Webology, *Volume 17, Number 1, June, 2020*

Method of Artificial Neural Networks Teaching

Khudhair Abed Thamer

PhD, Kuliyyah Al-Maarif University College, Karrada Street 3, Ramadi, 31001, Republic of Iraq. ORCID: 0000-0002-1575-2294. E-mail: prokopenko_evg@ukr.net

Received January 15, 2020; Accepted April 20, 2020

Abstract

The technique of artificial neural networks training has been developed. A distinctive feature of the proposed technique is that it provides training not only for the synaptic weights of the artificial neural network but also for the type and parameters of the membership function. If it is impossible to provide, the specified quality of functioning of artificial neural networks due to the learning of the parameters of the artificial neural network, the architecture of artificial neural networks is trained. The choice of architecture, type and parameters of the membership function takes into account the computing resources of the tool and taking into account the type and amount of information that is coming to the input of an artificial neural network. Also, while using the proposed method, there is no accumulation of error learning artificial neural networks as a result of processing information, which is supplied to the input of artificial neural networks. The development of the proposed methodology is due to the need to train artificial neural networks, in order to process more information, with the uniqueness of the made decisions. According to the results of the research, it is established that the mentioned training method provides on average 16-23 percent higher efficiency of training of artificial neural networks and does not accumulate errors during training. This technique will allow to train artificial neural networks; identify effective measures to improve the performance of artificial neural networks. Also, the developed technique will increase the efficiency of the functioning of artificial neural networks by learning the parameters and architecture of artificial neural networks. The technique proposed by the authors reduces the use of computing resources for support and decision-making systems. Using the developed methodology will develop measures that are aimed at improving the efficiency of artificial neural networks training and increase the speed of the processing information.

Keywords

Artificial neural networks; Synaptic scales; Membership function; Processing information; Intelligent decision support systems

Introduction

Decision-making support systems (DMSS) are actively used in all areas of human life. They received special distribution in the processing of large data sets, providing information support to the decision-making process by decision-makers. The analysis of scientific researches shows that, at present, the basises of existing DMSS are the methods of artificial intelligence.

The creation of intelligent DMSS was a natural continuation of the widespread use of classic DMSS. Intelligent DMSS provide information support for all production processes and services of enterprises (organizations, institutions). With the help of intellectual DMSS design, production and marketing of products, financial and economic analysis, planning, personnel management, marketing, support for the creation (operation, repair) of products and prospective planning.

Also, these intellectual DMSS have been widely used to solve specific military tasks, namely (Kalantaievska et al, 2018): scheduling the deployment, operation of communications systems and data transmission; automation of control of troops and weapons; collecting, processing, and generalizing intelligence about the status of intelligence objects and more. The main tool for solving computational and other problems in modern intellectual DMSS is artificial neural networks (ANN) that are evolving. Evolving ANN has universal approximation properties and fuzzy inference capabilities. Evolving ANN has been widely used to address the various challenges of data mining, identification, emulation, forecasting, intellectual management, etc. Evolving ANN provides stable operation in conditions of nonlinearity, uncertainty and randomness, all kinds of disturbances (Kuchuk, Mohammed, Shyshatskyi & Nalapko, 2019). Despite their successful application these systems have several drawbacks that are associated with their use.

Among the most significant disadvantages are the following:

1. The complexity of choosing the architecture of the system. Typically, a model based on the principles of computational intelligence has a fixed architecture. In the context of ANN, it means that the neural network has a fixed number of neurons and connections. In this regard, adapting the system to new processing data that is different from the previous nature may be problematic.
2. Batch and multi-epoch training require considerable time resources. Such systems are not adapted to operate online with a sufficiently high rate of new data to be processed.

3. Many of the existing systems of computational intelligence cannot determine the evolving rules by which the system develops, and can also present the results of their work in terms of natural language. Thus, the urgent task is to develop new teaching methods for ANN that will solve these difficulties.

Literary Data Analysis and Problem Statement

The research done by Zhang and Ding (2017), an analysis of the properties of ANN was used, which was used to predict the concentration of air pollutants. The work emphasized that ANN has a low convergence rate and a local minimum. It is suggested to use an extreme training machine for ANN, which provides high efficiency of generalization at extremely high speed of training. The disadvantages of this approach include the accumulation of ANN errors during the calculations, the inability to select parameters and the type of membership function.

In a paper by Katranzhy, Podskrebko and Krasko (2018), an operational approach for spatial analysis in the maritime industry is presented to quantify and map-related ecosystem services. This approach covers the three-dimensionality of the marine environment, considering separately all marine areas (sea surface, water column and seabed). In fact, the method builds 3-dimensional models of the sea by estimating and mapping each of the three marine domains by adopting representative indicators. The disadvantages of this method include the impossibility of flexible adjustment (adaptation) of estimation models while adding (excluding) indicators and changing their parameters (compatibility and significance of indicators).

The research done by Manea, Carloa, Depellegrin, Agardy and Gissi (2019) presents a machine learning model for the automatic identification of requests and the provision of support information exchanges between members of the online community. This model is designed to handle a large number of messages from users of social networks. The disadvantages of this model are the lack of mechanisms for evaluating the adequacy of the decisions being made and the high computational complexity.

In a work by Çavdar, and Ferhatosmanoğlu (2018), the use of ANN for the detection of heart rhythm abnormalities and other heart diseases is presented. The error propagation algorithm is used as the ANN training method. The disadvantages of this approach are its limited training of only synaptic weights, without learning the type and parameters of the membership function.

The research done by Zhdanov (2016), the use of ANN for detecting avalanche origin is presented. The error propagation algorithm is used as the ANN training method. The disadvantages of this approach are its limited training of only synaptic weights, without learning the type and parameters of the membership function.

The paper by Kanev et al. (2017) presents the use of ANN to identify problems of anomaly detection in home authorization systems. Kohonen's “ANN Winner” algorithm is used as a method of training for ANN Kohonen. The disadvantages of this approach are the accumulation of errors in the learning process, limited learning only synaptic weights, without learning the type and parameters of the membership function.

The research done by Sreeshakthy and Preethi (2016), the use of ANN for the detection of anomaly detection problems in human encephalograms is presented. The method of fine-tuning of the ANN is used as a method of ANN training. The disadvantages of this approach are the accumulation of errors in the learning process, limited learning only synaptic weights, without learning the type and parameters of the membership function.

In the research conducted by Chica, Zaputt, Encalada, Salamea and Montalvo (2019), the use of machine learning methods, namely ANN and genetic algorithms, is presented. A genetic algorithm is used as a method of ANN training. The disadvantages of this approach are its limited training of only synaptic weights, without learning the type and parameters of the membership function.

In the study done by Massel, Gerget, Massel and Mamedov (2019), the use of machine learning methods, namely ANN and differential search method, is presented. During the research, the development of a hybrid method of training ANN, that is based on the use of the algorithm of backpropagation and differential search. The disadvantages of this approach are its limited training of only synaptic weights, without learning the type and parameters of the membership function.

