THE NEO-FUZZY NEURAL NETWORK STRUCTURE OPTIMIZATION USING THE GMDH FOR THE SOLVING FORECASTING AND CLASSIFICATION PROBLEMS

Bodyanskiy Ye.[1], Zaychenko Yu.[2], Pavlikovskaya E.[2], Samarina M.[2], Viktorov Ye.[1]

[1]*Control System Research Laboratory, Kharkiv National University of Radio Electronics, Lenin av., 14, 61166, Kharkiv, Ukraine*
[2]*Institute For Applied System, Kyiv Polytechnic Institute, Peremoga av., 37, 03056, Kyiv, Ukraine*

bodya@kture.kharkov.ua, baskervil@voliacable.com, elena.pavlikovskaya@gmail.com, aniramas@gmail.com, yevgen.viktorov@gmail.com

Abstract. *The problem of Neo-Fuzzy Neural Network structure optimization is considered in this paper. For its solution Group Method of Data Handling (GMDH) is suggested and the algorithm of structure optimization is described.. The experimental investigations were carried out and their results accuracy of forecasting by optimally constructed Neo-Fuzzy Neural Network and network with multilayer feedforward architecture are presented and compared. Also a classification problem was solved and results are given in this paper to show that proposed self-organized architecture is capable to perform classification as well as forecasting.*

Keywords

Artificial neural networks, neo-fuzzy neuron, group method of data handling, structure optimization, stock prices forecasting.

1 Introduction

Last years the problem of stock prices and market indexes forecasting is of great importance. For its solution various approaches were applied. The most prospective methods of forecasting at markets are neural networks, especially a fuzzy neural networks and the GMDH. Earlier it was proved that neural networks are universal approximators and have some remarkable properties, such as parallel processing of information, ability to work with incomplete noisy input data, and learning possibilities to achieve the desired response (output).

The GMDH, from the other side, uses the principle of self-organization that allows to construct an optimal structure of the forecasting model during the algorithm operation. It’s very promising to combine advantages of these both approaches for the solution of the problem – constructing an efficient model for the financial markets forecasting.

The goal of the present work is a synthesis of the Neo-Fuzzy Neural Network using the GMDH and its application for financial processes forecasting at stock markets. Experimental investigations of the efficiency of the proposed approach and its comparison with application of Neo-Fuzzy Neural Network with constant architecture are also the goal of this paper.

2 The neo-fuzzy neuron

The architecture of the neo-fuzzy neuron (NFN) was proposed by Takeshi Yamakawa and co-authors in [1-3]. The authors of the NFN admit among its most important advantages, the high rate of learning, computational simplicity, the possibility of finding the global minimum of the learning criterion in real time and also that it is characterized by fuzzy linguistic “if-then” rules. The neo-fuzzy neuron is a nonlinear multi-input single-output system shown in Fig.1.

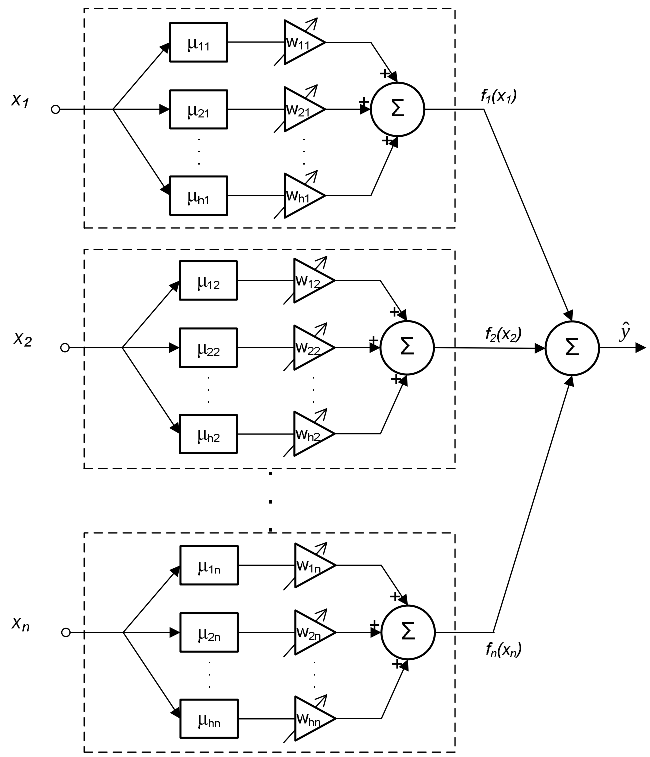


Fig. 1. The neo-fuzzy neuron.

It realizes the following mapping:

$$\hat{y} = \sum_{i=1}^{n} f_i(x_i), \tag{1}$$

where x_i is the *i*-th input (*i* = 1,2,…,*n*), $\hat{y}$ is a system output. Structural blocks of neo-fuzzy neuron are nonlinear synapses NS_i which perform transformation of *i*-th input signal in the from

$$f_i(x_i) = \sum_{j=1}^{h} w_{ji}\mu_{ji}(x_i)$$

and realize fuzzy inference

$$\text{IF } x_i \text{ IS } x_{ji} \text{ THEN THE OUTPUT IS } w_{ji}$$

where x_{ji} is a fuzzy set which membership function is μ_{ji}, w_{ji} is a singleton (synaptic weight) in consequent [2]. As it can be readily seen nonlinear synapse in fact realizes Takagi-Sugeno fuzzy inference of zero order.

Conventionally the membership functions $\mu_{ji}(x_i)$ in the antecedent are complementary triangular functions as shown in Fig. 2.

