Bulletin of National University of Uzbekistan: Mathematics and Natural Sciences

Volume 1 | Issue 3 Article 1

6-25-2018

# Neuro-fuzzy identification of nonlinear dependencies

Avazjon Marakhimov
*National University of Uzbekistan*, avaz.marakhimov@yandex.ru

Kabul Khudaybergenov
*National University of Uzbekistan*, kabul85@mail.ru

# NEURO-FUZZY IDENTIFICATION OF NONLINEAR DEPENDENCIES

Marakhimov A. R., Khudaybergenov K. K.
*National University of Uzbekistan, Tashkent, Uzbekistan* e-mail:
avaz.marakhimov@yandex.ru, kabul85@mail.ru

**Abstract**

The paper proposes a fuzzy multilayer perceptron (MLP) and a modified algorithm for its training for solving problems of identification of nonlinear dependencies. The obtained results show a sharp reduction in the search for the optimal parameters of the neuro-fuzzy model compared to classical MLP and increase its accuracy. In the work, questions of optimization of the rule base of the neuro-fuzzy model are also investigated and the temporal and spatial complexity of the proposed algorithm is analyzed. The results of computational experiments show that the number of training epochs has sharply decreased, and productivity has increased compared to the well-known MLP models.

***Keywords:*** identification, non-linear systems, neural networks, rule base, fuzzy inference, fuzzy models, fuzzy MLP.

***Mathematics Subject Classification (2010):*** 97R40, 03B52.

## 1 Introduction

Identification of nonlinear dependencies, for instance, the construction of their models based on the results of observations is an important task in engineering, economics, medicine and in other applied fields [1]. In [5,9,11], a method for identifying nonlinear dependencies using fuzzy knowledge bases was proposed. Identification of non-linear dependencies using fuzzy rule bases - represents the formation of a fuzzy knowledge base that roughly reflects the relationship between inputs and outputs using linguistic rules IF-THEN [2,12]. These rules are generated by an expert, or obtained as a result of extraction of fuzzy knowledge from experimental data. After that, the parametric identification of the studied dependence occurs by finding such weights of linguistic rules and such membership functions of fuzzy terms that minimize the deviation of the simulation results from experimental data from the training sample.

In recent years, significant progress has been made in the field of artificial neural networks (ANN). Multilayer perceptrons of forward propagation are widely studied in [1]. The combined method of fuzzy knowledge bases and neural networks called fuzzy neural networks has been applied to simulate some real problems, such as medical diagnostics [4] and in other areas. The new model was developed by A.R. Rotshtein and S.D. Shtovba [5] to identify a nonlinear dependence with a fuzzy learning network. This is a controlled neural network, called a fuzzy neural network and architecture with a single hidden layer.

This article focuses on the development of a fuzzy model of MLP for identifying multidimensional dependencies and training the network by optimizing the terms of sets of rule bases.

The performance of this model was compared with the implementation of the classic MLP model developed for the same tasks. It should be noted that MLP networks and fuzzy-MLP networks often use the backpropagation concept for training.

## 2 Statement of the problem

We will assume that the identifiable non-linear dependence is represented by input-output of data sample :

$$(X_r, y_r)\,,\ r = \overline{1, M}, \tag{1}$$

where $X_r = (x_{r,1}, x_{r,2}, ..., x_{r,n})$ - input vector in the $r$-th pair of the training set and $y_r$ - corresponding output; $M$-dataset size.

The problem of training the neuro-fuzzy Mamdani model in the sample (1) comes down to finding a vector $(B, C, W)$, which provides:

$$RMSE = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\left(y_r - F\left(X_r; B, C, W\right)\right)^2} \to \min, \tag{2}$$

where $B$ and $C$-vector of parameters of membership functions; $W$ -fuzzy knowledge base rule weight vectors; $F\left(X_r; B, C, W\right)$-output result for input vector $X_r$ according to a fuzzy model with parameters $(B, C, W)$.

The output of a fuzzy model depends on its structure - the base of rules and parameters: membership functions, implementations of logical operations, defuzzification method, as well as coefficients of membership functions in the conclusions of the rules for a model of Mamdani type. Finding the structure and parameters of a fuzzy model that provides the minimum value of criterion (2) is an identification problem.

