

Розроблено алгоритм навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень. Відмінна особливість запропонованого алгоритму полягає в тому, що він проводить навчання не тільки синаптичних ваг штучної нейронної мережі, але й виду та параметрів функції належності. В разі неможливості забезпечити задану якість функціонування штучних нейронних мереж за рахунок навчання параметрів штучної нейронної мережі відбувається навчання архітектури штучних нейронних мереж. Вибір архітектури, виду та параметрів функції належності відбувається з врахуванням обчислювальних ресурсів засобу та з врахуванням типу та кількості інформації, що надходить на вхід штучної нейронної мережі. Також при використанні запропонованого алгоритму не відбувається накопичення помилок навчання штучних нейронних мереж в результаті обробки інформації, що надходить на вхід штучних нейронних мереж.

Розробка запропонованого алгоритму обумовлена необхідністю проведення навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень, з метою обробки більшої кількості інформації, при однозначності рішень, що приймаються. За результатами дослідження встановлено, що зазначений алгоритм навчання забезпечує в середньому на 16–23 % більшу високу ефективність навчання штучних нейронних мереж та не накопичує помилок в ході навчання. Зазначений алгоритм дозволить проводити навчання штучних нейронних мереж; визначити ефективні заходи для підвищення ефективності функціонування штучних нейронних мереж. Також розроблений алгоритм дозволить підвищити ефективність функціонування штучних нейронних мереж за рахунок навчання параметрів та архітектури штучних нейронних мереж. Запропонований алгоритм зменшує використання обчислювальних ресурсів систем підтримки та прийняття рішень. Використання розробленого алгоритму дозволить виробити заходи, що спрямовані на підвищення ефективності навчання штучних нейронних мереж, та підвищити оперативність обробки інформації

Ключові слова: штучні нейронні мережі, синаптичні ваги, функція належності, обробка інформації, інтелектуальні системи підтримки прийняття рішень

UDC 681.324.01

DOI: 10.15587/1729-4061.2020.192711

DEVELOPMENT OF AN ALGORITHM TO TRAIN ARTIFICIAL NEURAL NETWORKS FOR INTELLIGENT DECISION SUPPORT SYSTEMS

O. Sova

Doctor of Technical Sciences, Senior Research, Head of Department
Department of Automated Control Systems*

O. Turinskyi

PhD, Head of University
Ivan Kozhedub Kharkiv University of Air Force
Sumska st., 77/79, Kharkiv, Ukraine, 61023

A. Shyshatskyi

PhD, Senior Research
Research Department of Electronic Warfare Development **
E-mail: ierikon12@gmail.com

V. Dudnyk

PhD
Department of Fire Training
Hetman Petro Sahaidachnyi National Army Academy
Heroiv Maidanu str., 32, Lviv, Ukraine, 79026

R. Zhyvotovskiy

PhD, Senior Researcher, Head of Department
Research Department of the Development of
Anti-Aircraft Missile Systems and Complexes**

Ye. Prokopenko

PhD
Institute of Information Technology***

T. Hurskyi

PhD, Associate Professor*

V. Hordiichuk

PhD, Head of Department
Scientific and Organizational Department
Institute of Naval Forces National University "Odessa Maritime Academy"
Hradonachalnytska str., 20, Odessa, Ukraine, 65029

A. Nikitenko

Adjunct
Institute of Public Military Management***

A. Remez

Adjunct
Institute of Public Military Management***
*Military Institute of Telecommunications and
Informatization named after Heroes of Kruty
Moskovska str., 45/1, Kyiv, Ukraine, 01011
**Central Scientifically-Research Institute of Arming
and Military Equipment of the Armed Forces of Ukraine
Povitroflorskyi ave., 28, Kyiv, Ukraine, 03168
***Ivan Chernyakhovsky National Defense University of Ukraine
Povitroflorskyi ave., 28, Kyiv, Ukraine, 03168

Received date 11.12.2019

Accepted date 06.02.2020

Published date 29.02.2020

Copyright © 2020, O. Sova, O. Turinskyi, A. Shyshatskyi, V. Dudnyk,

R. Zhyvotovskiy, Ye. Prokopenko, T. Hurskyi, V. Hordiichuk, A. Nikitenko, A. Remez

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0>)

1. Introduction

Decision support systems (DSS) are actively used in all spheres of human life. They became widely spread when pro-

cessing large data arrays, providing the informational support of the process of making decisions by decision-makers.

The creation of intelligent DSS has become a natural continuation of a wide application of the classical type of

DSS. Intelligent DSS provide informational support of all production processes and services of enterprises (organizations, institutions). Intelligent DSS are used in designing, manufacturing and sales of products, in financial and economic analysis, planning, personnel management, marketing, support of products creation (operation, repair) and perspective planning. These intelligent DSS have been widely used for the fulfillment of specific military tasks, specifically [1, 2]:

- planning of deployment, operation of communication systems and data transmission;
- automation of army and weapon control;
- gathering, processing, and generalizing the information of intelligence about reconnaissance objects, etc.

The structure of intelligent DSS can be conditionally divided into 4 large layers:

- the interface layer (interactivity and visualization);
- the modeling layer (statistical models and machine training; numerical models; models based on game theory, etc.);
- the data processing layer (data flow organization, work with databases and expert estimations);
- the data collection layer (web scanning, sensors, and programming interface).

One of the ways to increase the effectiveness of decision support systems is to introduce artificial neural networks at all levels of their operation.

2. Literature review and problem statement

Analysis of the experience of creating intelligent DSS shows that the most promising for the construction is the information technology based on neural network modeling [1–8], including the application of the evolutionary approach to ANN construction [4, 5], which are used on each of the layers of intelligent DSS.

