (2002). Nearest-Neighbour Predictions in Foreign Exchange Markets. Fundacion de Estudios de Economia Aplicada, 5, 36 p.
9. Singh, S. (2000). Pattern Modelling in Time-Series Forecasting. Cybernetics and Systems-AnInternational Journal, 31.(1), PP. 49 - 65.
10. Teslenko P., Polshakov I., Bedrii D. (2016) Strategic management of evolving project-oriented organization. Science and Education a New Dimension, Economics, IV (2), Issue: 94, Budapest. PP. 33-35. available at: [http://www.seanewdim.com/uploads/3/4/5/1/34511564/econ\\_iv\\_2\\_\\_94.pdf](http://www.seanewdim.com/uploads/3/4/5/1/34511564/econ_iv_2__94.pdf)