In the work (Abaci and Yamacli, 2019), the development of ANN training methods using a combined approximation of the response surface, which provides the smallest learning and forecasting errors, was developed. The disadvantage of this method of accumulation of errors during training and the inability to change the architecture of the ANN during training.

The research done by Mishchuk and Vitynskyi (2018) describes the use of ANN to evaluate the efficiency of the unit using the previous time series of its performance. SBM (Stokhastic Block Model) and DEA (Data Envelopment Analysis) models are used to teach ANN. The disadvantages of this approach are the limitations in the choice of network architecture, training only synaptic weights.

In the study conducted by Kazemi and Faezirad (2018), the use of ANN for the estimation of geomechanical properties is presented. The error propagation algorithm is used as the ANN training method. Improving the characteristics of the backpropagation algorithm is achieved by increasing the training sample. The disadvantages of this approach are its limited training of only synaptic weights, without learning the type and parameters of the membership function.

In the research done by Parapuram, Mokhtari and Hmida (2018), the use of ANN for the estimation of traffic intensity is presented. The error propagation algorithm is used as the ANN training method. Improving the characteristics of the backpropagation algorithm is achieved by using bandwidths between each layer so that each layer sets only a residual function over the results of the previous layer. The disadvantages of this approach are its limited training of only synaptic weights, without learning the type and parameters of the membership function.

The analysis of scientific works showed that well-known teaching methods are used to train artificial neural networks. These methods tend to focus on the training of synaptic scales or function functions. The use of known algorithms (methods, techniques) for training artificial neural networks, even with advanced characteristics, does not satisfy the existing and promising requirements for them. Considering the stated purpose of this article is to develop a methodology for training artificial neural networks for intelligent decision support systems to solve the following problems:

- increasing the amount of information capable of processing artificial neural networks;
- increasing decision-making reliability by intelligent decision support systems;
- increasing the speed of adaptation of the architecture and parameters of artificial neural networks in accordance with emerging tasks;
- avoiding deadlock situations for us training artificial neural networks;
- ensuring predictability of the learning process of artificial neural networks;
- ensuring the uniqueness of decisions made by intelligent decision support systems.

The Purpose and Objectives of the Research

The purpose of the research is to develop a methodology for training artificial neural networks for intelligent decision support systems, which allows for the processing of more information, with the uniqueness of the made decisions. To achieve this goal, the following tasks were set:

– to set the task of artificial neural networks training;
– to develop an algorithm for artificial neural networks training teaching for the intelligent decision support systems;
– to identify the advantages and disadvantages of the proposed methodology.

Setting the Task of Artificial Neural Networks Training

The architecture, the multilayer neuro-phase evolution system, that is shown in Figure 1 consists of five consecutive layers. The input (zero) layer of the neuro-phase system is given $(n \times 1)$ as a dimensional vector of signals-images $x(k) = (x_1(k), x_2(k), ..., x_n(k))^T$ (here $k = 1, 2...$ is the current discrete time), to be processed.

The first hidden layer contains *nh* membership functions, $\mu_{li}(x_i), i=1,2....,n; l=1,2,...,h.$ Thus, for each input, *h* membership functions are given. The first hidden layer performs the phasing of the input space. This implies that in the process of learning-evolution must adjust the actual parameters of these functions and their number: Figure 1 is the number of nodes of the first hidden layer μ_h. The second hidden layer provides an aggregation of the levels of belonging calculated in the first hidden layer and consists of *h* multiplication blocks. The third hidden layer is a layer that adjusts the synaptic weights $w_1, w_2, ..., w_h$ to be determined in the controlled learning process. The fourth hidden layer is formed by two adders, which calculates the sum of the output signals of the second and third hidden layers. And finally, in the fifth (output) layer, a decapitation is performed, which calculates the NFS (neuro-fuzzy system) $\hat{y}(k)$ output. Thus, if a vector signal $x(k)$ is an input to the neuro-phase of the system, the elements of the first hidden layer do its fuzzyfication by calculating the levels of belonging $0 < \mu_{li}(x_i(k)) \le 1.$ Usually, traditional Gaussians are used as the membership functions.

$$\mu_{li}(x_i(k)) = \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma_i^2}\right), \qquad (1)$$

where c_{li} is the parameter of the center of the *l*-th function of belonging to the *i*-th input, σ_i is the width parameter of the *i*-th accessory membership function. It is worth noting that pre-normalizing the data at some interval, for example, $-1 \le x_i(k) \le 1$ allows to simplify the calculations, since the width parameters σ_i can be assumed to be the same for all inputs, that is $\sigma_i = \sigma$. In addition to Gaussians (1), other nuclear functions can be used, such as B-splines corresponding to the single-break condition, paired wavelets, flexible activation-membership functions (Prokoptsev, Alekseenko & Kholodov, 2018), and others.