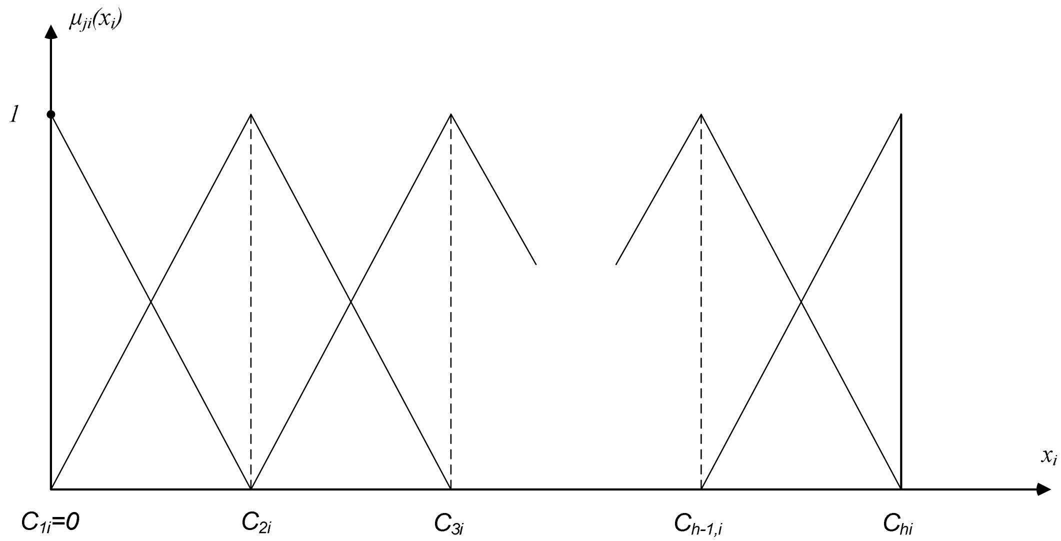


Fig. 2. The triangular membership functions.

For preliminary normalized input variables x_i (usually $0 \le x_i \le 1$), membership functions can be expressed in the form:

$$\mu_{ji}(x_i) = \begin{cases} \dfrac{x_i - c_{j-1,i}}{c_{ji} - c_{j-1,i}}, x \in [c_{j-1,i}, c_{ji}], \\ \dfrac{c_{j+1,i} - x_i}{c_{j+1,i} - c_{ji}}, x \in [c_{ji}, c_{j+1,i}], \\ 0 - otherwise, \end{cases}$$

where c_{ji} are arbitrarily selected centers of corresponding membership functions. Usually they are equally spaced on interval [0, 1]. This contributes to simplify the fuzzy inference process. That is, an input signal x_i activates only two neighboring membership functions simultaneously and the sum of the grades of these two membership functions equals to unity (Ruspini partitioning), i.e.

$$\mu_{ji}(x_i) + \mu_{j+1,i}(x_i) = 1. \quad (2)$$

Thus, the fuzzy inference result produced by the Center-of-Gravity defuzzification method can be given in the very simple form

$$f_i(x_i) = w_{ji}\mu_{ji}(x_i) + w_{j+1,i}\mu_{j+1,i}(x_i).$$

By summing up $f_i(x_i)$, the output $\hat{y}$ of Eq. (1) is produced.

It should be noticed that triangular activation functions provide only piecewise-linear approximation and this fact can in most of the cases can lead to decreasing of the received results accuracy. To minimize its negative effect we can increase number of membership functions. But it results in increasing of synaptic weight coefficients quantity and therefore complexity of our architecture is rising as well as time required for its learning.

To avoid this disadvantage we propose to use the cubic-spline membership functions (3) that can be written down in the following form:

$$\mu(x)=\begin{cases}0.25\left(2+3\dfrac{2x-x_i-x_{i-1}}{x_i-x_{i-1}}-\left(\dfrac{2x-x_i-x_{i-1}}{x_i-x_{i-1}}\right)^3\right), & x\in[x_{i-1},x_i],\\ 0.25\left(2-3\dfrac{2x-x_{i+1}-x_i}{x_{i+1}-x_i}+\left(\dfrac{2x-x_{i+1}-x_i}{x_{i+1}-x_i}\right)^3\right), & x\in(x_i,x_{i+1}]\end{cases} \quad (3)$$

and shown in the Fig. 3.

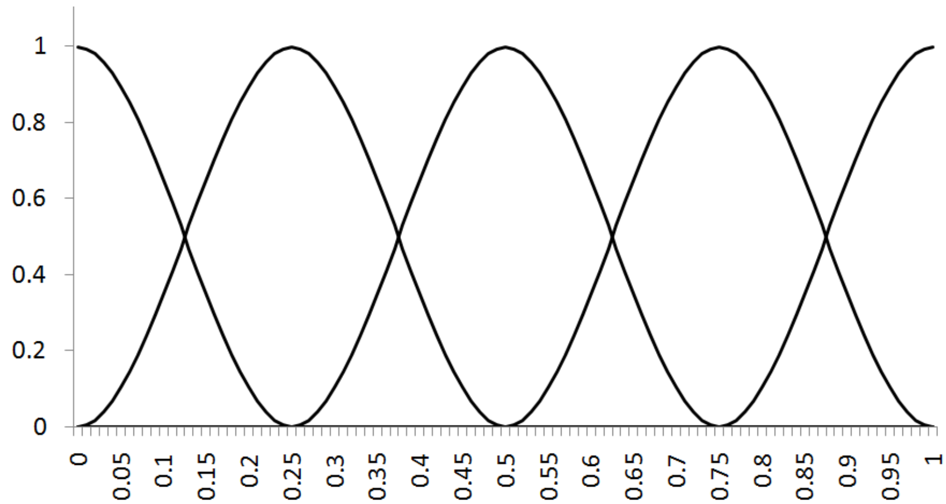


Fig. 3. Cubic-spline activation functions.

The cubic-spline activation functions (3) satisfy all requirements of the Ruspini partitioning (2) and it is considerably contributes to simplify the fuzzy inference process. On the other hand, usage of the cubic spline activation functions provides smooth polynomial approximation instead of piecewise-linear approximation and makes possible to perform a high quality modeling of significantly nonlinear nonstationary signals and processes.