## 3 Multilayer Perceptron Architecture

The architecture of a multilayer perceptron consists of several layers of neurons [8]. Input data is supplied to the first layer (input layer-0), after which this input data is distributed to the inputs of layer-1. In layer-0, calculation is not performed, it can be considered as a sensor layer. The last layer is the output layer that outputs the processed data. Between the input and output layers, the number of hidden layers can be increased or decreased based on the problem for which the model is developed. If to create a network with too many layers, then after training the network, the model will give too much accuracy. But such a model is considered a overtrained network and the use of such a model becomes unsuitable. We are considering a network having 3 layers: input layer, hidden layer and output layer. We show that a network having such a number of layers is sufficient to identify nonlinear systems.

The following formulas describe the architecture structure of a multilayer perceptron.

$$net_j = \sum_{i=1}^{n} x_i w_{ji}^{(1)} + b_j \tag{3}$$

Where $w_{ji}^{(1)}$-weights of the input layer, $b_j$-threshold parameter of input layer,

$$z_j = f(net_j) \tag{4}$$

$$net^* = \sum_{j=1}^{m} z_j w_j^{(2)} + b_0 \tag{5}$$

where $w_{ji}^{(2)}$-weights of the output layer, $b_0$-threshold parameter of output layer,

$$y = f(net^*) \tag{6}$$

$$\Delta w_j^{(2)} = \eta E f(net^*) y, \ \Delta b_0 = \eta E f(net^*) \tag{7}$$

$$\Delta w_{ji}^{(1)} = \eta E f'(net_j) x^{(i)} f'(net^*) w_j^{(2)}, \ \Delta b_i = \eta E f'(net_j) f'(net^*) w_j^{(2)} \tag{8}$$

where $i = \overline{1, n}$, $j = \overline{1, m}$, $f(\cdot)$- activation function. For the multilayer perceptron, we used the activation function of the sigmoidal type.

*Pseudo code for a multilayer perceptron:*

*1. Initialize* $w^{(1)}, w^{(2)}, b, b^0, \eta$ *and convergence criterion* $\varepsilon$
*2. Repeat until (stop criterion):*
  *//forward propogation*
  *1)* $x \leftarrow$ *randomly selected training sample*
  *2) For all hidden neurons:*
    *calculate* $net^j$ *and* $z^j$
  *3) calculate* $net^*$ *and* $y = f(net^*)$
  *4) calculate error,* $ERROR = d - y$;
  *// backward propogation*
  *5) calculate* $\Delta w_j^{(2)}$, $\Delta b_0$, *update the value of weights and thresholds*
  *6) For all hidden neurons:*
    *calculate* $\Delta w_{ji}^{(1)}$, $\Delta b_i$ *and update input weights and input thresholds*
*3. Output:* $w^{(1)}$, $w^{(2)}$, $b$, $b^0$.

where $d$-experimental value of the output.

# 4 Neuro-fuzzy identification

A model is considered in which the relationship between inputs and outputs is described by a knowledge base of fuzzy rules like IF-THEN. We will use knowledge base in Mamdani type. We write the neuro-fuzzy network with the rule base Mamdani as follows:

$$\text{R}_j\text{:}\ \ \text{IF}\left(x_1=a_1^{j1}\right)\ \text{AND}\left(x_2=a_2^{j2}\right)\ldots$$
$$\text{AND}\ \left(x_n=a_n^{jp}\right)\ \text{with weight}\left(w_{jp}\right)\ \text{THEN}\ y=d_j \quad (9)$$

where $\text{R}_j$ - $j$-rule of knowledge base, $a_i^{jp}\in A_i$ - fuzzy term, evaluating variable $x_i$; $A_j$ - linguistic term set ($i=\overline{1,n}$, $j=\overline{1,m}$, $p=\overline{1,k}$ ); $w_{jp}$-weight coefficient; $d_j$-fuzzy conclusion $j$-rule; $m$-number of rules in the knowledge base;