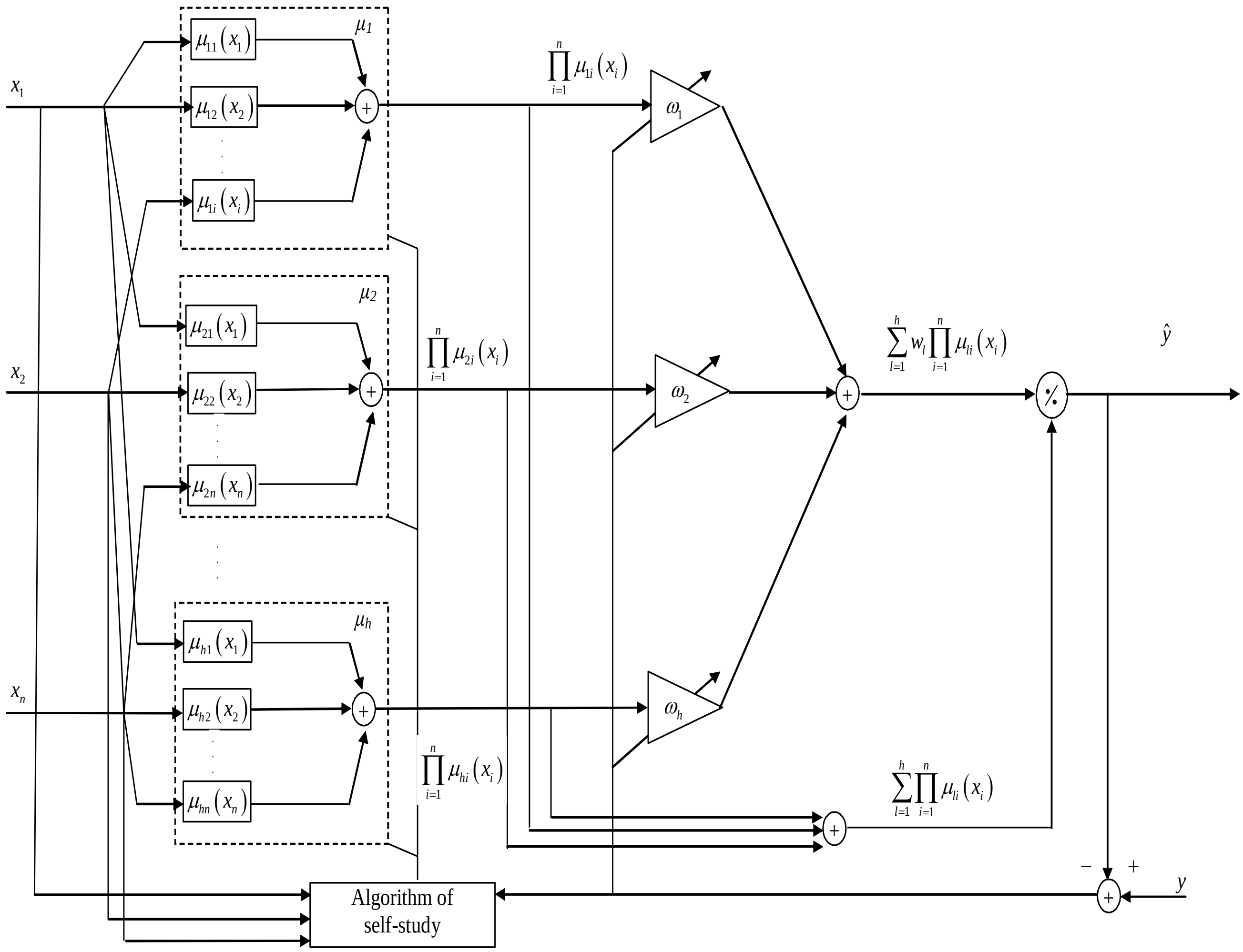

Figure 1. The architecture of a multi-layer neuro-phase evolving system

In the second hidden layer, the aggregate values $\prod_{i=1}^{n}\mu_{li}\left(x_i\left(k\right)\right)$ are calculated for Gaussians with the same width parameters (Bodyanskiy, Pliss and Vynokurova, 2013):

$$\prod_{i=1}^{n}\mu_{li}\left(x_i\left(k\right)\right)=\prod_{i=1}^{n}\exp\left(-\frac{\left(x_i\left(k\right)-c_{li}\right)^2}{2\sigma^2}\right)=\exp\left(\frac{\left\|x\left(k\right)-c_l\right\|^2}{2\sigma^2}\right),$$

where $c_l=\left(c_{l1},c_{l2},...,c_{l\,n}\right)^T$.

Thus, the signals at the outputs of the multiplication units of the second hidden layer are similar to the signals at the outputs of the neurons of the first hidden layer of conventional RBFN (radial basis function networks) (Bodyanskiy, Pliss & Vynokurova, 2013).

The outputs of the third hidden layer are the values $w_l \prod_{i=1}^{n} \mu_{li}\left(x_i\left(k\right)\right)$, the fourth is the $\sum_{l=1}^{h} w_l \prod_{i=1}^{n} \mu_{li}\left(x_i\left(k\right)\right)$ i $\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}\left(x_i\left(k\right)\right)$, finally, a signal appears at the output of the system (the fifth output layer)

$$\hat{y}\left(x\left(k\right)\right)=\frac{\sum_{l=1}^{h} w_l \prod_{i=1}^{n} \mu_{li}\left(x_i\left(k\right)\right)}{\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}\left(x_i\left(k\right)\right)}=\sum_{l=1}^{h} w_l \frac{\prod_{i=1}^{n} \mu_{li}\left(x_i\left(k\right)\right)}{\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}\left(x_i\left(k\right)\right)}=\sum_{l=1}^{h} w_l \varphi_l\left(x\left(k\right)\right)=w^{hT}\varphi^h\left(x\left(k\right)\right),$$

where

$$\varphi_l\left(x\left(k\right)\right)=\frac{\prod_{i=1}^{n} \mu_{li}\left(x_i\left(k\right)\right)}{\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}\left(x_i\left(k\right)\right)},\ w^h=\left(w_1,w_2,\ldots,w_2\right)^T,$$

$$\varphi^h\left(x\left(k\right)\right)=\left(\varphi_1\left(x\left(k\right)\right),\varphi_2\left(x\left(k\right)\right),\ldots,\varphi_h\left(x\left(k\right)\right)\right)^T.$$

It is easy to notice that the system under consideration realizes nonlinear mapping of the input space into a scalar output signal $\left(R^n \to R^1\right)$, which is similar to a normalized RBFN (Haykin, 1999), and in architecture (at a fixed *h*) coincides with a TSK system (Takagi, Sugeno, Kang) of zero-order, that is called Wang-Mendel architecture (Nelles, 2001).

Development of an algorithm for artificial neural networks teaching for intelligent decision support systems

In order to understand the relationship of the proposed methodology in the process of ANN operation, the authors constructed a typical algorithm for the operation of the ANN with the methodology of ANN training (Figure 2). The initial stage is the input of the initial data, namely the initial architecture and parameters of the artificial neural network (step 1-4).

Step 1. Determine the number of layers in an artificial neural network.

Step 2. Determine the number of nodes in the layer of the artificial neural network.

Step 3. Determine the number of connections between layers and nodes in an artificial neural network.

Step 4. Determine the number of hidden layers of an artificial neural network.

Step 5. Determine the number and parameters of the membership function (MF).

Step 6: Determine the synaptic communication weights of the artificial neural network.

Step 7. Check the compliance of the architecture and the parameters of the artificial neural network with the requirements.

Step 8. Accepted decisions on the correction of the architecture and parameters of the artificial neural network.

Consider in detail the steps of the proposed teaching methodology.