The application of the evolutionary approach to the construction of neural networks in comparison with traditional approaches provides the following benefits:

- capability to adapt quickly to the domain, which virtually without any transformations makes it possible to form the ANN structure that complies with this process;
- capability to learn quickly based on the models of neurons with appropriate thresholds, weights, and transmission functions, in which the trained ANN will be constructed in first approximation;
- capability to operate under conditions of uncertainty, non-linearity, in stochastic and chaotic conditions and under conditions of various disturbances and interferences;
- have both universal approximating properties, and possibilities of fuzzy output.

Evolving ANN became widely used to solve various problems of intelligent data analysis, identification, emulation, forecasting, intelligent management, etc., on each of the layers of intelligent DSS.

Despite their rather successful application for solving a wide range of problems of intelligent data analysis, these systems have a series of shortcomings associated with their use.

The following shortcomings can be separated among the most essential ones:

- 1) complexity of choosing the architecture of a system. As a rule, the model based on the principles of computational intelligence has the fixed architecture. In the ANN context it

means that a neural network has a fixed number of neurons and relations. In this regard, it may turn out to be problematic for a system to adapt to the new data coming to be processed, which are different from the preceding data in nature;

- 2) training in the packet mode and training over several epochs requires considerable time resources. Such systems are not adapted for the online operation mode with a sufficiently high tempo of new data coming to be processed;

- 3) many of existing systems of computation intelligence cannot determine the evolving rules, according to which the system develops, and can present the results of their operation in the terms of a natural language.

Thus, the task of developing new training algorithms for the ANN that will make it possible to solve these problems is relevant.

The analysis of scientific research that was carried out [1–10] reveals that the methods of artificial intelligence are currently the basis for the existing DSS.

Paper [3] analyses the properties of the ANN, which were used in prediction of air pollutants concentration. The paper focused on the fact that the ANN have a low convergence rate and a local minimum. It was proposed to use an extreme training machine for the ANN, which provides high efficiency of generalization at extremely high training rate. The shortcomings of this approach include accumulation of the ANN errors in the course of calculations, impossibility to select parameters and type of membership function.

Paper [5] provides an operative approach from spatial analysis in the maritime area for quantitative assessment and displaying of related ecosystem services. This approach encompasses three-dimensional marine environment, examining separately all marine areas (sea surface, water pillar and seabed). In fact, the method constructs 3-dimensional models of the sea by estimating and mapping, which are associated with each of the three sea domains through adopting representative indicators. It should be considered that the shortcomings of this method include the impossibility of flexible adjustment (adaptation) of evaluation models at adding (exclusion) indicators and changing their parameters (compatibility and significance of indicators).

Article [6] presents a machine learning model for the automatic identification of requests and providing information support services that are exchanged between the members of the Internet community. The specified model is designed to process a large number of messages of social media users. The disadvantages of this model are the lack of mechanisms to assess the adequacy of decisions made and great computational complexity.

Paper [7] describes the use of the ANN to detect cardiac rhythm abnormalities and other heart diseases. The algorithm of reverse error propagation is used as the method for ANN training. The shortcomings of the specified approach are its being limited to training only synaptic weights, without training the type and parameters of the membership function.

Article [8] represents the use of the ANN to detect avalanches. The algorithm of reverse error propagation is used as the method for ANN training. The disadvantages of this approach are its being limited to training only synaptic weights, without training the type and parameters of membership function.

Paper [9] demonstrates the ANN use for detection of anomalies in domestic authorization systems. The algorithm “The winner takes it all” is used as the method for training the Kohonen ANN. The shortcomings of this approach are

the error accumulation in the training process, limitation to training only synaptic weights, without training the type and parameters of membership function.

Article [10] presents the ANN use to identify the problems of detection anomalies in human encephalopathy. The method of fine tuning of ANN parameters is used as the ANN training methods. The shortcomings of this approach are error accumulation in the training process, limitation to training only synaptic weights without training the type and parameters of membership function.

Paper [12] shows the use of machine learning methods, such as the ANN and genetic algorithms. The genetic algorithm is used as the ANN training method. The disadvantages of this approach are its being limited to training only synaptic weights without training the type and parameters of membership function.

Paper [13] presents the use of machine learning methods, specifically, the ANN and the differential search method. In the course of the research, a hybrid method for ANN training, based on the use of the algorithm for reverse error propagation and differential search, was developed. The disadvantages of this approach are its being limited to training only synaptic weights without training the type and parameters of membership function.

In research [14], the methods for ANN training with the use of combined approximation of response surface, which ensures the smallest training and forecasting training, were developed. The disadvantage of the specified method is error accumulation in the course of training and impossibility of changing the ANN architecture during training.

Article [15] describes the use of the ANN to assess the performance of the machine using the previous time series of its performance. The SBM (Stochastic Block Model) and DEA (Data Envelopment Analysis) models are used for ANN training. The disadvantages of this approach are the limitation in the choice of the network architecture, training only synaptic weights.

Paper [16] presents the use of the ANN to assess geo-mechanical properties. The method of reverse error propagation is used as the method for ANN training. Improvement of the characteristics of the algorithm of reverse error propagation is achieved by increasing the training sample. The disadvantages of this approach are its being limited to training only synaptic weights without training the type and parameters of membership function.

Research [17] uses the ANN to assess the traffic road intensity. The method of reverse error propagation is used as the ANN training method. The improvement of the characteristics of the algorithms of reverse error propagation is achieved through the use of throughput links between each layer, so that each layer has only the residual function regarding the results of the previous layer. The disadvantages of this approach are its being limited to training only synaptic weights without training the type and parameters of membership function.

Analysis of scientific papers [1–17] has revealed that known methods for training artificial neural networks have the following shortcomings:

- are not capable to adapt the architecture of artificial neural networks depending on the amount of input information (the number of layers, the number of hidden layers, the number of links between neurons in the layer and between layers);
- are not capable to adapt the parameters of membership function depending on computation resources;

- are limited to training only synaptic weights or the type of membership function;

- accumulate a training error during training, which leads to an increase in training error with an increase in operation time;

- require a large volume of training sampling;
- are not capable to conduct training in real-time mode, but only in the packet mode (with a certain time interval of reaction delay).