When a vector signal $x(k)=(x_1(k),x_2(k),...,x_n(k))^T$ (here $k=1,2,...$ is a discrete time) is fed to the input of the neo-fuzzy neuron, the output of this neuron is determined by both the membership functions $\mu_{ji}(x_i(k))$ and tunable synaptic weights $w_{ji}(k-1)$, which have been obtained at the previous training epoch:

$$\hat{y}(k)=\sum_{i=1}^{n} f_i(x_i(k))=\sum_{i=1}^{n}\sum_{j=1}^{h} w_{ji}(k-1)\mu_{ji}(x_i(k))$$

and thereby neo-fuzzy neuron contains *h*n* synaptic weights which should be determined.

3 The neo-fuzzy neuron learning algorithm

The learning criterion (goal function) is the standard local quadratic error function:

$$E(k)=\frac{1}{2}(y(k)-\hat{y}(k))^2=\frac{1}{2}e(k)^2=\frac{1}{2}\left(y(k)-\sum_{i=1}^{n}\sum_{j=1}^{h} w_{ji}\mu_{ji}(x_i(k))\right)^2$$

It is minimized via the conventional gradient stepwise algorithm. And as a result the following weight update procedure is obtained:

$$w_{ji}(k+1)=w_{ji}(k)+\eta e(k+1)\mu_{ji}(x_i(k+1))=w_{ji}(k)+\eta\left(y(k+1)-\sum_{i=1}^{n}\sum_{j=1}^{h} w_{ji}(k)\mu_{ji}(x_i(k+1))\right)\mu_{ji}(x_i(k+1)),$$

where $y(k)$ is the target value of the output, η is the scalar learning rate parameter which determines the speed of convergence and is chosen empirically.

For the purpose of increasing training speed [4, 5] Kaczmarz-Widrow-Hoff optimal one-step algorithm [6-8] can be used in the following form

$$w(k+1) = w(k) + \frac{y(k+1) - w^T(k)\mu(x(k+1))}{\|\mu(x(k+1))\|^2}\mu(x(k+1)),$$

where $\mu(x(k+1)) = (\mu_{11}(x_1(k+1)),...,\mu_{h1}(x_1(k+1)),...,\mu_{h2}(x_2(k+1)),...,\mu_{ji}(x_i(k+1)),...,\mu_{hn}(x_n(k+1)))^T$,

$w(k) = (w_{11}(k),...,w_{h1}(k),...,w_{h2}(k),...,w_{ji}(k),...,w_{hn}(k))^T$ - $(hn)\times 1$-vectors, generated by the corresponding variables, and its exponentially weighted modification

$$\begin{cases} w(k+1) = w(k) + r^{-1}(k+1)\left(y(k+1) - w^T(k)\mu(x(k+1))\right)\mu(x(k+1)), \\ r(k+1) = \alpha r(k) + \|\mu(x(k+1))\|^2, 0 \le \alpha \le 1, \end{cases}$$

which possesses both smoothing and filtering properties.

In case we have priori defined data set training process can be performed in a batch mode for one epoch using conventional least squares estimation. The neo-fuzzy neuron can be used as an elementary node of the architecture called the Neo-Fuzzy Neural Network.

4 The Neo-Fuzzy Neural Network and its architecture optimization using the Group Method of Data Handling.

The Neo-Fuzzy Neural Network is a multilayer feedforward architecture that consists of neo-fuzzy neurons. 3-layers Neo-Fuzzy Neural Network [9] with *n* inputs and *m* outputs is shown of Fig. 4.

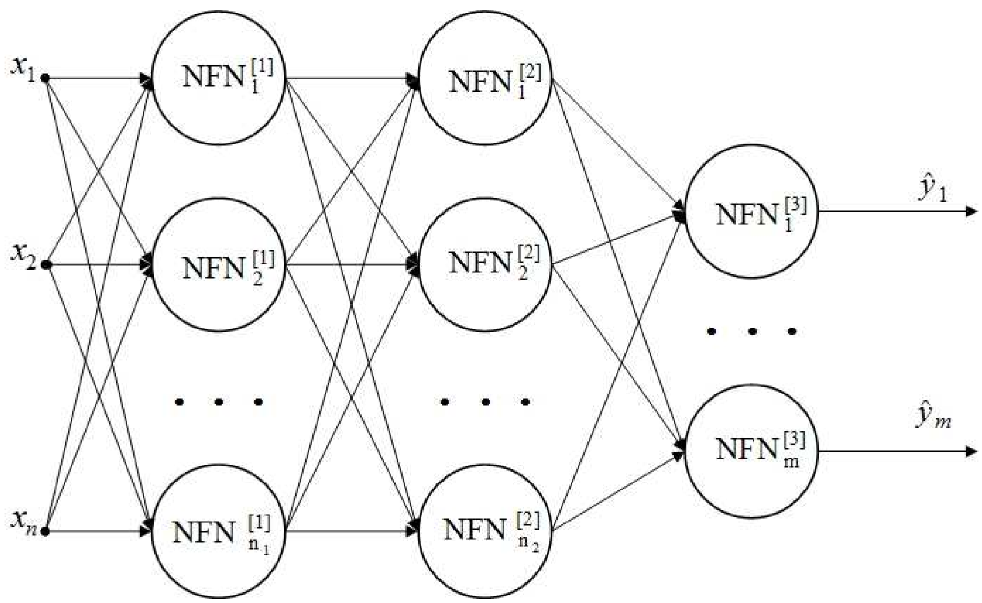


Fig. 4. The Neo-Fuzzy Neural Network.

Given architecture is completely coincide with the structure of the 3-layer perceptron, except that the neo-fuzzy neurons are used here as an elementary nodes instead of Rosenblatt perceptrons. Therefore, for the adjustment of the weight coefficients of such architecture it is necessary to use backpropagation algorithms. As it generally known, such algorithms are quite complex from the computational point of view and they operate slowly.