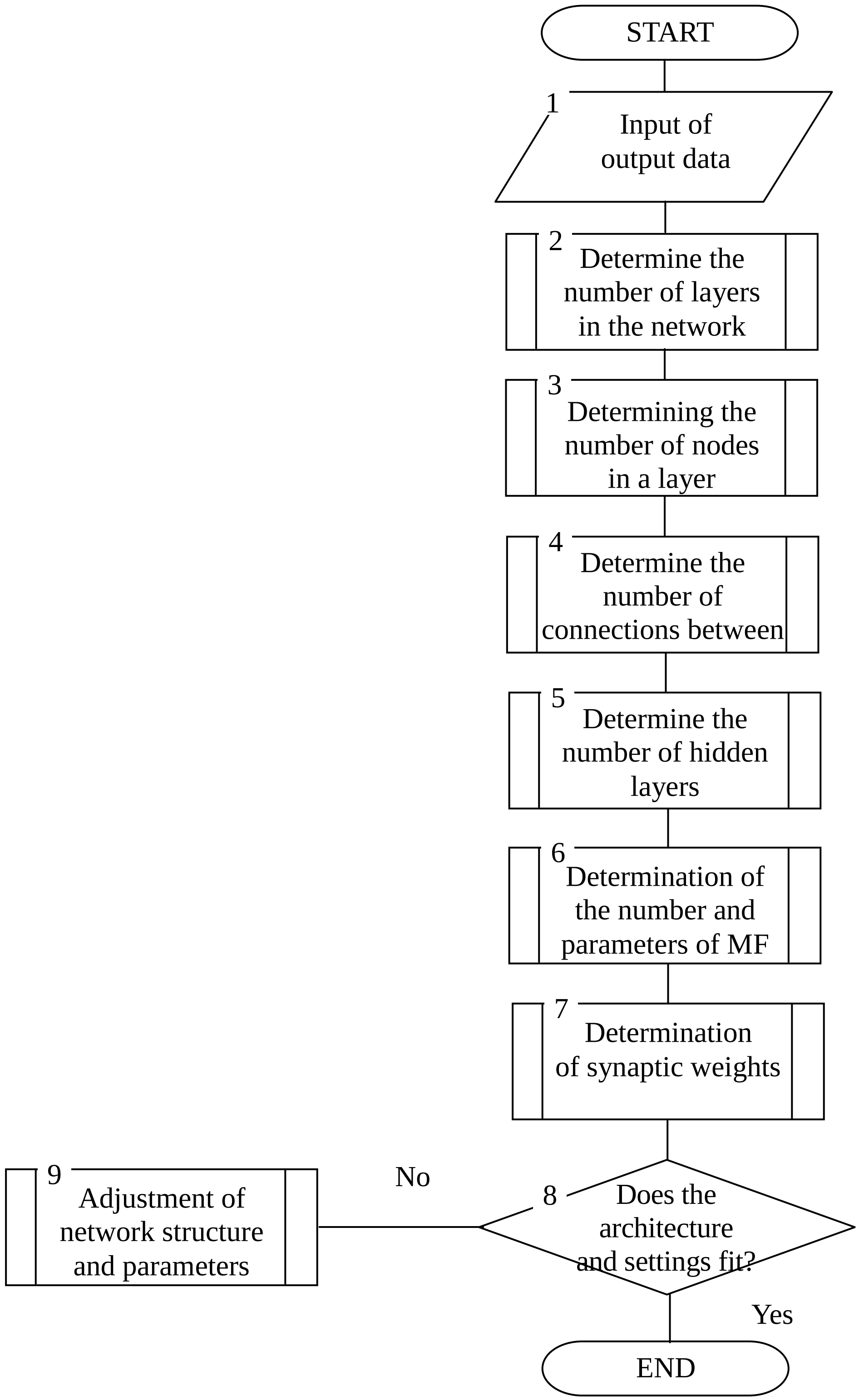

Figure 2. The algorithm of functioning and training of an evolving artificial neural network

Step 5. Configure parameters and the number of membership functions.

The process of setting parameters and the number of membership functions is as follows. Let the first observation of the training sample be sent to the input of the system, which in the initial state in the first hidden layer lacks membership functions $x(1)=\left(x_1(1), x_2(1), \dots, x_n(1)\right)^{\mathrm{T}}$.

This observation forms the first node of the first hidden layer μ_1 so that $c_{1i}(1)=x_i(1)$. In this way, *n* membership functions are created and a single synaptic weight $w_1(0)$ is formed, which is given randomly in the interval $-1 \le w_1(0) \le 1$.

The neighborhood radius *r*, given by the maximum possible number of the membership function μ_1 in NFS *h*, is then given for this membership function. If the axis membership functions are evenly distributed, then

$$r=\frac{2}{h-1}. \qquad (2)$$

Then, when you receive the second observation $x(2)$, the condition should be checked

$$\max_i \left|c_{1i}-x_i(2)\right| \le r. \qquad (3)$$

If this condition is met, then the μ_1 node ownership function centers are corrected according to the rule

$$c_{1i}(2)=c_{1i}(1)+\eta(2)\left(x_i(2)-c_{1i}(1)\right), \qquad (4)$$

where η is the parameter of the learning step.

Thus, when $\eta(2)=0,5$

$$c_{1i}(2)=\frac{c_{1i}(1)+x_i(2)}{2}. \qquad (5)$$

If condition (3) is not fulfilled, then the second node μ_2 of the membership functions of the first hidden layer is formed, the centers of which

$$c_{2i}(2)=x_i(2). \qquad (6)$$

At the same time with the μ_2 node, a second synaptic weight w_2 is introduced into the NFS, which is also randomly assigned. Let the p nodes of the membership function $\mu_1, \mu_2, ..., \mu_p$, $p < h$ be generated by the time the NFS input is received, and the condition centers $c_{li}(k-1), l = 1, 2, ..., p; i = 1, 2, ..., n.$ Upon $x(k)$ receipt, the condition should be checked

$$\max_i \left| c_{li} - x_i(k) \right| \leq r \forall l = 1, 2, ..., p. \qquad (7)$$

If this condition is met, then the correction centers of the membership functions closest to the corresponding components $x(k)$ are corrected according to the rule

$$c_{li}(k) = c_{li}(k-1) + \eta(k)\left(x_i(k) - c_{li}(k-1)\right). \ (8)$$

It is easy to see that (8) is nothing but the rule of T. Kohonen's self-study “The winner receives everything” (Wang and Mendel, 1993) with the only difference that Kohonen's self-study is realized in the hypersphere

$$\left\| x(k) \right\|_2 = 1, \qquad (9)$$

and rule (8) is in the hypercube

$$\left\| x(k) \right\|_\infty = 1. \qquad (10)$$

If condition (7) is not fulfilled, the $(p+1)$ node $(p+1 \leq h)$ with centers of membership functions will be formed in the system.

$$c_{p+1,i}(k) = x_i(k). \qquad (11)$$

At the same time with the node μ_{p+1}, synaptic weight w_{p+1} is formed.

As can be seen, this procedure is a hybrid algorithm evolving by N. Kasabov (Kasabov, 2003) and self-organized maps by T. Kohonen (Kohonen, 1995). In this process, the evolution of the self-learning architecture of membership functions can occur continuously and until the number of membership functions is reached.

Adjustment of the center parameters and the width of the membership functions can be done in an algorithm with a teacher based on the minimization of the objective function. The customization is usually set using the Euclidean norm and for one pair of training data $(x(k), y(k))$ that looks like

$$E(k)=\frac{1}{2}\left(y(k)-\hat{y}(k)\right)^2=\frac{1}{2}\left(y(k)-w^{\mathrm{T}}\varphi\left(x(k)\right)\right)^2. \tag{12}$$

While applying the fastest descent method, the corresponding adaptation formulas in the general case for the dimensional vector $(n\times 1)$ of input signals take the form

$$c_{rj}(k+1)=c_{rj}(k)-\eta_c\frac{\partial E(k)}{\partial c_{rj}}, \tag{13}$$

$$\sigma_{rj}(k+1)=\sigma_{rj}(k)-\eta_\sigma\frac{\partial E(k)}{\partial \sigma_{rj}}, \tag{14}$$

where η_c is the learning step parameter for the parameters of the membership function center; η_σ is the learning step parameter for the width of the membership function;

$$r=1,2,\ldots,h,$$
$$j=1,2,\ldots,n.$$

To simplify the calculation of derivatives and speed up the calculation of the value of the membership function, the formula for the adaptation of the width parameter can be written in the form

$$-0.5\sigma_{rj}^{-2}(k+1)=\sigma_{rj}(k)-\eta_\sigma\frac{\partial E(k)}{\partial\left(-0.5\sigma_{rj}^{-2}\right)}. \tag{15}$$