The following system of fuzzy logical equations will correspond to this modified fuzzy knowledge base:

$$\mu^{d_j}\left(x_1,x_2,...,x_n\right)=\mathop{\vee}_{p=1}^{k}\left(w_{jp}\left[\mathop{\wedge}_{i=1}^{n}\mu^{a_i^{jp}}\left(x_i\right)\right]\right),\ j=\overline{1,m} \quad (10)$$

where $\mu^{a_i^{jp}}\left(x_i\right)$-membership function of the input $x_i$ fuzzy term $a_i^{jp}$; $\mu^{d_j}\left(x_1,x_2,...,x_n\right)$-membership function of the output $y$ fuzzy term $d_j$;

If the model of an object with a discrete output, which corresponds to a fuzzy knowledge base (9), it can be represented in a more compact form:

$$\mu^{d_j}\left(x_1,x_2,...,x_n\right)=\mu^{d_j}\left(X,B,C,W\right),\ j=\overline{1,m} \quad (11)$$

wher $X=(x_1,x_2,...,x_n)$ - vector of input variables; $B=(b_1,b_2,...,b_n)$ and $C=(c_1,c_2,...,c_n)$ - vectors of parameters of membership function (12); $W=(w_1,w_2,...,w_n)$ - weight vector of fuzzy knowledge base rule;

In this model, we use the bell-type membership function. It should be noted that the membership function of this type is considered due to its simplicity. However, other types of membership function can also be implemented for the architecture of a fuzzy multilayer perceptron.

The identification problem for an object with a continuous output can be formulated as follows: find a matrix that satisfies the restrictions on the ranges of variation of parameters $(B,C,W)$ and provide (2).

For the discrete case:

$$\sum_{p=1}^{M}\left\{\sum_{j=1}^{m}\left(\mu^{d_j}\left(X_p,B,C,W\right)-\mu_p^{d_j}\left(y\right)\right)^2\right\}=\min_{B,C,W}$$

The proposed architecture is similar to the architecture of the classical architecture of the multilayer perceptron and is shown in Fig.1.

The process of training a neural-fuzzy network is similar to the procedure for training traditional neural networks according to the «back-propagation» rule. Learning of this algorithm also consists of two phases - forward and reverse. Therefore, here we give only the formulas included in the fuzzy multilayer perceptron, and the rest of the formulas are given in formulas 3-8. Calculation of the degree of membership of the input values to linguistic terms by the formula (12).

$$\mu^j\left(x_i\right)=\frac{1}{1+\left(\frac{x_i-b_i^j}{c_i^j}\right)^2},\ i=\overline{1,n},\ j=\overline{1,m} \quad (12)$$