The conducted analysis indicates that it is possible to achieve an increase in the efficiency of decision support systems through the development of the scientific and methodological apparatus for training artificial neural networks as the basis for intelligent decision support systems.

This will make it possible:

- to increase the amount of information that is possible to be processed by artificial neural networks;

- to enhance the reliability of decision making by intelligent systems of decision-making support;

- to adopt the speed of adaptation of the architecture and parameters of artificial neural networks according to arising tasks;

- to prevent dead-end situations during training artificial neural networks;

- to ensure predictability of the process of training artificial neural networks;

- to ensure unambiguous decisions made by intelligent decision support systems.

3. The aim and objectives of the study

The aim of this study is to develop an algorithm to train artificial neural networks for intelligent decision support systems, which makes it possible to perform processing of more information given the unambiguous character of decisions being made.

To achieve the aim, the following tasks were set:

- to formalize the task of training artificial neural networks;

- to determine the order of training artificial neural networks taking into consideration the shortcomings of the existing approaches with further development of the algorithm;

- to evaluate the developed algorithm.

4. Formalization of the task of training artificial neural networks

In order to determine the order of the operation of algorithms of training artificial neural networks, the generalized algorithm of training artificial neural networks is presented in this section. The algorithm of training artificial neural networks will be subsequently improved.

The architecture of the evolving multi-layered neural-fuzzy system is shown in Fig. 1 and consists of five consecutive layers.

The $(n \times 1)$ -dimensional vector of signals-images $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T$ (here $k = 1, 2, \dots$ is the current discrete time) that need to be processed is sent to the input (zero) layer.

The first hidden layer contains nh of membership functions, $\mu_h(x_i)$, $i = 1, 2, \dots, n$; $l = 1, 2, \dots, h$. Thus, h membership

functions are assigned for each input. This means that the hidden layer performs fuzzification of the input space. In this case, it is implied that in the process of training-evolution, it is necessary to configure both proper parameters of these functions, and their number: in Fig. 1 – the number of nodes of the first hidden layer μ_h .

The second hidden layer ensures the aggregation of membership levels, calculated in the first hidden layer, and consists of h multiplication units.

The third hidden layer is the layer that adjusts the synaptic weights $\omega_1, \omega_2, \dots, \omega_h$, that are subject to determining in the process of controlled training.

The fourth hidden layer, formed by two adders, calculates the total of the output signals of the second and the third hidden layers.

And, finally, in the fifth (output) layer, defuzzification is performed, which results in the computation of the output signal NFS (neural-fuzzy system) $\hat{y}(k)$.

Thus, if vector signal $x(k)$, is sent to the input of the neural-fuzzy system, the elements of the first hidden layer perform its fuzzification, calculating the level of membership $0 < \mu_{li}(x_i(k)) \leq 1$. Traditional Gaussians are normally used as membership functions

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

where c_{li} is the parameter of the center of the l -th membership function of the i -th input; σ_i is the parameter of the width of membership function of the i -th input.

It is worth noting that preliminary data normalization for a certain interval, for example, $-1 \leq x_i(k) \leq 1$, makes it possible to simplify calculations, because the width parameters σ_i can be taken as the same for all inputs, that is, $\sigma_i = \sigma$ (specified indicators are determined at the DSS design stage).

In addition to Gaussians (1), other nuclear functions can be used, for example, B-splines that meet the condition of a single breakdown, paired wavelets, flexible activation-membership functions [18, 19], etc.

Aggregated values $\prod_{i=1}^n \mu_{li}(x_i(k))$, are calculated in the second hidden layer for the Gaussians with the same width parameters:

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

where $c_l = (c_{l1}, c_{l2}, \dots, c_{ln})^T$.

Thus, the signals at the outputs of multiplication units of the second hidden layer are similar to the signals at the outputs of the neurons of the first hidden layer of usual RBFN (Radial basis function networks) [20].

The outputs of the third hidden layer is the value of $\omega_l \prod_{i=1}^n \mu_{li}(x_i(k))$, of the fourth layer – $\sum_{l=1}^h \omega_l \prod_{i=1}^n \mu_{li}(x_i(k))$ and $\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))$, and lastly, the signal appears at the output of the system (the fifth output layer)

$$\begin{aligned} \hat{y}(x(k)) &= \frac{\sum_{l=1}^h \omega_l \prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))} = \\ &= \sum_{l=1}^h \omega_l \frac{\prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))} = \\ &= \sum_{l=1}^h \omega_l \varphi_l(x(k)) = \omega^{hT} \varphi^h(x(k)). \end{aligned}$$