When traditional Gaussians are used as a function of belonging (9), the corresponding gradient formulas of the objective function (12) for one pair of training data takes the form (Sugeno and Kang, 1998)

$$\frac{\partial E(k)}{\partial c_{rj}}=\left(w^{\mathrm{T}}\varphi\left(x(k)\right)-\left(x(k)\right)\right)\sum_{l=1}^{h}w_l\frac{\partial\varphi_l\left(x(k)\right)}{\varphi c_{rj}}, \tag{16}$$

$$\frac{\partial E(k)}{\partial\left(-0.5\sigma_{rj}^{-2}\right)}=\left(w^{\mathrm{T}}\varphi\left(x(k)\right)-y\left(x(k)\right)\right)\sum_{l=1}^{h}w_l\frac{\partial\varphi_l\left(x(k)\right)}{\partial\left(-0.5\sigma_{rj}^{-2}\right)}, \tag{17}$$

where

$$\varphi_l\left(x(k)\right)=\frac{\prod_{i=1}^{n}\mu_{li}\left(x_i(k)\right)}{\sum_{p=}^{h}\prod_{i=1}^{n}\mu_{pi}\left(x_i(k)\right)}. \qquad (18)$$

Derivatives

$\frac{\partial\varphi_l(x)}{\partial c_{rj}}$ and $\frac{\partial\varphi_l(x)}{\partial\left(-0.5\sigma_{rj}^{-2}\right)}$, which are defined on the basis of (9) and (18) can be written as

$$\frac{\partial\varphi_l(x)}{\partial c_{rj}}=\frac{\delta_{lr}m(x)-t_l(x)}{\left(m(x)\right)^2}\prod_{i=1,i\neq j}^{n}\mu_{ri}(x_i)\frac{\partial\mu_{ri}(x_j)}{\partial c_{rj}}, \qquad (19)$$

$$\frac{\partial\varphi_l(x)}{\partial\left(-0.5\sigma_{rj}^{-2}\right)}=\frac{\delta_{lr}m(x)-t_l(x)}{\left(m(x)\right)^2}\prod_{i=1,i\neq j}^{n}\mu_{ri}(x_i)\frac{\partial\mu_{ri}(x_j)}{\partial\left(-0.5\sigma_{rj}^{-2}\right)}, \qquad (20)$$

where δ_{lr} is the delta of Kronecker,

$$t_l(x)=\prod_{i=1}^{n}\mu_{li}(x_i),$$

$$m(x)=\sum_{p=1}^{h}\prod_{i=1}^{n}\mu_{pi}(x_i).$$

Derivatives $\frac{\partial\mu_{rj}(x_j)}{\partial c_{rj}}$ and $\frac{\partial\mu_{rj}(x_j)}{\partial\left(-0.5\sigma_{rj}^{-2}\right)}$, which are defined on the basis of (9) can be written as

$$\frac{\partial\mu_{rj}(x_j)}{\partial c_{rj}}=\frac{x_j-c_{rj}}{\sigma_{rj}^2}\exp\left(-\frac{\left(x_j-c_{rj}\right)^2}{2\sigma_{rj}^2}\right), \qquad (21)$$

$$\frac{\partial\mu_{ri}(x_j)}{\partial\left(-0.5\sigma_{rj}^{-2}\right)}=\left(x_j-c_{rj}\right)^2\exp\left(-\frac{\left(x_j-c_{rj}\right)^2}{2\sigma_{rj}^2}\right). \qquad (22)$$

Step 6: Determine the synaptic communication weights of the artificial neural network.

As it was noted, well-known algorithms of training-identification can be used to adjust the synaptic weights of the neuro-phase system.

(Ljung, 1987):

– exponentially weighted recurrent least squares method (Otto, 2003);

$$\begin{cases} w^h(k) = w^h(k-1) + \dfrac{P^h(k-1)\left(y(k) - w^{hT}(k-1)\varphi^h(x(k))\right)}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))}\varphi^h(x(k)) = \\ = w^h(k-1) + \dfrac{P^h(k-1)\left(y(k) - \hat{y}^h(x(k))\right)}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))}\varphi^h(x(k)), \\ P^h(k) = \dfrac{1}{\beta}\left(P^h(k-1) - \dfrac{P^h(k-1)\varphi^h(x(k))\varphi^{hT}(x(k))P^h(k-1)}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))}\right) = \\ = \left(\sum_{r=1}^{k}\varphi^h(x(\tau))\varphi^{hT}x(\tau)\right)^{-1}, 0 < \beta \le 1, \end{cases} \tag{23}$$

where $y(t)$ is the external training signal,

β is the parameter for outdated information forgetting.

– optimum speed one-step gradient Kachmage–Widrow–Hoff algorithm

$$w^h(k) = w^h(k-1) + \frac{y(k) - w^{hT}(k-1)\varphi^h(x(k))}{\left\|\varphi^h(x(k))\right\|^2}\varphi^h(x(k)); \tag{24}$$

– a learning algorithm that has tracking and smoothing properties (Narendra and Parthasarathy, 1990)

$$\begin{cases} w^h(k) = w^h(k-1) + \left(\beta^h(k)\right)^{-1}\left(y(k) - w^{hT}(k-1)\varphi^h(x(k))\right)\varphi^h(x(k)), \\ \beta^h(k) = \beta\beta^h(k-1) + \left\|\varphi^h(x(k))\right\|^2, 0 \le \beta \le 1 \end{cases} \tag{25}$$

and similar procedures. Procedure (25) is related to the algorithm (23) by the relation,

$$\beta^h(k) = TrP^h(k), \tag{26}$$

when $\beta = 0$, we get the form of the algorithm (23).

The process of tuning synaptic scales can occur at the same time as the self-learning of the first hidden layer. Let the p nodes of the membership functions $\mu_1, \mu_2, ..., \mu_p$ be formed by the time of the observation and calculate the synaptic weight vector $w^p(k-1)$. Let condition (7) not be fulfilled, which immediately leads to the formation of the node μ_{p+1} and the arbitrary initial value of the synaptic weight w_{p+1}.

The output signal NFS can be represented in the form

$$\hat{y}^{p+1}\left(x(k)\right)=\left(w^{p+1}(k-1)\right)^{\mathrm{T}}\varphi^{p+1}\left(x(k)\right)=w^{p\mathrm{T}}(k-1)\varphi^{p}\left(x(k)\right)+w_{p+1}\varphi_{p+1}\left(x(k)\right), \tag{27}$$

but the algorithm (24) has the form

$$\begin{cases} w^{p+1}(k)=\left(\dfrac{w^{p}(k-1)}{w_{p+1}}\right)+\left(\beta^{p+1}(k)\right)^{-1}\left(y(k)-\hat{y}^{p+1}\left(x(k)\right)\right)\left(\dfrac{\varphi\left(x(k)\right)}{\varphi_{p+1}\left(x(k)\right)}\right), \\ \beta^{p+1}(k)=\beta\beta^{p}(k-1)+\left\|\varphi^{p}\left(x(k)\right)\right\|^{2}+\varphi_{p+1}^{2}\left(x(k)\right). \end{cases} \tag{28}$$

As you can see, the process of simultaneous evolution-self-study-controlled learning does not cause any computational problems.