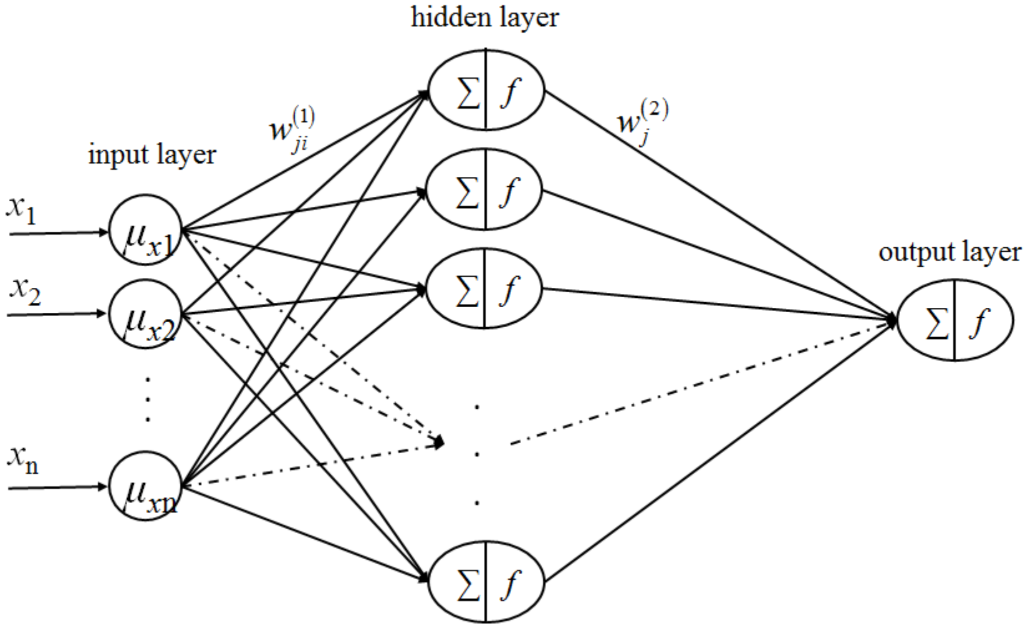


Figure 1: Architecture of the fuzzy multilayer perceptron

Accordingly, the coefficients $b_i$ and $c_i$ will be updated with the following formulas:

$$\frac{\partial \mu^j(x_i)}{\partial c_i^j} = \frac{2c_i^j \left(x_i - b_i^j\right)^2}{\left(\left(c_i^j\right)^2 + \left(x_i - b_i^j\right)^2\right)^2} \tag{13}$$

$$\frac{\partial \mu^j(x_i)}{\partial b_i^j} = \frac{2\left(c_i^j\right)^2 \left(x_i - b_i^j\right)}{\left(\left(c_i^j\right)^2 + \left(x_i - b_i^j\right)^2\right)^2}, \; i = \overline{1,n}, \; j = \overline{1,m} \tag{14}$$

On the output, defuzzification method of the center of gravity is used to obtain a crisp value. It is believed that this defuzzification method provides the best indicators of accuracy and tuning speed of a fuzzy model [7]. The method of defuzzification of the center of gravity has the following form:

$$y = \frac{\sum_{j=1}^{m} y_j \mu^{d_j}}{\sum_{j=1}^{m} \mu^{d_j}}$$

*Pseudo code for a fuzzy multilayer perceptron:*
*1. Initialize $w^{(1)}, w^{(2)}, p, p^0, b, c, \eta$ and convergence criterion $\varepsilon$ 2. Repeat until (stop criterion):*
 *//forward propogation*
 *1) $x \leftarrow$ randomly selected training sample*
 *2) For all hidden neurons:*

*calculate* $net^j$ *and* $z^j$

*3) calculate* $net^*$ *and* $y = f(net^*)$

*4)* $y =$ *Defuzzify the output* $\mu_y(y)$

*5) calculate error,* $ERROR = d - y$;

*// backward propogation*

*6) Calculate* $\Delta w_j^{(2)}$, $\Delta p_0$, *update the value of weights and thresholds*

*7) For all hidden neurons:*

*Calculate* $\Delta w_{ji}^{(1)}$, $\Delta p_i$ *and update input weights and input thresholds*

*8) For all parameters of the membership function:*

*Calculate* $\Delta b_i$, $\Delta c_i$ *and update coefficients*

*3. Output:* $w^{(1)}$, $w^{(2)}$, $p$, $p^0$, $b, c$.

where $d$-experimental value of the output.

# 5 Analysis of algorithm complexity

The complexity of time and memory space for both proposed algorithms is the same. In the proposed model, in fuzzy MLP, one additional matrix is used, in which the fuzzified input data(sample) is stored. It should be noted that this space of memory does not affect the learning speed for the proposed algorithms and, as well as for modern computers, this small memory space is not a problem. In both models, the size of the input neurons is $n$; the size of the hidden neurons is $m$, and the size of the output neuron is 1. The calculation is carried out in a forward propagation and back propagation. With a forward propagation, the calculation occurs from the calculation of the input (layer-0) to the hidden layer (layer-1) and has a total time of $O(n\times m)$ calculations. The next forward calculation takes place from the hidden layer to the output layer (layer-2), and it will take $O(m)$ time. Similarly, in the process of back propagation, from output to hidden layers and from hidden to the input layer, calculations will be performed on $O(m)$ and $O(n \times m)$ times, respectively. Thus, the time complexity of the algorithm can be calculated, as shown in the following form:

$$T(n, m) = 2O(n \times m) + 2O(m)$$

consequently,

$$T(n, m) = O(n \times m).$$

# 6 Simplification of rule base

The transparency and compactness of the initial rule base can often also be improved by reducing and simplifying the rule base. Distinctness of rules and terms (fuzzy sets) arising as a result of projection depends on determining the correct number of rules in the data. To reduce the complexity of the rule base, we iteratively seek to determine redundancy in the rule base and apply simplification of the rule base.