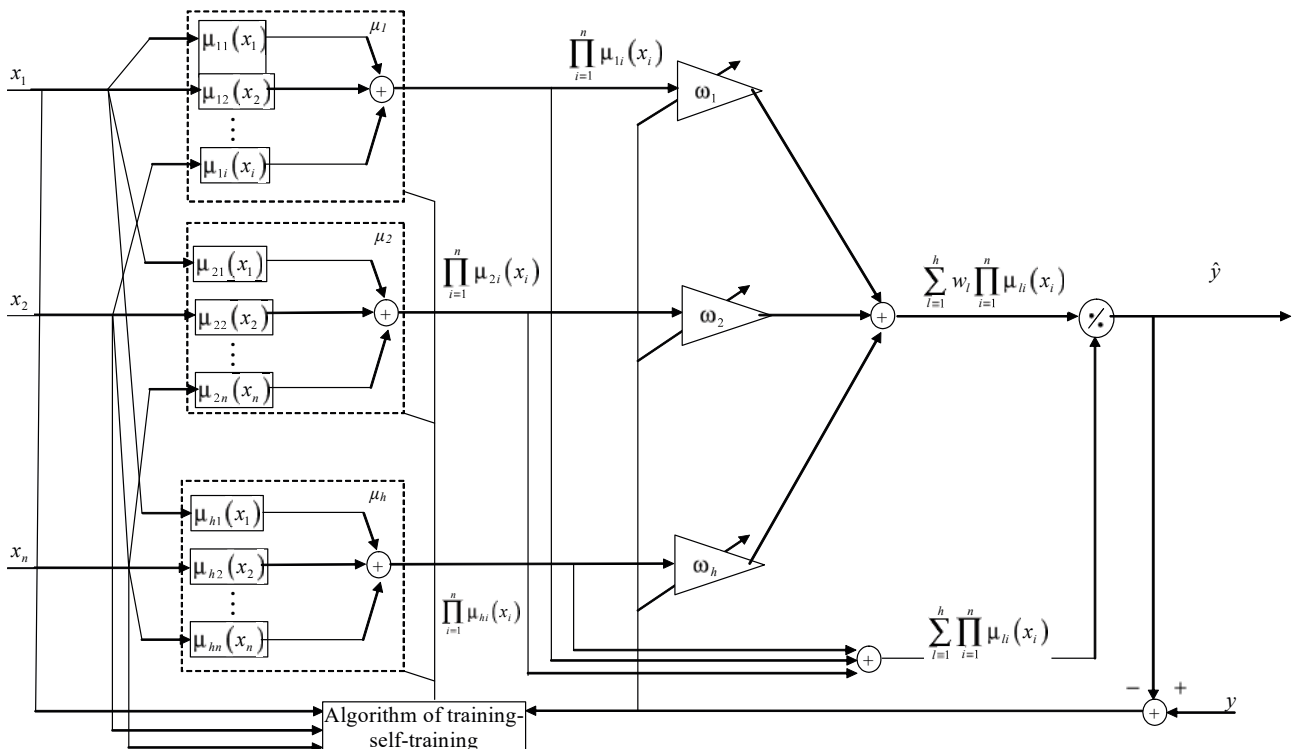


Fig. 1. Architecture of multi-layered neural-fuzzy evolving system

Where

$$\varphi_l(x(k)) = \frac{\prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))},$$

$$\omega^h = (\omega_1, \omega_2, \dots, \omega_2)^T,$$

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

It is not difficult to notice that the considered system implements a nonlinear display of input space into scalar output signal ($R^n \rightarrow R^1$) similarly to the normalized RBFN [21], and by the architecture (at fixed h) coincides with the TSK-system (Takagi, Sugeno, Kang) of the zero order, that is, the Vang–Mendel architecture [22].

5. The order of training artificial neural networks taking into consideration the shortcomings of existing approaches with subsequent development of the algorithm

Fig. 2 shows the proposed algorithm for training an artificial neural network. The improvement of the specified training algorithm is related to adding actions 5, 7, 8 to the known methods for training artificial neural networks. That is, there is an additional training of artificial neural networks, which was not taken into consideration in papers [1–17]:

- the architecture of artificial neural networks, depending on the amount of input information (the number of layers, the number of hidden layers, the number of links between neurons in the layer and between layers);
- the number and parameters of membership function.

The initial stage is entering the source data, specifically, the original architecture and the parameters of artificial neural network (stages 1–4).

Stage 1. Determining the number of the layers in the artificial neural network.

Stage 2. Determining the number of nodes in the layer of the artificial neural network.

Stage 3. Determining the number of links between the layers and nodes of the artificial neural network.

Stage 4. Determining the number of hidden layers of the artificial neural network.

Stage 5. Determining the number and parameters of membership functions (MF).

Stage 6. Determining synaptic weights of links of an artificial neural network.

Stage 7. Checking that the architecture and the parameters of an artificial neural network meet the demands that are put forward.

Stage 8. Making decision on adjustment of the architecture and the parameters of an artificial neural network.

Consider in detail the stages of the proposed training algorithm

Stage 9. Adjusting the parameters and the number of membership functions.

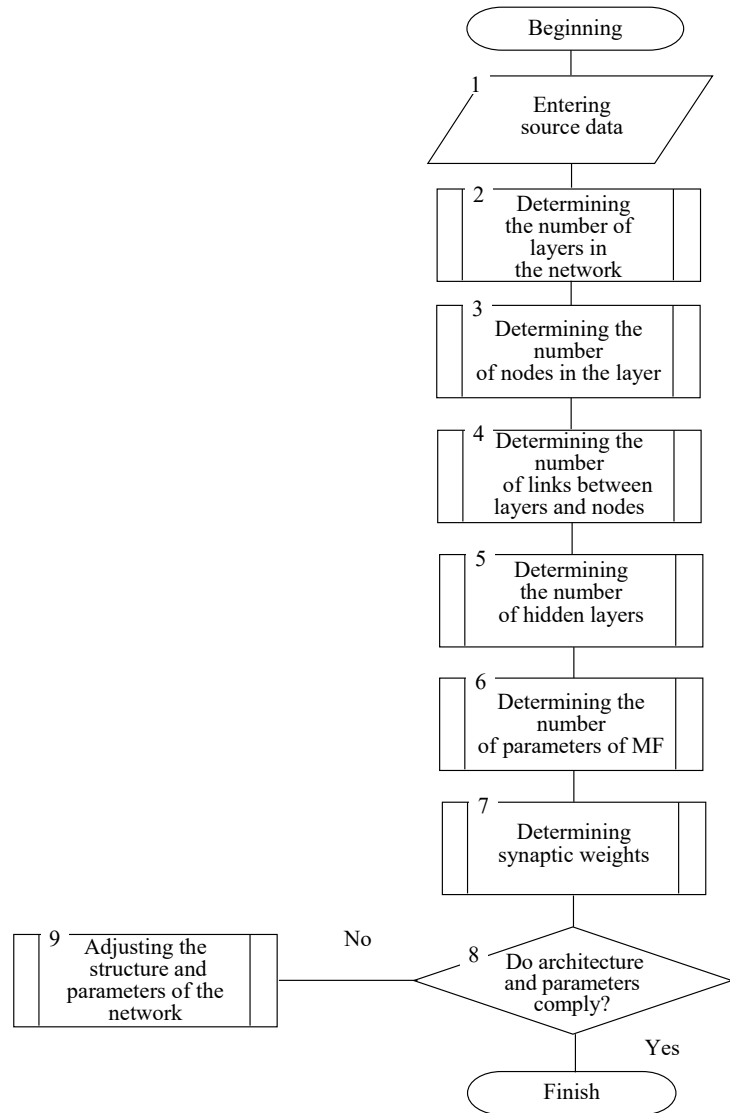


Fig. 2. Algorithm of functioning and training of an evolving artificial neural network