If we use neo-fuzzy neurons that have only two inputs, the GMDH can be applied for the synthesis of the Neo-Fuzzy Neural Network with optimal architecture.

The main idea of the GMDH algorithm lay in successive synthesis of the neuron layers until the external criterion begins to increase. Algorithm description [10-14]:

1) Form pairs from the neo-fuzzy neuron outputs of the current layer (at the first iteration we use the set of input signals). Each pair is fed to the corresponding neo-fuzzy neuron.

2) Using the learning subsample adjust synaptic weight coefficient of each neo-fuzzy neuron.

3) Using the test subsample calculate the value of the external criterion (regularity) for each neo-fuzzy neuron:

$$\varepsilon_p^{[s]} = \frac{1}{N_{nep}} \sum_{i=1}^{N_{nep}} \left(y(i) - \hat{y}_p^{[s]}(i)\right)^2 \tag{4}$$

where N_{nep} is a size of the test subsample, s is the layer number, p is a neuron number in the current layer $p = \overline{1, n_s}$, $\hat{y}_p^{[s]}(i)$ is the p-th neuron of the s-th layer response signal for the i-th input vector.

4) Find the minimal value of the external criteria for all neo-fuzzy neurons of the current layer

$$\varepsilon^{[s]} = \min_p \varepsilon_p^{[s]}.$$

Check the condition

$$\varepsilon^{[s]} > \varepsilon^{[s-1]} \tag{5}$$

where $\varepsilon^{[s]}, \varepsilon^{[s-1]}$ are the criterion values for the best neurons of the and s-th and (s-1)-th layers correspondingly. If the condition (5) is true then return to the previous layer and find the best neuron that has minimal value of the criterion (4). Otherwise, select F best neurons according to the criterion (4) value and go to the step 1 to construct the next layer of neurons.

5) Determine the final structure of the network. Moving backward from the best neuron of the (m-1)-th layer along the input connections and passing successively all the layers of neurons, preserve in the final structure only such neurons that are used in the next layer.

After the GMDH finishes its functioning it can be said that the final optimal structure of the Neo-Fuzzy Neural Network is synthesized. As it can be readily seen we obtain not only optimal structure, but also trained neural network that is ready to process new data. One of the most important advantages of GMDH usage for the Neo-Fuzzy architecture synthesis is a capability to use simple but very quick learning procedures for the neo-fuzzy neuron weights adjustment because network is trained layer-by-layer.

5 The experimental investigations of forecasting with neo-fuzzy neural network

The experimental investigations of neo-fuzzy neural network in the problem of forecasting were carried out. The goal contained in RTS index forecasting on the base of current stock prices of the leading Russian companies.

Input data: daily stock prices and the value of RTS index in the period from 5 of February till 5 of May 2009.

The output is RTS index on the next day.

Sample size was 100 values.

Forecast criteria were the following:

1. mean squared error (MSE);
2. mean absolute percentage error (MAPE).

Types of experiments for Neo-fuzzy neural network:

1) Variation of ratio learning/ test samples in the range: 25:75, 50:50, 75:25;
2) Change the number of layers: 1-3-5;

3) Change the number of iterations: 1000, 10000, 100000;
4) Variation of a number of points to be forecasted: 1-3-5;
5) Change of maximal error – the condition of stop: 0.01 та 0.09;

Some of the obtained experimental results are presented below.

Experiment type 1. Variation of ratio learning / test sample

Experiment A) Ratio 75: 25 Flow chart of real and forecasted values are presented on Fig .5.

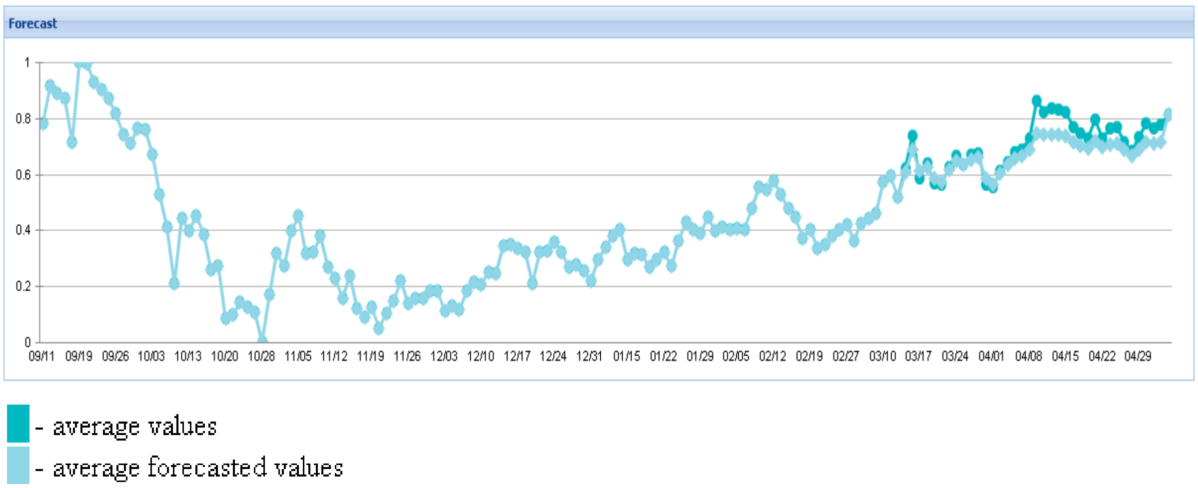


Fig.5. Forecasted results under ratio learning / test sample 75:25

MSE= 0.050158

Experiment B) ratio 50:50 MSE= 0.053562

Experiment C) ratio learning/test – 25:75. The results are presented on Fig.6.