A method of simplifying the rule base, based on similarity, was proposed in [6]. A similarity measure is used to quantify redundancy in fuzzy sets in the rule base.

Similar fuzzy sets representing compatible notion are combined to produce a generalized notion represented by a new fuzzy set that replaces similar ones in the rule base. This reduces the number of similar fuzzy sets (linguistic terms) used in the model. A similarity measure is also used to define “non-caring” terms, that is, fuzzy sets in which all elements of a domain have an affinity close to 1. Thus, the set of model terms can be reduced without the need to remove any rules.

We apply a similarity measure based on set-theoretic intersection and union operations.

$$S\left(A,B\right)=\frac{\left|S\left(A\cap B\right)\right|}{\left|S\left(A\cup B\right)\right|}$$

where $A$ and $B$ - fuzzy set, $|\cdot|$ - denotes the power of a set, operators $\cap$ and $\cup$ represent the intersection and union, respectively. For discrete sets $X=\{x_j|j=1,2,...,m\}$, this can be written as follows:

$$S\left(A,B\right)=\frac{\sum_{j=1}^{m}\left[\mu_A\left(x_j\right)\wedge\mu_B\left(x_j\right)\right]}{\sum_{j=1}^{m}\left[\mu_A\left(x_j\right)\vee\mu_B\left(x_j\right)\right]}$$

where $\wedge$ and $\vee$-minimum and maximum operators, respectively. $S$ is a symmetric measure in $[0,1]$. If $S(i,j)=1$, then two membership functions $A_i$ and $A_j$ are equal and $S(i,j)$ becomes 0, if membership functions do not intersect.

Similar fuzzy sets combine when their similarity exceeds a user-set threshold $\theta\in[0,1]$. In the experiment, the value $\theta=0.6$ is used. The optimal choice of this parameter depends on the number and type of membership functions and, therefore, this is a problem characteristic of some degree; however, as a rule, values are in the range $[0,4-0,7]$ can be the best choice. Different settings lead to a more or less conservative fusion of fuzzy sets. Combining reduces the number of different fuzzy sets used in the model, and thereby improves transparency.

# 7 Experimental Details and Results

An object with two inputs is considered. $x_1,x_2\in[-2,2]$ and one output given by the dependence (Fig.2a):

$$y=x_1\times\mathrm{e}^{\left(-x_1^2-x_2^2\right)}\tag{15}$$

It is necessary to conduct training based on the above proposed models, synthesize a fuzzy model and configure it according to a fuzzy training data set. The adequacy of the fuzzy model have to be checked by the criterion (1). Compare identification results for crisp and fuzzy training samples. Data set of the samples are given in (15).

Graphics of membership functions of fuzzy terms of input variables after tuning and optimization are shown in Fig. 4 (a), and the values of their parameters $b$ and $c$ are given in Tables 2 and 3 , respectively.

Testing of crisp and fuzzy models indicates an acceptable quality of identification of nonlinear dependence (15) (Fig.3a,3b). Table 1 shows the convergence for different