Step 7. Check the compliance of the architecture and the parameters of the artificial neural network with the requirements.

At this stage, the compliance of the architecture and parameters of the artificial neural network with the requirements is checked. These requirements for the functioning of an artificial neural network are made at the design stage of decision support systems and depend on the type and scope of tasks that are performed by decision support systems.

Step 8. Accepted decisions on the correction of the architecture and parameters of the artificial neural network.

Based on the comparative assessment of the requirements for the functioning of the ANN and their real efficiency, a decision is made to adjust the parameters of the ANN, namely (action 6-8 in the scheme of Figure 2):

- the type and parameters of the membership function; and
- synaptic weights between the links.

In the event that it is impossible to meet the necessary requirements that are imposed on the ANN, a decision is made to change the architecture of the ANN and determine the initial parameters of the ANN (action 1-5 in the scheme of Figure 2).

Discussion

The technique of artificial neural networks training for intelligent decision support systems is proposed. The work of the proposed methodology in the MathCad 14 software environment is simulated. The effectiveness of the proposed multilayer neuro-phase system evolving with hybrid learning has been demonstrated in the evaluation of a radio frequency pseudorandom tuning (RFPT) system under the influence of intentional noise interference. The simulations were performed with the following parameters:

- – radiocommunication with RFPT: frequency range 30-512 MHz; transmitter power is 10 watts; transmitted bandwidth is 12.5 kHz, receiver sensitivity is 110 dB; the number of RCD in the network is 4; the number of frequency channels for tuning is 10000; the number of adjustments is 333, 5 jumps/sec;
- – radio-electronic suppression (RES) complex: frequency range is 30-2000 MHz; transmitter power is 2000 watts; the maximum suppression bandwidth is 80 MHz; the number of radio-lines with RFPT that can be suppressed simultaneously is 4, type of interference is noise interference with frequency manipulation; the strategy of the RES complex is dynamic.

To evaluate the impact of deliberate noise interference on the RFPT radio communication facilities, a neuro-phase system with a number of inputs was used $n=5$. System training was conducted on a sample basis, with control signals:

$$f\left(x_1,x_2,x_4,x_5\right)=\frac{x_1x_2x_4x_5\left(x_3-1\right)+x_4}{1+x_3^2+x_2^2}. \quad (29)$$

The number of iterations for estimating the state of the radio communication system with the RFPT is 10. To test the effectiveness of the proposed method, suppose that the operation time of the programmed radio communication with the RFPT at one frequency is the same as the RES complex (Alieinykov et al, 2019).

Briefly describe the graphical relationships obtained in Figure 3. As can be seen from Figure 3 that the best estimate of the bit error ratio of the signal to noise ratio is the method proposed by the authors. Figure 3 and 4 were obtained by averaging the signal-to-noise ratio on the frequency subchannels. This advantage is due to the greater accuracy of the signal-to-noise ratio in the frequency subchannels due to the rapid tuning of the parameters of the artificial neural network to the type and intensity of the data entering its input. At the same time, it would be desirable to emphasize separately that the estimation by the proposed method does not increase the estimation error during the operation of an artificial neural network unlike other approaches (Figure 3 and 4).

Figure 3 shows dependence of bit error probability on signal-to-noise ratio for different methods under the influence of fluctuation noise and noise barrier with overlap factor ρ=1 at 10 iterations of artificial neural network estimation. Figure 4 shows a plot of the bit error probability on the signal-to-noise ratio (for the case of noise interference in the band with frequency manipulation ρ=0,5).

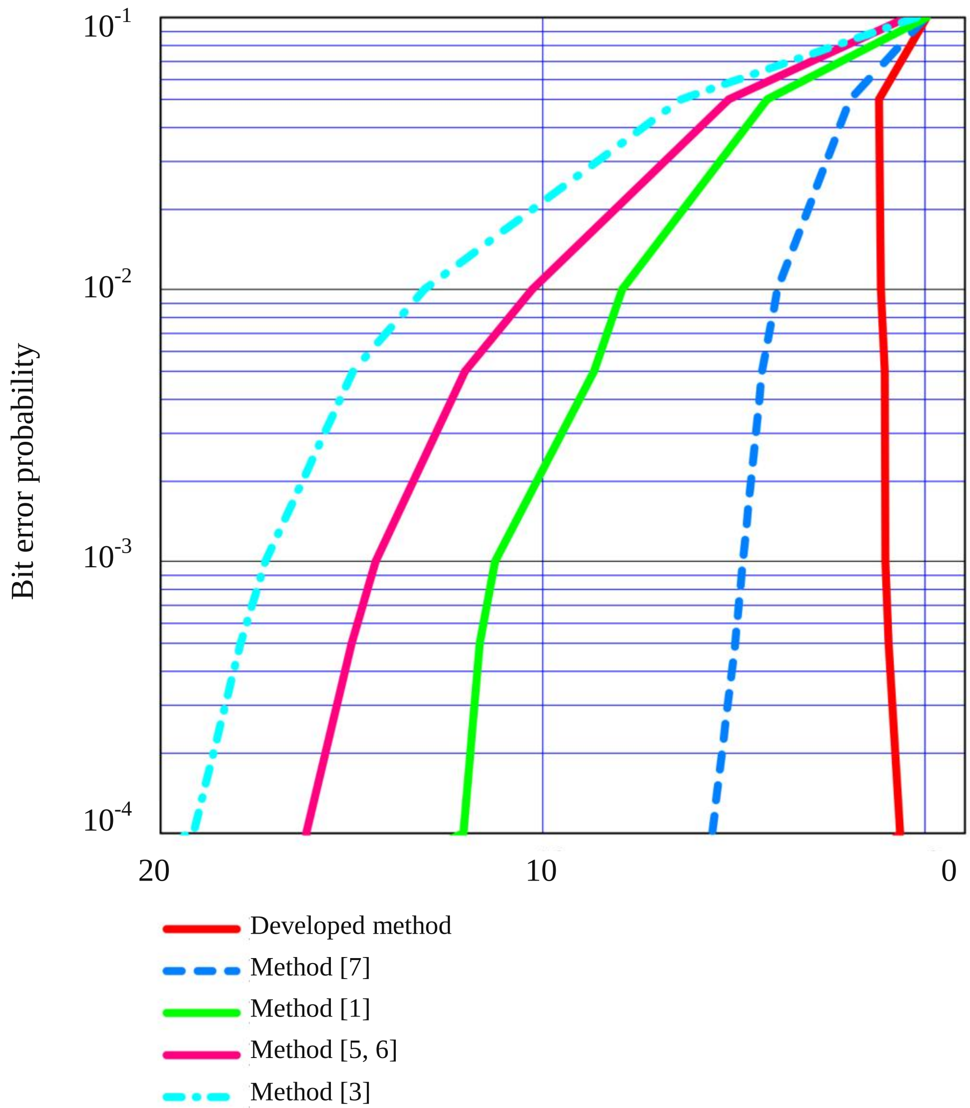

Figure 3. Dependence of bit error probability on signal-to-noise ratio for different methods under the influence of fluctuation noise and noise barrier