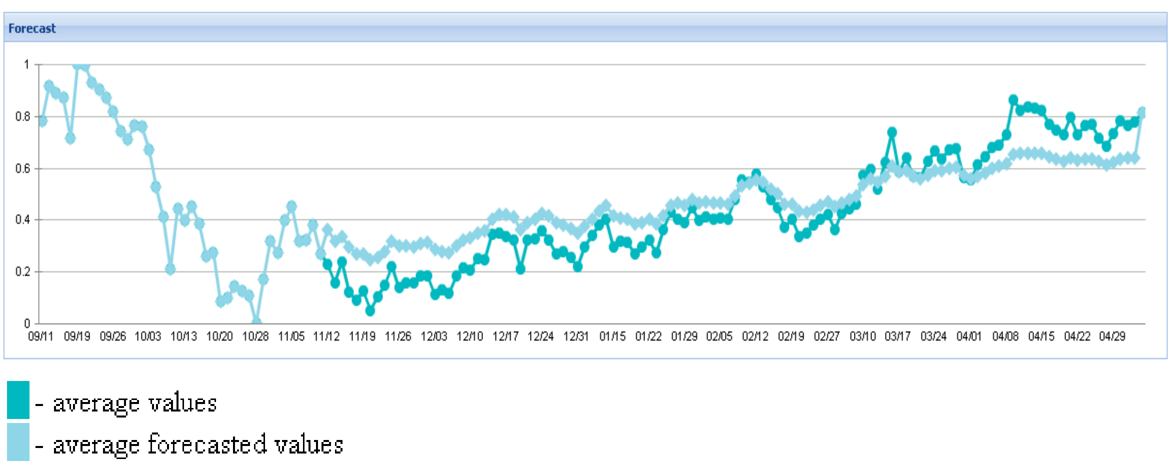


Fig.6. Forecasted results under ratio learning / test sample 25:75

MSE=0.068489

Experiment type 2. Variation of layers number

Comparison of algorithm work when number of layers is varied: 1-3-5-7 while forecast at 1 point under ratio learning / test sample 75:25

Experiment A) layers number – 1 MSE = 0.04662. The results are presented on Fig. 7.

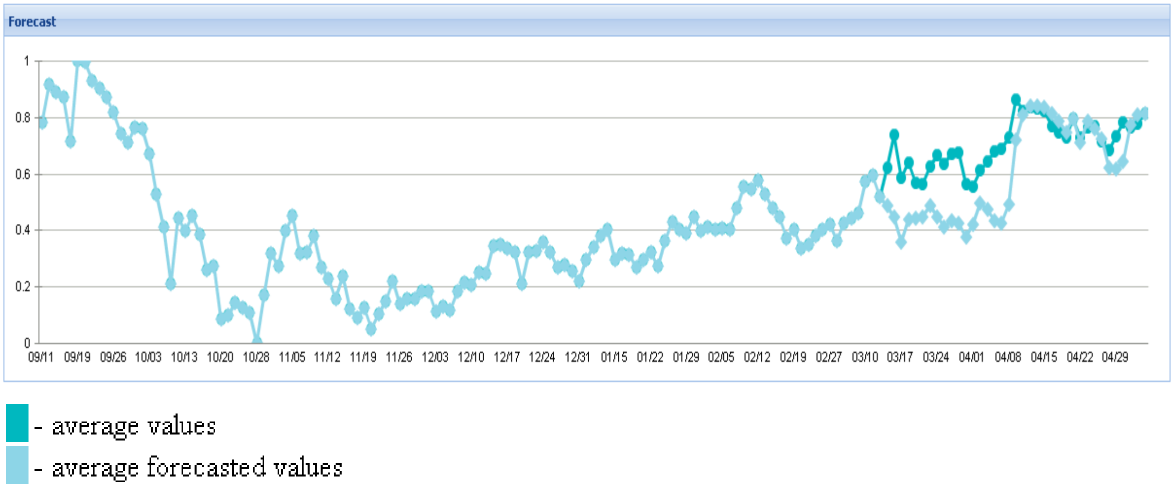


Fig. 7. Forecasted results with one layer

*Experiment B***)** layers number- 3, MSE=0.255

Experiment C) layers number- 5, MSE=0.0446

Experiment D) layers number- 7, MSE=0.0544

Experiments type 3. Variation of iterations number: 1000, 10000, 100000

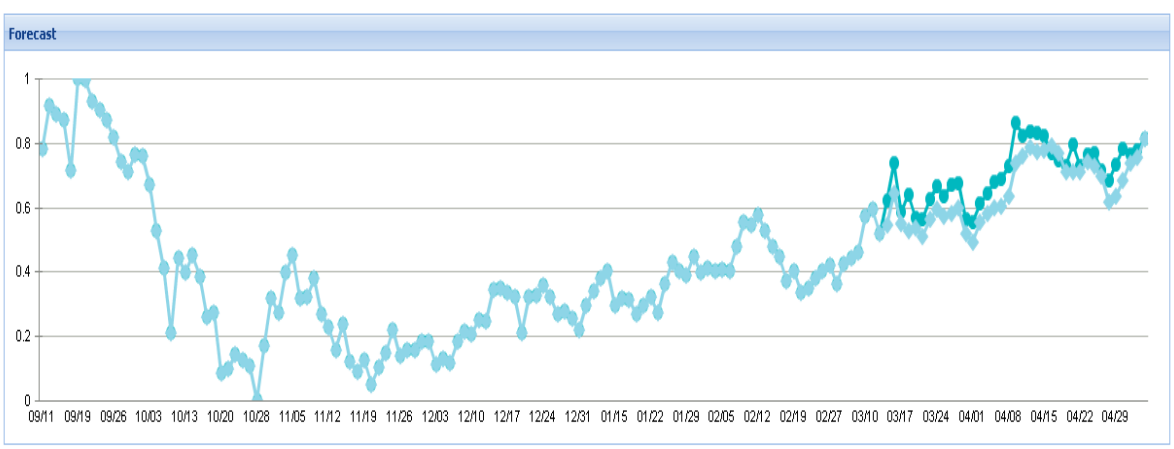
Experiment B) iterations number – 10000, MSE= 0.0575

Experiment C) iterations number – 100000, MSE=0.0525

Experiments type 4. Variation of number of forecasted points

Comparison of algorithm forecasting accuracy when varying a number of forecasted points 1-3-5, using ratio learning / test sample 75:25

Experiment A) a number of forecasted points – 1 (Fig.8).