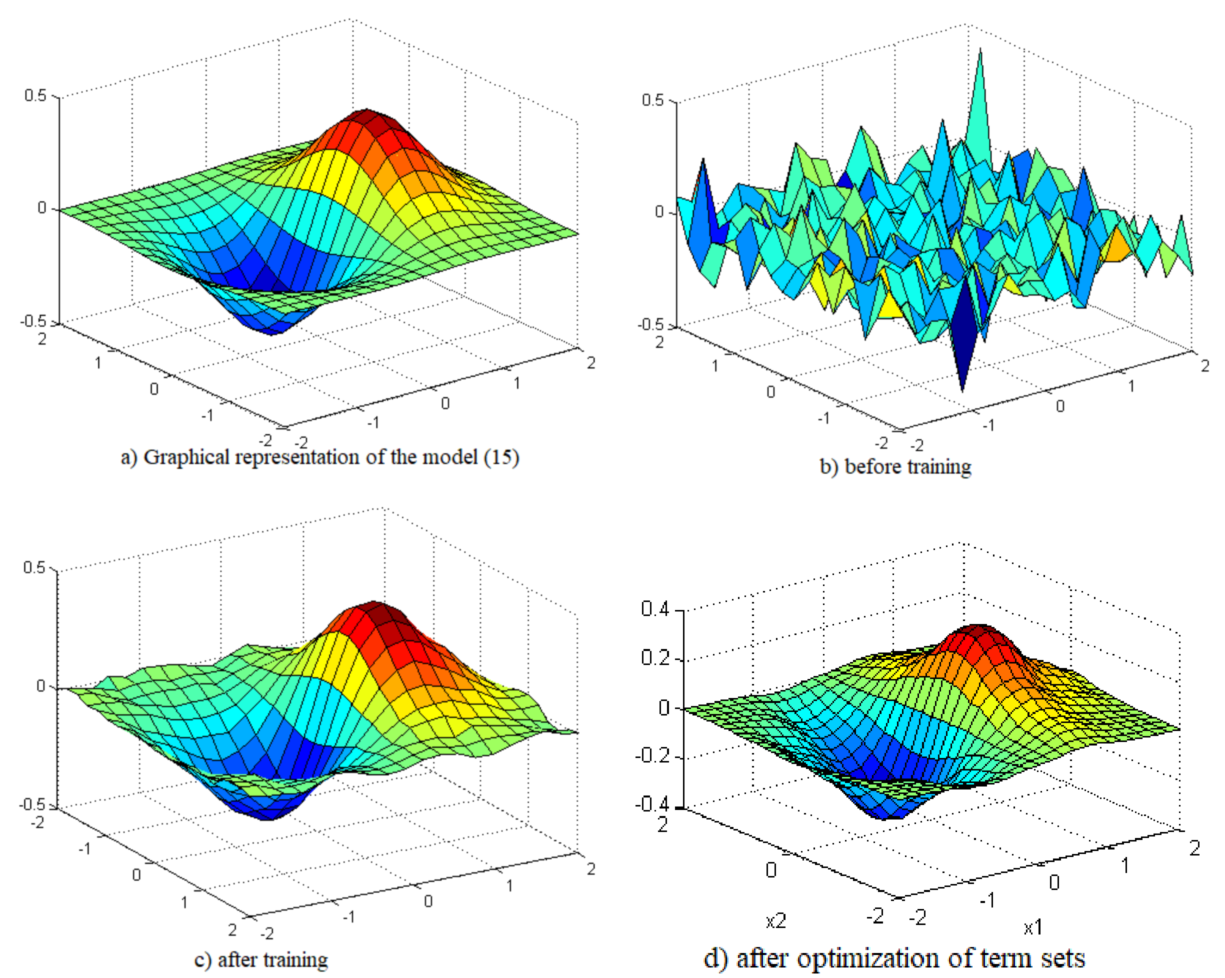


Figure 2: Geometric images of the characteristics of the object (15)

values of learning speed in $\alpha$. Figure 3b shows that the RMSE obtained using the proposed fuzzy MLP algorithm approaches the corresponding minimum after 600-700 epochs.

From the test results, we can assume that the proposed fuzzy model is exceeds compared to a crisp model in convergence.