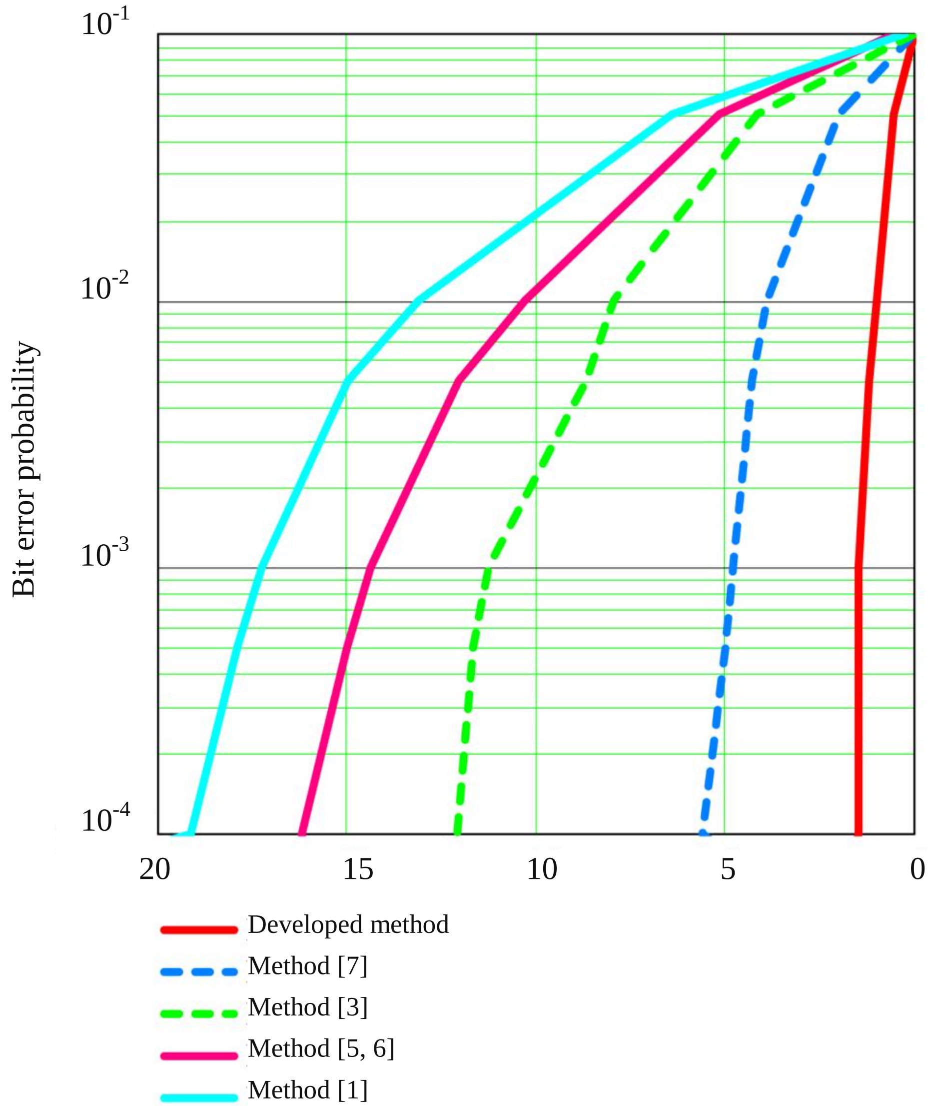

Figure 4. Graph of bit error probability versus signal-to-noise ratio (for the case of noise interference in the part of frequency band with overlap factor ρ=0.5) at 10 iterations of artificial neural network estimation

An evaluation of the computational complexity of the implementation of the developed methodology showed that for the given output data and when using the ADSP-21261 processor, the status of the radio system channel can be performed in real-time. This takes into account the delay required to transmit information about these values via the feedback service channel. The research of the developed methodology showed that the mentioned training method provides on average 16-23 percent higher efficiency of training of artificial neural networks and does not accumulate errors during training. The main advantages of the proposed evaluation methodology are:

– does not accumulate learning errors during the training of artificial neural networks by adjusting the parameters and architecture of the artificial neural network;
– the uniqueness of the obtained results;
– wide scope (support and decision-making systems);
– the simplicity of mathematical calculations;
– the ability to adapt the system during operation;
– ability to synthesize the optimal structure of the support and decision-making system.

The disadvantages of the proposed methodology include:

- loss of information in the estimation (forecasting) due to the construction of the membership function. This loss of information can be reduced by choosing the type of membership function and its parameters in the practical implementation of the proposed methodology in the systems of support and decision making. Choosing the type of membership function depends on the computing resources of a particular electronic computing device;
- lower accuracy of estimation on a separate parameter of a state estimation; and
- loss of accuracy of results during the restructuring of the artificial neural network architecture.

This method will allow:
– to train artificial neural networks;
– to identify effective measures to improve the performance of artificial neural networks;
– to increase the efficiency of artificial neural networks by learning the parameters and architecture of the networks;
– to reduce the use of computing resources for support and decision-making systems;
– to develop measures, that are aimed at increasing the efficiency of training of artificial neural networks;
– to increase the speed of processing information in artificial neural networks.
The areas of further research should be aimed at reducing the computational cost of processing different types of data in special-purpose systems.

Conclusions

1. The scientific novelty of this technique of training artificial neural networks for intelligent decision support systems is as follows.
   – conducts training not only the synaptic weights of the artificial neural network but also the type and parameters of the membership function;
   – in case, when it is impossible to provide the specified quality of functioning of artificial neural networks due to the learning of parameters, the architecture of artificial neural networks is trained;

– the choice of architecture, type and parameters of the membership function takes into account the computing resources of the tool and type, amount of information, which is supplied to the input of an artificial neural network;
– there is no accumulation of the error of learning artificial neural networks as a result of processing information entering the input of artificial neural networks.

2. The practical value of the proposed methodology is that it is based on the development of practical recommendations for improving the effectiveness of training artificial neural networks.

The research of the developed methodology showed that the mentioned training method provides on average 16-23 percent higher efficiency of training of artificial neural networks and does not accumulate errors during training. Therefore, the purpose of the article is to develop a methodology for teaching artificial neural networks for intelligent decision support systems, with the uniqueness of the decisions that we consider to be achieved. The direction of further research should be the development of advanced training techniques for artificial neural networks for intelligent decision support systems.