- average values
- average forecasted values

Fig.8. Forecasted results with one forecasted point

MSE= 0.0495

Experiment B) a number of forecasted points – 3, MSE=0.4469

Experiment C) a number of forecasted points – 5, MSE= 1.0418

Tab 1. Summary table of real and forecasted results

date	average values	average forecasted	average deviation	average quadratic deviat	standard deviation	relative deviation
04/10	0.8234217749313815	0.75902057	0.06440121	0.0041475156	0.07821169	7.82%
04/13	0.833028362305581	0.785795	0.047233403	0.0022309944	0.056700833	5.67%
04/14	0.8301006404391582	0.77435535	0.055745304	0.003107539	0.06715487	6.71%
04/15	0.8206770356816102	0.77505255	0.045624495	0.0020815944	0.055593725	5.55%
04/16	0.768892955169259	0.7893609	0.020467937	0.00041893646	0.026620008	2.66%
04/17	0.7446477584629461	0.76939434	0.024746597	0.00061239407	0.03323262	3.32%
04/20	0.7282708142726441	0.7093161	0.018954754	0.00035928268	0.026027067	2.60%
04/21	0.7950594693504117	0.71122634	0.0838331	0.0070279883	0.105442554	10.5%
04/22	0.7262580054894785	0.70938486	0.016873121	0.00028470223	0.023232957	2.32%
04/23	0.7618481244281794	0.7413372	0.020510972	0.00042069994	0.02692265	2.69%
04/24	0.7659652333028363	0.72680837	0.039156854	0.0015332592	0.05112093	5.11%
04/27	0.7136322049405307	0.6966812	0.016951025	0.00028733723	0.023753166	2.37%
04/28	0.6817017383348581	0.6155766	0.066125095	0.0043725283	0.09700004	9.70%
04/29	0.731747483989021	0.6322194	0.099528134	0.00990585	0.13601431	13.6%
04/30	0.7817017383348582	0.6843565	0.09734523	0.0094760945	0.12452989	12.4%

Conclusions on experimental results

After having carried out the series of experiments with neo-fuzzy neural network of full structure and of optimal structure constructed by GMDH the following results were obtained which are presented in the table 2:

Tab.2. Comparison of the Neo-Fuzzy Neural Network with full structure and structure constructed by the GMDH.

Experiment series	Experiment type	Network constructed by the GMDH	Full network
Variation of ratio learning / test sample	75% : 25%	0.0484	0.0501
	50% : 50%	0.0532	0.0536
	25% :75%	0.0608	0.0684
Number of layers	1	0.0628	0.0626
	3	0.0381	0.0544
	5	0.0434	0.0652
Number of iterations	1000	0.0588	0.0674
	10000	0.0479	0.0485
	100000	0.0459	0.0482

Number of points to be forecasted	1	0.0495	0.0587
	3	0.4469	1.0844
	5	1.0418	1.3901
Value of maximal error (stop threshold)	0.01	0.0507	0.0881
	0.09	0.0899	0.0898

The best results are highlighted with the grey color. As it can be readily seen the Neo-Fuzzy Neural Network with optimal structure constructed by GMDH gives better results than the conventional network with full structure (full network).

This may be explained by the utilization of self-organization mechanism for constructing not full network. But at the same time there are some disadvantages of this approach – the rate of convergence is slower in comparison with full network. But taking into account the better criterion values this disadvantage may be neglected.

For better estimation of the suggested approach the forecasting error obtained at the experiments is presented on the Fig. 9 and Fig. 10. These are the charts of MAPE obtained by Neo-fuzzy neural network constructed by GMDH.

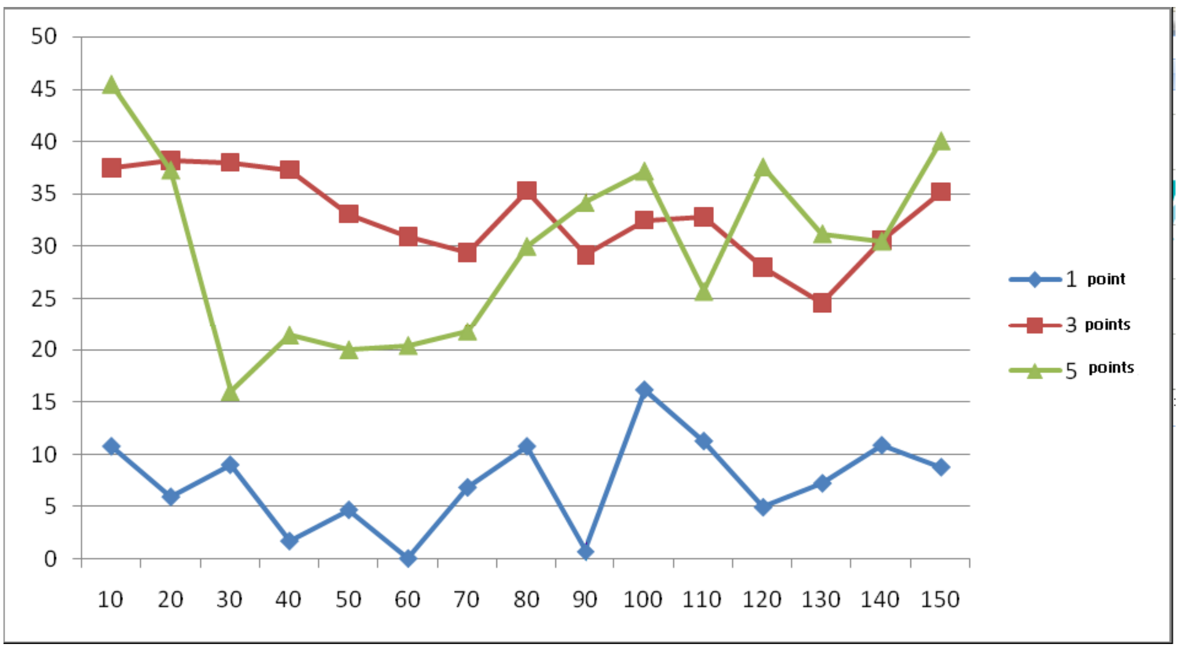


Fig. 9. The curves of error while forecasting 1,3 or 5 points

As we may see while forecasting 1 point ahead we obtain rather high precision –less than 15%. In case of increase the number of points forecasted the accuracy drops- the error lies in the range 15-45%.