# 8 Conclusions

A fuzzy MLP has been developed to identify nonlinear systems. The results show that the proposed model converges to its minimum RMSE for 600-700 epochs and reaches a convergence coefficient $93-95\%$. From the results of the conducted computational experiments, we can assume that the proposed fuzzy model significantly exceeds the

Table 1: Performance of the models

| | Minimum RMSE | | Time (sec.) ) | |
|---|---|---|---|---|
| $\alpha$ | MLP | Fuzzy-MLP | MLP | Fuzzy-MLP |
| 0.30 | 1.0161 | 0.0234 | 26.71 | 24.10 |
| 0.40 | 1.0145 | 0.0207 | 25.74 | 22.42 |
| 0.45 | 1.0754 | 0.0159 | 25.97 | 21.53 |
| 0.50 | 1.2273 | 0.0268 | 23.68 | 20.04 |
| 0.55 | 1.3205 | 0.0272 | 23.55 | 20.19 |

Table 2: Parameters of membership functions of variable terms $x_1$

| Terms | | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ |
|---|---|---|---|---|---|---|---|
| before training | $b$ | -1.9600 | -1.2000 | -0.4000 | 0.4000 | 1.2000 | 2.0000 |
| | $c$ | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| after training | $b$ | -1.9947 | -1.1667 | -0.4678 | 0.4538 | 1.2048 | 1.9823 |
| | $c$ | 0.4203 | 0.2297 | 0.4620 | 0.4519 | 0.2225 | 0.4247 |

Table 3: Parameters of membership functions of variable terms $x_2$

| Terms | | $B_1$ | $B_2$ | $B_3$ | $B_4$ | $B_5$ | $B_6$ |
|---|---|---|---|---|---|---|---|
| before training | $b$ | -1.9600 | -1.2000 | -0.4000 | 0.4000 | 1.2000 | 2.0000 |
| | $c$ | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| after training | $b$ | -1.9941 | -0.6719 | -0.0760 | 0.0801 | 0.6851 | 2.0225 |
| | $c$ | 0.2281 | 0.4192 | 0.1911 | 0.1907 | 0.4293 | 0.2350 |

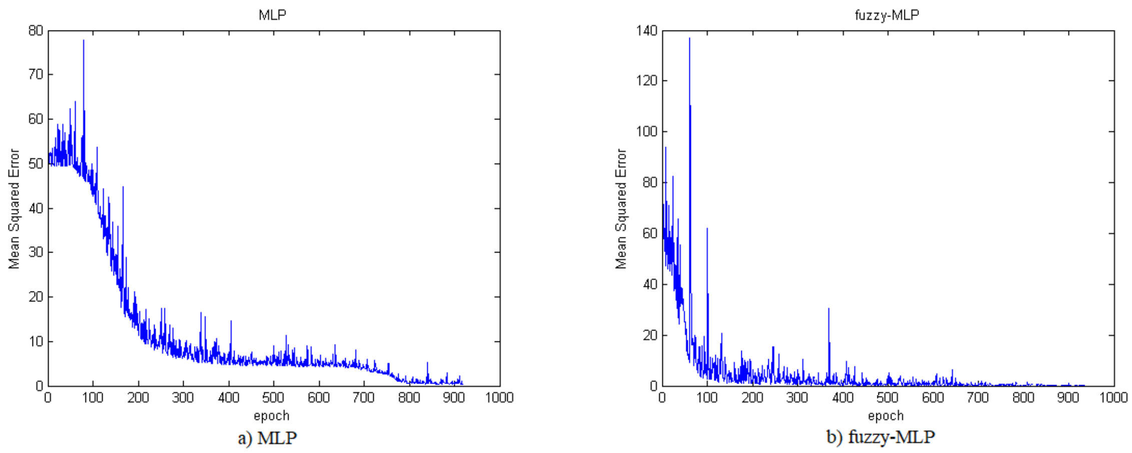


Figure 3: Graphic of convergence

MLP in terms of learning speed and accuracy. And also, the performed computational experiments show that the fuzziness in the experimental data is not an obstacle to the identification of nonlinear dependencies.