The process of adjustment of parameters and the number of membership functions goes on as follows. Let the first observation of learning sample $x(1) = (x_1(1), x_2(1), \dots, x_n(1))^T$ arrive at the input of the system, in which there are no membership functions in the output state in the first hidden layer.

This observation forms the first node of the first hidden layer μ_1 so that $c_{i1}(1) = x_i(1)$. In this way, n membership functions are formed and a single synaptic weight $\omega_1(0)$, is formed, which is randomly assigned in the interval $-1 \leq \omega_1(0) \leq 1$.

Further, for this membership function μ_1 we assign neighborhood radius r , determined by the maximum possible number of membership function in NFS h . Thus, if membership functions are distributed by axes in a uniformed way,

$$r = \frac{2}{h-1}. \tag{2}$$

Subsequently, when second observation $x(2)$ arrives, the condition is checked

$$\max_i |c_{ii} - x_i(2)| \leq r. \tag{3}$$

If this condition is met, the centers of membership functions of node μ_1 are adjusted according to the rule

$$c_{i1}(2) = c_{i1}(1) + \eta(2)(x_i(2) - c_{i1}(1)), \quad (4)$$

where η is the parameter of the training step.

Thus, at $\eta(2) = 0.5$

$$c_{i1}(2) = \frac{c_{i1}(1) + x_i(2)}{2}. \quad (5)$$

In case condition (3) is not met, there is formed the second node μ_2 of membership functions of the first hidden layer, the centers of which

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

Simultaneously with node μ_2 , the second synaptic weight w_2 , which is also assigned randomly, is introduced in NFS.

Let us assume that by the moment of arrival of observation $x(k)$ at the NFS input, p nodes of membership functions $\mu_1, \mu_2, \dots, \mu_p$, $p < h$, with the centers $c_{li}(k-1)$, $l = 1, 2, \dots, p$; $i = 1, 2, \dots, n$. have been formed. With the arrival of $x(k)$, the condition is checked

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

If this condition is met, the correction of the centers of membership function closest to the relevant components $x(k)$ is performed in accordance with the rules

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

It is not difficult to notice that (8) is nothing but the rule of self-training by T. Kohonen "The winner takes it all" [23] with the only difference that self-training of the Kohonen map is implemented on a hypersphere

$$\|x(k)\|_2 = 1, \quad (9)$$

and rule (8) – on a hypercube

$$\|x(k)\|_\infty = 1. \quad (10)$$

In case condition (7) is not met, $(p+1)$ node ($p+1 \leq h$) is formed in the system with the centers of membership function

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

Synaptic weight w_{p+1} is formed simultaneously with node μ_{p+1}

As it can be seen, the above procedure is an evolving hybrid algorithm by N. Kasabov [24] and the self-organized maps by T. Kohonen [23]. In this case, the process of evolution of architecture – self-training of membership functions can take place both continuously and until the number of membership functions is achieved.

Adjusting the parameters of the center and the width of membership functions can be performed by the teacher-assisted algorithm based on minimizing the target function. Adjustment is usually assigned using the Euclidean norm and for one pair of training data $(x(k), y(k))$ which has the form of

$$\begin{aligned} E(k) &= \frac{1}{2}(y(k) - \hat{y}(k))^2 = \\ &= \frac{1}{2}(y(k) - w^\top \varphi(x(k)))^2. \end{aligned} \quad (12)$$

When applying the method of the fastest descend, the appropriate adaptation formulas in the general case for $(n \times 1)$ -dimensional vector of input signals take the form of

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

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

where η_c is the parameter of the training step for the parameters of the centers of membership function; η_σ is the parameter of the training step for the parameter of width of membership function; $r = 1, 2, \dots, h$, $j = 1, 2, \dots, n$.

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

$$\begin{aligned} -0.5\sigma_{rj}^{-2}(k+1) &= \\ &= \sigma_{rj}(k) - \eta_\sigma \frac{\partial E(k)}{\partial (-0.5\sigma_{rj}^{-2})}. \end{aligned} \quad (15)$$

When using traditional Gaussians (9) as membership functions, the corresponding formulas of the target function gradient (12) for one pair of training data takes the form

$$\begin{aligned} \frac{\partial E(k)}{\partial c_{rj}} &= (w^\top \varphi(x(k)) - (x(k))) \times \\ &\times \sum_{l=1}^h w_l \frac{\partial \varphi_l(x(k))}{\partial c_{rj}}, \end{aligned} \quad (16)$$

$$\begin{aligned} \frac{\partial E(k)}{\partial (-0.5\sigma_{rj}^{-2})} &= (w^\top \varphi(x(k)) - y(x(k))) \times \\ &\times \sum_{l=1}^h w_l \frac{\partial \varphi_l(x(k))}{\partial (-0.5\sigma_{rj}^{-2})}, \end{aligned} \quad (17)$$

where

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

Derivatives

$$\frac{\partial \varphi_l(x)}{\partial c_{rj}}$$

and

$$\frac{\partial \varphi_l(x)}{\partial (-0.5\sigma_{rj}^{-2})},$$

determined based on (9) and (18) can be written as follows:

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

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

where δ_{lr} in the Kronker delta,

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

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

Derivatives

$$\frac{\partial \mu_{rj}(x_j)}{\partial c_{rj}}$$

and

$$\frac{\partial \mu_{rj}(x_j)}{\partial (-0.5\sigma_{rj}^2)},$$

determined based on (9) can be written as

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

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

Stage 6. Determining synaptic weights of the links of an artificial neural network.