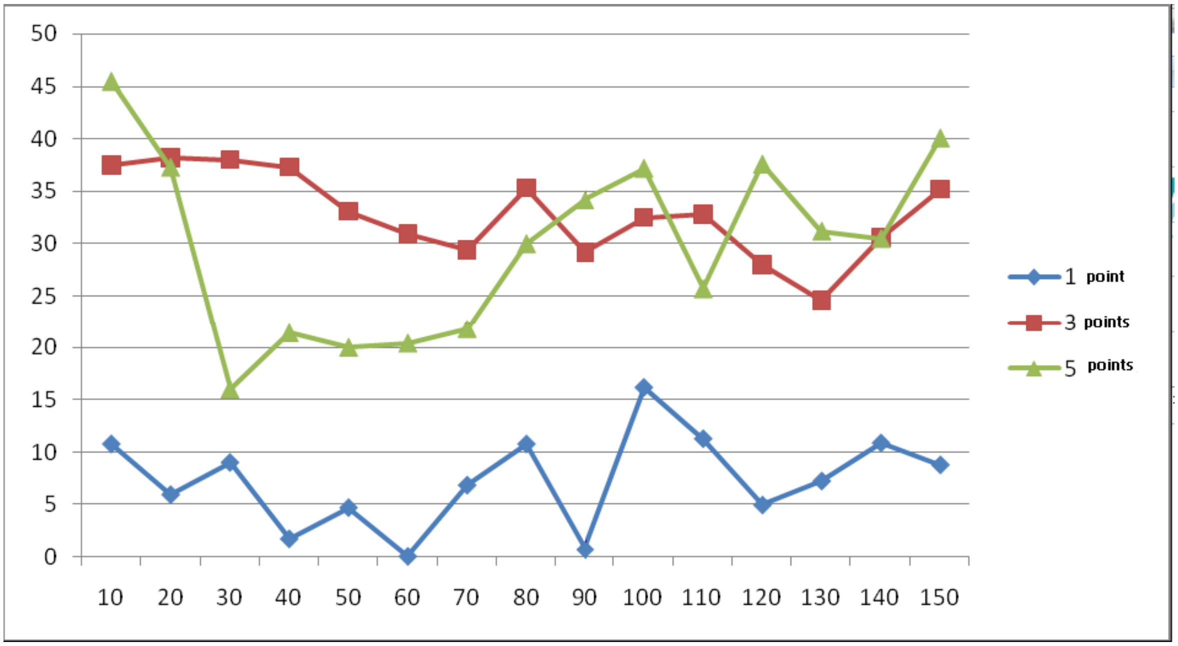


Fig.10. Forecasting error (MAPE) versus number of layers (1,3,5, 7) of neo-fuzzy network

Analyzing the presented curves we conclude that the Neo-Fuzzy Neural Network has the best results with 3 hidden layers- the error is small and less than 10%.

With one hidden layer error is also not high but is not uniformly distributed and may exceed 30%.

For 5 hidden layers the MAPE increases and may reach 35%. And finally with 7 layers MAPE reaches 60%. Thus the maximal precision we obtain with 3 hidden layers .

Besides, in process of experimental investigations were found the *optimal parameters* for algorithms for full and constructed by GMDH *neo*-fuzzy networks: The ideal ratio of learning and test samples – 75% : 25%.

- The best number of layers – 3.
- The best result at 100000. iterations
- The best result with 1 forecasted point .
- The best result with maximal error (threshold of algorithm stop) – 0.01.

6 Solving of the classification problem using the Neo-Fuzzy Neural Network

We have applied proposed Neo-Fuzzy Neural Network synthesized by the GMDH to solve the 'breast cancer in Wisconsin' benchmark classification problem.

Dataset containing 699 points have been used for this purpose (ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/cancer/cancer1/datacum). 16 points had parameters with missed values so they have been eliminated from the dataset and remaining 683 points have been separated on training set – 478 points (70%) and test set – 205 points (30%).

Each point has 9-dimensional feature vector and 1 class parameter which should be determined and identifies either benign or malignant tumor have current examined patient. Features values have been normalized on interval [-1; 1].

For comparison the same classification problem was solved using the Neo-Fuzzy Neural Network with full 3-layer structure: 10 NFNs in the first layer, 5 in the seconds, and 1 output NFN. Obtained results of classifications can be found in table 3.

When output signal be found within the range [0.3; 0.7] it is lesser probability that classification were correct. We quantify and marked out such classified samples as points outside the 'belief zone'.

Tab.3. Comparison of the Neo-Fuzzy Neural Network with full structure and structure constructed by the GMDH for the 'breast cancer in Wisconsin' benchmark classification problem.

ANN Architecture	Accuracy on training set / Points outside the 'belief zone'	Accuracy on testing set / Points outside the 'belief zone'
Network constructed by the GMDH	99,8% / 1	98% / 4
Full network	98% / 3	94% / 15

It can be seen that the Neo-Fuzzy Neural Network with architecture synthesized by the GMDH shows quite good results of classification and sufficiently exceeds in the classification quality as compared with the full network, especially on the testing set. It can be explained by fact, that full network is a more complex model and as generally known, complexness of the model leads to generalization loss and therefore classification accuracy decreases. The GMDH allows to synthesis the optimal structure that neglects inputs which are not significant. In the Fig. 11 the architecture of the Neo-Fuzzy Neural Network constructed by the GMDH is shown. It is considerably simpler, than the full network, but in spite of this it allows to achieve higher classification quality.