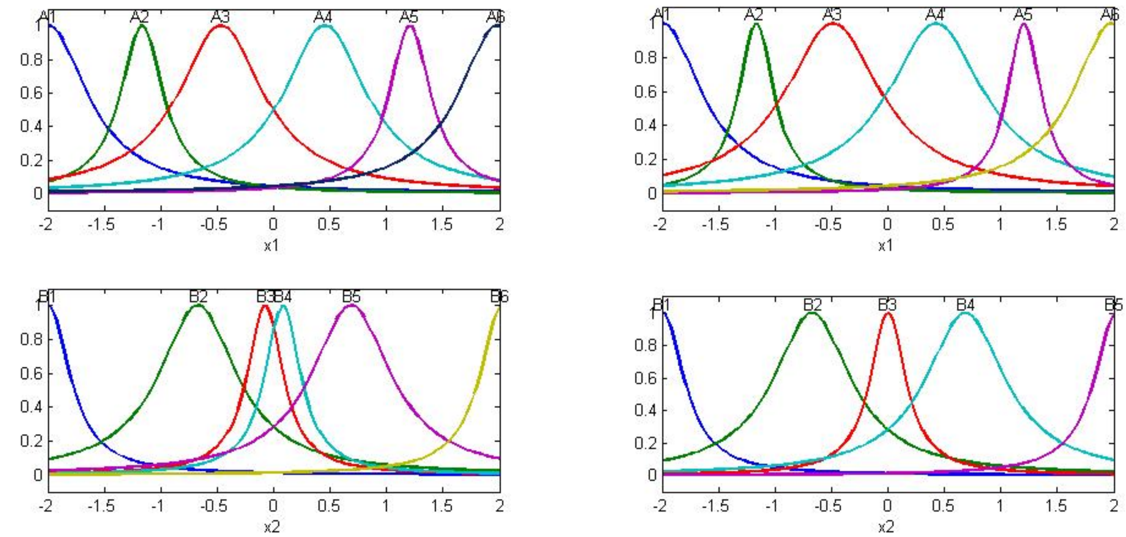


Figure 4: Graphic of convergence

The possibilities of using a fuzzy training set and neuro-fuzzy models can be used in control systems, medicine, and other applied fields, where experimental data for identifying the studied input-output dependence are formed on the basis of a subjective nature.

# References

[1] Haykin S. Neural Networks: A Comprehensive Foundation (2nd Edition). Prentice Hall, (1999). – 842 p.

[2] Yen J., Wang L., Gillespie C.W. Improving the interpretability of TSK fuzzy models by combining global learning with local learning. IEEE Transactions on Fuzzy Systems, Vol. 6, Issue 4, 530–537 (1998).

[3] Nikov A., Georgiev T. A fuzzy neural network and its matlab simulation. Proceedings of ITI99 21st International Conference on Information Technology Interfaces, Pula, Croatia, June 15-18, 413–418 (1999).

[4] Rotshtein A.P. Design and tuning of fuzzy if-then rules for medical diagnosis. Fuzzy and Neural-Fuzzy Systems in Medical and Biomedical Engineering. CRC Press, 35–97, (1998).

[5] Rotshtein A.P., Shtovba S.D. Identification of a nonlinear dependence by a fuzzy knowledgebase in the case of a fuzzy training set. Cybernetics and Systems Analysis, Vol. 42, Issue 2, 176–182 (2006).

[6] Setnes M., Babuska R., Kaymak U., van Nauta Lemke H.R. Similarity measures in fuzzy rule base simplification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol. 28, Issue 3, 376–386 (1998).

[7] Zimmermann H.J. Fuzzy Set Theory and Its Applications. Kluwer, Academic Publisher, Dordrecht, Boston, MA, 2nd ed., (1991). – 315 p.

[8] Borisov B., Kruglov B., Fedulov A. Fuzzy models and networks, 2-edition. Goryachays, (2012). – 284 p.

[9] Mityushkin Yu.I., Mokin B.I., Rotshtein A.P. Soft Computing: identification of patterns by fuzzy knowledge bases. UNIVERSUM, Vinnytsia, 2002. – 145 p.

[10] Pegat A. Fuzzy modeling and control, trans. from English 2nd ed. Beanom. Knowledge lab, Moscow, (2013). – 798 p.

[11] Tarek T.A. Parametric system identification using neural networks. Applied Soft Computing, Vol. 47, 251–261 (2016).

[12] Ganesh K., Chellasamy R., Deepa S.N. Formation of Fuzzy if-then Rules and Membership Function using Enhanced Particle Swarm Optimization. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 21, No. 1, 103–126 (2013).