As it has already been noted, the known algorithms of training – identification [26, 27] can be used for the adjustment of synaptic weights of a neural-fuzzy system:

– exponentially-weighted recurrent method of least squares;

$$\left\{ \begin{aligned} &w^h(k) = w^h(k-1) + \\ &\frac{P^h(k-1)(y(k) - w^{hT}(k-1)\varphi^h(x(k)))}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))} \times \\ &\times \varphi^h(x(k)) = w^h(k-1) + \\ &\frac{P^h(k-1)(y(k) - \hat{y}^h(x(k)))}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))} \varphi^h(x(k)), \\ &P^h(k) = \\ &\frac{1}{\beta} \left(\frac{P^h(k-1) - \frac{P^h(k-1)\varphi^h(x(k))\varphi^{hT}(x(k))P^h(k-1)}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))}}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))} \right) = \\ &= \left(\sum_{\tau=1}^k \varphi^h(x(\tau))\varphi^{hT}(x(\tau)) \right)^{-1}, 0 < \beta \leq 1, \end{aligned} \right. \tag{23}$$

where $y(t)$ is the external training signal, β is the parameter of forgetting outdated information;

– optimal by performance gradient one-step algorithm by Kachmar–Widrow–Hoff

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

– training algorithm that possesses both monitoring and smoothing properties [27].

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

and similar procedures.

Procedure (25) is related to algorithm (23) via ratio

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

and at $\beta=0$ we obtain the form of algorithm (23).

The process of adjustment of synaptic weights can occur simultaneously with self-training – evolution of the first hidden layer.

Let by the moment of arrival of observation $x(k)$, p nodes of membership function $\mu_1, \mu_2, \dots, \mu_p$ have been formed and vector of synaptic weights $w^p(k-1)$ has been calculated. Let us assume that condition (7) is not met, which at once leads to the formation of node μ_{p+1} and assigning an arbitrary initial value of synaptic weight w_{p+1} . In this case, the output signal of the NFS can be represented in the form

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

and algorithm (24) –

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

As it can be seen, the process of simultaneous evolution – self-training-controlled training does not cause any computation problems.

Stage 7. Checking the compliance of the architecture and the parameters of the artificial neural network parameters with the stated requirements.

At this stage, there is a verification of compliance of the architecture and the parameters of an artificial neural network with the requirements. The specified requirements for functioning of an artificial neural network are stated at the stage of designing the decision support systems and depend on the type and scope of tasks performed by decision support systems.

Stage 8. Making decision on the correction of the architecture and the parameters of an artificial neural network.

Based on the comparative estimation of the requirements stated for the ANN functioning and their actual efficiency, the decision to adjust the ANN parameters is made, specifically, (for 6–8 in the diagram in Fig. 2):

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

In case it is impossible to ensure the necessary requirements for the ANN, the decision to change the ANN architecture and to determine the initial ANN parameters is made (for 1–5 in the diagram Fig. 2).

6. Discussion of the results of development of the algorithm to train artificial neural networks for intelligent decision support systems

The algorithm of training of artificial neural networks for intelligent decision support systems was proposed.

The effectiveness of the proposed evolving multi-layered neural-fuzzy system with hybrid training was demonstrated in assessing the radio communication system with the pseudo-random re-adjustment of working frequency (SRWF) under the influence of deliberate noise interference.

Simulation was carried out under the following parameters:

- radio communication means with the SRWF: frequency range – 30–512 MHz; power of the transmitter – 10 W; width of the band of radiating frequencies – 12.5 kHz, sensitivity of the receiver –110 dB; number of means of radio communication in the network – 4; number of frequency channels for readjustment – 10,000; readjustment rate – 333.5 jumps/sec;

- complex of radio-electronic suppression (RES): frequency range – 30–2,000 MHz; power of the transmitter – 2,000 W; maximal frequency band that can be suppressed – 80 MHz; number of radio lines with the SRWF that can be suppressed simultaneously – 4, interference type – noise barrier interference with frequency manipulation as one of the most common and the influence of which is well known, the strategy of the RES complex is dynamic

A practical check of effectiveness of the developed methodology was performed through the development of software Python 3.6 (Jupiter-notebook). This software was installed on the PC, which is connected to 4 programmable transceivers LimeSDR with the GNU Radio software and connected to the RIGOL DG5252 noise generator, which simulated the operation of the RES complex.

To compare the effectiveness of the proposed algorithm with the known ones, the methods for training artificial neural networks used in decision support systems were used.

To assess the impact of deliberate noise interferences on radio communication devices with the SRWF, the neural-fuzzy system with the number of inputs $n=5$. was used.

The system was trained based on the sample, with the control signals:

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

The number of iterations of estimations of the state of the radio connection system with the SRWF is 10.

To check the effectiveness of the proposed algorithm, we will assume that the operation time of the programmable device of radio communication with the SRWF at the same frequency is the same with the RES complex.

We will briefly describe the graphical ratios obtained in Fig. 3. Fig. 3, 4 were obtained by averaging signal/noise ratio in the frequency sub-channels. This advantage is due to the greater accuracy of the assessment of the signal/noise ratio in the frequency sub-channels due to the quick adjustment of parameters of artificial neural network to the type and intensity of the data coming to its input. At the same time, we would like to focus attention on the fact that during the evaluation with the help of the proposed algorithm, the evaluation error does not increase during the operation of the artificial neural network as opposed to other approaches (Fig. 3, 4).