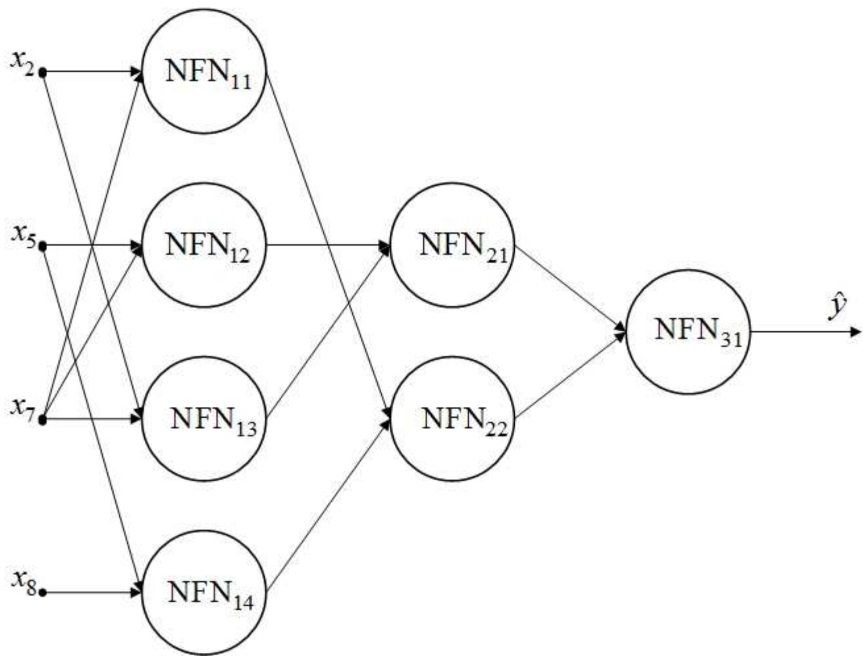


Fig. 11. The architecture of the Neo-Fuzzy Neural Network for solving the 'breast cancer in Wisconsin' benchmark classification problem synthesized by the GMDH.

7 Conclusion

The method that allows to synthesize an optimal architecture of the Neo-Fuzzy Neural Network is proposed. It based on the Group Method of Data Handling. Theoretical justification and experimental results prove the efficiency of the developed approach of the Neo-Fuzzy Neural Network architecture self-organization.

References

[1] Yamakawa T., Uchino E., Miki T., Kusanagi H. A neo fuzzy neuron and its applications to system identification and prediction of the system behavior // *Proc. 2-nd Int.Conf. on Fuzzy Logic and Neural Networks* “LIZUKA–92”. Lizuka, Japan. – 1992. – P. 477–483.

[2] Uchino E., Yamakawa T. Soft computing based signal prediction, restoration and filtering // *Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks and Genetic Algorithms. Ed. Da Ruan.* – Boston: Kluwer Academic Publisher. – 1997. – P. 331–349.

[3] Miki T., Yamakawa T. Analog implementation of neo-fuzzy neuron and its on-board learning // *Computational Intelligence and Applications. Ed. N. E. Mastorakis.* – Piraeus: WSES Press. – 1999. – P. 144–149.

[4] Bodyanskiy Ye., Kokshenev I., Kolodyazhniy V. An adaptive learning algorithm for a neo-fuzzy neuron // *Proc. 3rd Int. Conf. of European Union Soc. for Fuzzy Logic and Technology (EUSFLAT 2003).* – Zittau, Germany. – 2003. – P. 375–379.

[5] Kolodyazhniy V., Bodyanskiy Ye., Otto P. Universal approximator employing neo-fuzzy neurons // *In “Computational Intelligence: Theory and Applications.” Ed. by B. Reusch.* – Berlin-Heidelberg: Springer. – 2005. – P. 631–640.

[6] Kaczmarz S. Angenaeherte Ausloesung von Systemen Linearer Gleichungen // *Bull. Int. Acad. Polon. Sci.* – 1937. – Let. A. – S. 355–357.

[7] Kaczmarz S. Approximate solution of systems of linear equations // *Int. J. Control.* – 1993. – 53. – P. 1269–1271.

[8] Widrow B., Hoff Jr. M. E. Adaptive switching circuits // *1960 URE WESCON Convention Record.* – N.Y.: IRE. – 1960. – Part 4. – P. 96–104.

[9] Bodyanskiy Ye., Kolodyazhniy V. Adaptive nonlinear control using neo-fuzzy model / *Eds. by O. Sawodny, P. Scharff “Synergies between Information Processing and Automation”.* – Aachen: Shaker Verlag, 2004. – P. 122-127.

[10] Zaychenko Yu. “The Fuzzy Group Method of Data Handling and Its Application for Economical Processes Forecasting” - *Scientific Inquiry,* - Vol. 7, No.1, June, 2006 - p.83-96.

[11] Zaychenko Yu. Fuzzy method of inductive modeling in problems of macroeconomic indexes forecasting. *System researches and informational technologies,* #3 of 2003, p. 25-45.

[12] Zaychenko Yu. P., and Zayetz I.O. The synthesis and adaptation of fuzzy forecasting model based of self-organization method. *Science News of NTUU “KPI”,* #2 of 2001.

[13] Zaychenko Yu. P., and Zayetz I.O. Research of different types of partial descriptions in problems of synthesis of fuzzy forecasting models, *Science works of Donetsk NTU,* vol. 47, p. 341-349.

[14] Zaychenko Yu. P. Comparative analysis of forecasting models built using distinct and fuzzy GMDH with different algorithms of fuzzy forecasting models generation. *Materials of international seminar of inductive modeling IWIM 2005.*