References

Abaci, K., & Yamacli V. (2019). Hybrid artificial neural network by using differential search algorithm for solving power flow problem. *Advances in Electrical and Computer Engineering.* 19(4), 57–64. https://doi.org/10.4316/AECE.2019.04007.

Alieinykov, I., Thamer, K. A., Zhuravskyi, Y., Sova, O., Smirnova, N., Zhyvotovskyi, R., Hatsenko, S., Petruk, S., Pikul, R., & Shyshatskyi, A.(2019). Development of a method of fuzzy evaluation of information and analytical support of strategic management. *Eastern-European Journal of Enterprise Technologies*, 6(2), 16–27. https://doi.org/10.15587/1729-4061.2019.184394.

Bodyanskiy, Y., Pliss, I., & Vynokurova, O. (2013). Flexible neo-fuzzy neuron and neuro-fuzzy network for monitoring time series properties. *Information Technology and Management Science*. 16, 47–52.

Bodyanskiy, Ye., Pliss, I., & Vynokurova, O. (2013). Flexible wavelet-neuro-fuzzy neuron in dynamic data mining tasks. *Oil and Gas Power Engineering*. 2 (20), 158–162.

Çavdar, A. B., & Ferhatosmanoğlu, N. (2018). “Airline customer lifetime value estimation using data analytics supported by social network information.” *Journal of Air Transport Management*. 67, 19–33. https://doi.org/10.1016/j.jairtraman.2017.10.007.

Chica, J. F., Zaputt, S., Encalada, J., Salamea, C., & Montalvo, M.(2019). Objective assessment of skin repigmentation using a multilayer perceptron. *Journal of Medical Signals and Sensors.* 9(2), 88–99.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, N.J.: Prentice Hall, Inc., 842 p

Kachayeva, G. I., & Mustafayev, A. G. (2018). The use of neural networks for the automatic analysis of electrocardiograms in diagnosis of cardiovascular diseases. *Herald of Dagestan State Technical University. Technical Sciences*. 45(2), 114-124. https://doi.org/10.21822/2073-6185-2018-45-2-114-124

Kalantaievska, S., Pievtsov, H., Kuvshynov, O., Shyshatskyi, A., Yarosh, S., Gatsenko, S., Zubrytskyi, H., Zhyvotovskyi, R., Petruk, S., & Zuiko, V. (2018). Method of integral estimation of channel state in the multiantenna radio communication systems. *Eastern-European Journal of Enterprise Technologies*, 5(9), 60–76. https://doi.org/10.15587/1729-4061.2018.144085.

Kanev, A., Nasteka, A., Bessonova, C., Nevmerzhitsky, D., Silaev, A., Efremov, A., & Nikiforova, K. (2017). Anomaly detection in wireless sensor network of the “smart home” system. *Proceeding of the 20th conference of fruct association*. 776 (20), 118–124. https://doi.org/10.23919/FRUCT.2017.8071301

Kasabov, N. (2003). *Evolving Connectionist Systems*. London: Springer. Verlag.

Katranzhy, L., Podskrebko, O., & Krasko, V. (2018). Modelling the dynamics of the adequacy of bank’s regulatory capital. *Baltic Journal of Economic Studies*. 4(1), 188-194. https://doi.org/10.30525/2256-0742/2018-4-1-188-194

Kazemi, M., & Faezirad, M.(2018). Efficiency estimation using nonlinear influences of time lags in DEA Using Artificial Neural Networks. *Industrial Management Journal*, 10(1), 17–34.

Kohonen, T. (1995). *Self-Organizing Maps*. Berlin: Springer-Verlag, 362 p.

Kuchuk, N., Mohammed, A. S., Shyshatskyi, A., & Nalapko O.(2019). The method of improving the efficiency of routes selection in networks of connection with the possibility of self-organization. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(1), 1–6. https://doi.org/10.30534/ijatcse/2019/0181.22019.

Ljung, L. (1987). *System identification: theory for the user.* Upper Saddle River: Prentice Hall, Inc., 432 p.

Manea, E., Carloa, D. Di., Depellegrin, D., Agardy, T., & Gissi E.(2019). Multidimensional assessment of supporting ecosystem services for marine spatial planning of the Adriatic Sea. *Ecological Indicators*. 101, 821–837. https://doi.org/10.1016/j.ecolind.2018.12.017.

Massel, L.V., Gerget, O. M., Massel, A. G., & Mamedov, T. G. (2019). The use of machine learning in situational management in relation to the tasks of the power industry. *EPJ Web of Conferences.* 217, 1–6. https://doi.org/10.1051/epjconf/201921701010.

Mishchuk, O. S., & Vitynskyi, P. B.(2018). Neural network with combined approximation of the surface of the response. *Research Bulletin of the National Technical University of Ukraine "Kyiv Polytechnic Institute"*. 2, 18–24. https://doi.org/10.20535/1810-0546.2018.2.129022.

Narendra, K. S., & Parthasarathy, K.(1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1), 4-26.

Nelles, O. (2001). *Nonlinear System Identification*. Berlin: Springer.

Otto, P. (2003). A new learning algorithm for a forecasting neuro-fuzzy network. *Integrated Computer-Aided Engineering*, 10 (4), 399–409.

Parapuram, G., Mokhtari, M., & Hmida, J. B. (2018). An artificially intelligent technique to generate synthetic geomechanical well logs for the Bakken formation. *Energies*, 11 (3), 680. https://doi.org/10.3390/en11030680

Prokoptsev, N. G., Alekseenko, A. E., & Kholodov Y. A.(2018). Traffic flow speed prediction on transportation graph with convolutional neural networks. *Computer Research and Modeling*. 10(3), 359–367. https://doi.org/10.20537/2076-7633-2018-10-3-359-367.

Sreeshakthy, M., & Preethi J. (2016). Classification of human emotion from deap EEG signal using hybrid improved neural networks with Cuckoo search. *Brain: Broad Research in Artificial Intelligence and Neuroscience*, (6)3–4, 60–73.

Sugeno, M., & Kang, G. T. (1998). Structure identification of fuzzy model. *Fuzzy Sets and Systems*. 28, 15–33.

Wang, L.-X., & Mendel, J. M.(1993). Fuzzy basis functions, universal approximation and orthogonal least squares learning. *IEEE Trans. on Neural Networks*. 3, 807-814.

Zhang, J., & Ding, W.(2017). Prediction of air pollutants concentration based on an extreme learning machine: The Case of Hong Kong. *International Journal of Environmental Research and Public Health*. 14(2), 1–19. https://doi.org/10.3390/ijerph14020114.

Zhdanov, V. V. (2016). Experimental method to predict avalanches based on neural networks. *Lëd i Sneg*, 56(4), 502–510. https://doi.org/ 10.15356/2076-6734-2016-4-502-510.

Bibliographic information of this paper for citing:

Thamer, K. A. (2020). "Method of artificial neural networks teaching." *Webology*, 17(1), Article 207. Available at: http://www.webology.org/2020/v17n1/a207.pdf