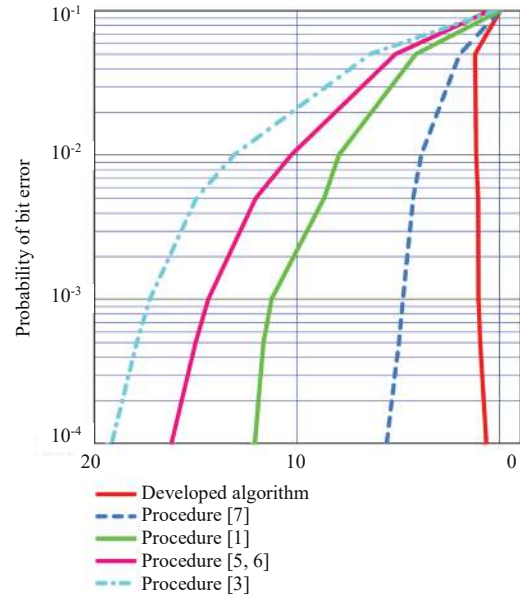


Fig. 3. Dependence of probability of the bit error on the signal/noise ratio Q_0^2 for difference procedures at the influence of fluctuating noise and noise barrier interference with coefficient of overlapping $r=1$ at 10 iterations of assessment of an artificial neural network

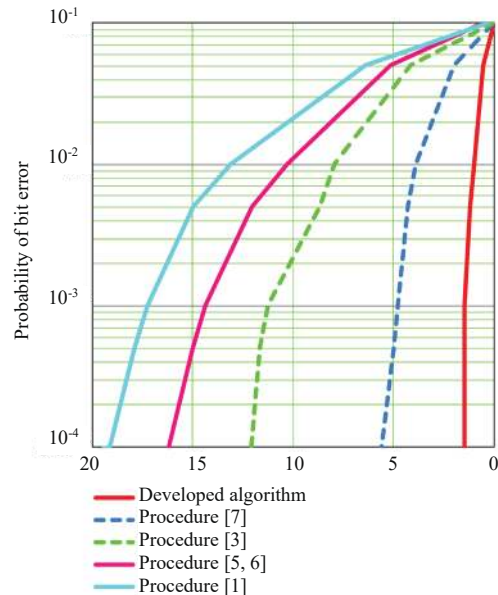


Fig. 4. Diagram of dependence of the probability of a bit error on the signal/noise ratio Q_0^2 (for the case of the noise interference in the part of the band with frequency manipulation with coefficient of overlapping $r=0.5$) at 10 iterations of evaluation of an artificial neural network

Fig. 4 shows the diagram of dependence of the probability of a bit error on the signal/noise ratio (for the case of noise interference in the part of the band with frequency manipulation $r=0.5$).

Evaluation of computational complexity of the implementation of the developed algorithm showed that for the assigned output data and when using the processor ADSP-21261, the assessment of the state of channel of a radio communication system can be carried out in real time. At the same time, the delay required for the transmission of the information about these values in the service feedback channel is taken into consideration.

The research of the developed algorithm showed that the indicated training algorithm provides the efficiency of training of artificial neural networks that is on average by 16–23 % higher and does not accumulate errors during training. This efficiency is shown in Fig. 3 and Fig. 4, that is, comparative assessment of procedure [7] and the proposed algorithm under the specified operating conditions (6 dB and 2 dB on the last training iteration).

The gain in the range of 16–23 % was obtained during the operative control of the current state and the interference situation in the channels employed for the transmission, for the time comparable with the duration of the information exchange cycle. The ambiguity of determining the state of the channels is caused by various parameters of the receive signal due to different signal trajectories and signal/noise levels in the subchannels.

The main advantages of the proposed learning algorithm are:

- does not accumulate training errors in the process of training artificial neural networks due to the adjustment of the parameters and the architecture of an artificial neural network;
- unambiguous obtained results;
- a wide scope of application (decision-making support systems);
- simplicity of mathematical calculations;
- the possibility of adapting the system in the course of operation;
- the possibility to synthesize the optimal structure of the decision support system.

The shortcomings of the proposed algorithm include:

- loss of informativeness during the estimation (prediction) due to the construction of membership function in the course of procedures of input parameters fuzzification/defuzzification (transition from clear to fuzzy evaluation and vice versa). This loss of informativeness can be reduced by selecting the type of membership function and its parameters at practical implementations of the proposed algorithm in decision support systems. Selection of the type of membership function depends on the computational resources of a specific electronic-computing device.

- lower evaluation accuracy by a separate parameter of the state assessment;
- loss of accuracy of results during the readjustment of the architecture of an artificial neural network.

The specified algorithm will make it possible:

- to conduct training of artificial neural networks;
- to determine effective measures to enhance the operation efficiency of artificial neural networks;
- to improve the operation efficiency of artificial neural networks by training the parameters and the architecture of networks;

- to reduce the use of computational resources of decision support systems;
- to develop the measures aimed at improving the effectiveness of training artificial neural networks;
- to increase the efficiency of information processing in artificial neural networks.

This research is a further development of the research conducted by the authors aimed at developing the theoretical foundations for the enhancement of the effectiveness of artificial intelligence systems, published earlier [1, 2, 29, 30].

The directions for further research should be aimed at reducing the computational costs for processing different data in special-purpose systems.

7. Conclusions

1. The task of training artificial neural networks was formalized. In the course of formalization, the working order of the algorithms of training artificial neural networks was determined. The generalized algorithm for training artificial neural networks was introduced. Based on the specified algorithm, the improved algorithm of training artificial neural networks was developed.

2. The algorithm of training artificial neural networks for intelligent decision support systems implements the following:

- performs training of not only the synaptic weights of an artificial neural network, but also of the type and the parameters of membership function;
- in case of impossibility to ensure the given quality of functioning of artificial neural networks by training parameters, training the architecture of an artificial neural networks is performed;
- the architecture, the type and the parameters of membership function are selected taking into consideration the computing resources of the device and considering the type and the amount of information entering the input of an artificial neural network;
- there is no accumulation of error of training artificial neural networks as a result of processing the information coming to the input of artificial neural networks.

3. The study of the developed algorithm showed that the specified training algorithm provides the efficiency of training artificial neural networks that is on average by 16–23 % higher and does not accumulate errors during learning.

Acknowledgement

Authors express their gratitude for assistance to:

Doctor of Technical Sciences, Professor Mr. Oleksiy Viktorovych Kuvshynov – the deputy head of the Educational and Scientific Institute at Ivan Chernyakhovsky National University of Defense of Ukraine;

Doctor of Technical Sciences, Senior Scientific Researcher Mr. Yuriy Volodymyrovich Zhuravskiy – the leading scientific researcher of the research center at Zhytomyr Military Institute named after S.P. Korolev;

Candidate of Technical Sciences, Associate Professor Mr. Olexander Mykolayovych Bashkirov – the leading scientific researcher of the Central scientific research institute of armament and military equipment of the Armed forces of Ukraine.

References

1. Kalantaievska, S., Pievtsov, H., Kuvshynov, O., Shyshatskiy, A., Yarosh, S., Gatsenko, S. et. al. (2018). Method of integral estimation of channel state in the multiantenna radio communication systems. *Eastern-European Journal of Enterprise Technologies*, 5(9 (95)), 60–76. doi: <https://doi.org/10.15587/1729-4061.2018.144085>
2. Kuchuk, N., Mohammed, A. S., Shyshatskiy, 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 (1.2), 1–6. Available at: <http://www.warse.org/IJATCSE/static/pdf/file/ijatcse01812sl2019.pdf>
3. 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), 114. doi: <https://doi.org/10.3390/ijerph14020114>
4. 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. doi: <https://doi.org/10.30525/2256-0742/2018-4-1-188-194>
5. Manea, E., Di Carlo, D., 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. doi: <https://doi.org/10.1016/j.ecolind.2018.12.017>
6. 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. doi: <https://doi.org/10.1016/j.jairtraman.2017.10.007>
7. 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. doi: <https://doi.org/10.21822/2073-6185-2018-45-2-114-124>
8. Zhdanov, V. V. (2016). Experimental method to predict avalanches based on neural networks. *Ice and Snow*, 56 (4), 502–510. doi: <https://doi.org/10.15356/2076-6734-2016-4-502-510>
9. 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. 2017 20th Conference of Open Innovations Association (FRUCT). doi: <https://doi.org/10.23919/fruct.2017.8071301>
10. 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.
11. Chica, J., Zaputt, S., Encalada, J., Salamea, C., Montalvo, M. (2019). Objective assessment of skin repigmentation using a multilayer perceptron. *Journal of Medical Signals & Sensors*, 9 (2), 88. doi: https://doi.org/10.4103/jmss.jmss_52_18
12. 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, 01010. doi: <https://doi.org/10.1051/epjconf/201921701010>
13. 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. doi: <https://doi.org/10.4316/aee.2019.04007>
14. Mishchuk, O. S., Vitynskiy, P. B. (2018). Neural Network with Combined Approximation of the Surface of the Response. *Research Bulletin of the National Technical University of Ukraine “Kyiv Politechnic Institute”*, 2, 18–24. doi: <https://doi.org/10.20535/1810-0546.2018.2.129022>
15. 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. doi: <http://doi.org/10.22059/imj.2018.129192.1006898>
16. Parapuram, G., Mokhtari, M., Ben Hmida, J. (2018). An Artificially Intelligent Technique to Generate Synthetic Geomechanical Well Logs for the Bakken Formation. *Energies*, 11 (3), 680. doi: <https://doi.org/10.3390/en11030680>
17. 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. doi: <https://doi.org/10.20537/2076-7633-2018-10-3-359-367>
18. 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 (1). doi: <https://doi.org/10.2478/itms-2013-0007>
19. 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.
20. Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 842.
21. Nelles, O. (2001). *Nonlinear System Identification*. Springer. doi: <https://doi.org/10.1007/978-3-662-04323-3>
22. Wang, L.-X., Mendel, J. M. (1992). Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Transactions on Neural Networks*, 3 (5), 807–814. doi: <https://doi.org/10.1109/72.159070>
23. Kohonen, T. (1995). *Self-Organizing Maps*. Springer. doi: <https://doi.org/10.1007/978-3-642-97610-0>
24. Kasabov, N. (2003). *Evolving Connectionist Systems*. Springer. doi: <https://doi.org/10.1007/978-1-4471-3740-5>
25. Sugeno, M., Kang, G. T. (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28 (1), 15–33. doi: [https://doi.org/10.1016/0165-0114\(88\)90113-3](https://doi.org/10.1016/0165-0114(88)90113-3)
26. Ljung, L. (1987). *System Identification: Theory for the User*. Prentice Hall, 432.
27. Otto, P., Bodyanskiy, Y., Kolodyazhnyi, V. (2003). A new learning algorithm for a forecasting neuro-fuzzy network. *Integrated Computer-Aided Engineering*, 10 (4), 399–409. doi: <https://doi.org/10.3233/ica-2003-10409>
28. Narendra, K. S., Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1 (1), 4–27. doi: <https://doi.org/10.1109/72.80202>
29. I. Aliinykov, K. A. Thamer, Y. Zhuravskiy, O. Sova, N. Smirnova, R. Zhyvotovskiy et. al. (2019). Development of a method of fuzzy evaluation of information and analytical support of strategic management. *Eastern-European Journal of Enterprise Technologies*, 6 (2 (102)), 16–27. doi: <https://doi.org/10.15587/1729-4061.2019.184394>
30. Koshlan, A., Sahnikova, O., Chekhovska, M., Zhyvotovskiy, R., Prokopenko, Y., Hurskiy, T. et. al. (2019). Development of an algorithm for complex processing of geospatial data in the special-purpose geoinformation system in conditions of diversity and uncertainty of data. *Eastern-European Journal of Enterprise Technologies*, 5 (9 (101)), 35–45. doi: <https://doi.org/10.15587/1729-4061.2019.180197>