UDC 004.85(045)
DOI:10.18372/1990-5548.60.13813

[1]V. M. Sineglazov,
[2]R. S. Koniushenko

DEEP LEARNING FUZZY CLASSIFIER

[1,2]Aviation Computer-Integrated Complexes Department, Faculty of Air Navigation Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine
E-mails: [1]svm@nau.edu.ua ORCID 0000-0002-3297-9060, [2]romankonyushenko@gamil.com

***Abstract*—*It is considered a classification problem solution based on analysys of represented review. It's shown that the neural networks have important advantages beside other methods, such as: classification using the nearest neighbor method, support vector classification, classification using decision trees, etc. Amount of artificial neural networks exists futher networks have the simplest structure, but the precision of the solution can be increased with help of deep learning approach, which is supposes the use of additional neural network for the solution of pretraining tasks(deep believe networks). It's proposed new tophology which consist of: Takagi-Sugeno-Kang fuzzy classifier and Limited Boltzmann Machine neural network. Despite on this thopology was proposed early in this article it's carried out enough researches that permited to specify the learning algorithm. An example of proposed algorithm implantation is represented.*

Index Terms—Neural network; fuzzy neural network; deep learning.

I. INTRODUCTION

The classification problem is a widespread especially in math and technic. It may be called, for example in medical diagnostic systems, images recognition – especially for search systems, recognition and determination of the moving objects for example unmanned aerial vehicles and others.

From mathematical point of view the task of classification is the task of splitting a set of objects or observations into a priori given groups, called classes. Inside every group each object related to it considered to be similar to each other, and have approximately similar properties and attributes. A finite set of objects is specified, for which it is known for what classes they belong to. In this case, the solution is based on the analysis of the meanings of attributes, for the class determination.

For solving the classification problem, neural networks are used, in particular fuzzy neural networks. Many classifiers were created on base of neural network architecture, which are now widely used to solve medical diagnosis, image classification, data classification, bankruptcy prediction, and so on. These technologies become indispensable in the business, industry and science fields. Neural networks and fuzzy neural networks have a high output quality, but like all systems, they are not ideal and make mistakes.

II. PROBLEM STATEMENTS

Let $\boldsymbol{X}$ set of object descriptions, $\boldsymbol{Y}$ set of class names. There is a target addiction – reflection $y^*: X \to Y$, the meaning of which is known on the objects of the training sample $X^m = \{(x_1, y_1), ..., (x_m, y_m)\}$. Need to build algorithm (neural network) $a: X \to Y$, capable to classify an arbitrary object $x \in X$ [4].

The main approaches for solving the classification problem are the following: decision trees, reference vectors, closest neighbor, CBR-method, linear regression, neural network method [1]. Analysis of these methods showed that the most effective is the neural network method.

The good classification problem solution often demands the use of multilayer neural networks that is determined by learning sample. To simplify the structure of neural networks can be achieved fuzzy neural networks, that due to fuzzification and linguistics rules use. The representatives of this neural network class are NEFCLASS [16], ANFIS [18], TSK [6].

But still exist a problem how to optimize logic rules? And how to determine its number. If system create number of rules by its own, how can we be sure that resulted number of rules is optimal? And if number of rules hardcoded by human problem still the same. Large number of rules create higher output error, small number of rules has also high error in output. But if problem of rules number is hard to solve, maybe we need to overview the problem from another angle? Let's look for basic structure of fuzzy neural classifier in Fig. 1.

Further improvement of the efficiency of such classifiers can be achieved through deep learning

[3]. It is proved [2] that the study of deep neural networks (NN) is possible with the help of "greedy" layer training. The essence of "greedy" learning is finding locally-optimal solutions for each sub task, expecting the end result to be optimal in the end. Sub-task in the process of training the neural network is to study a single layer.

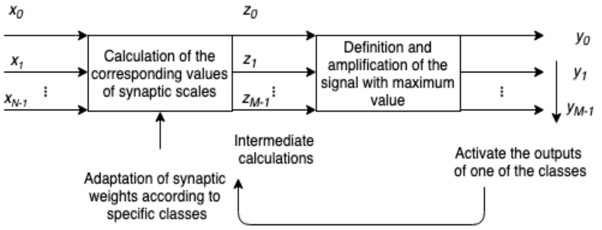

Fig. 1. Block diagram of the neural network classifier

It is proposed [2] the concept of training each layer separately with the help of NN autoassociate with the subsequent study of the network by a standard method, for example, the method of reverse error propagation.

Automobile associate networks are networks that, in the output, should be as accurate as possible representing the data that is fed to the network. For problems of deep learning, limited machines of Boltzmann and autoconverters are used. So, summing up the information, modem out the following statements:

– **Classify an object** is then, specify the number (or class name) to which this object belongs.

– **Classification of an object** is the number or class name, issued by the classification algorithm as a result of its application to this particular object.

III. PROBLEM SOLUTIONS

As it was shown early for the improvement of classification tasks it's advisable to use basic network (fuzzy neural network) and additional neural network for pre-training of basic NN, to use the topology of deep believe NN. So, it can be formulated a problem statements:

- it's necessary to choose the topology of NN;
- need to choose NN for pre-training Restricted Boltzmann Machine or Autoencoder;
- to develop new topology of deep neural fuzzy classifier;
- to develop new learning algorithm.

This problem belongs to optimization task with criterion of generalized error [19].

For the solution of this problem it was researched the NEFCLASS neural networks. But topology of this NN doesn't give possibility to use restricted Boltzmann machine (RBM) for its pre-training. So it is proposed to use a multilayer perceptron.

As network for pre-training was chosen RBM. The restricted Boltzmann machine more popular than for example autoencoder. It is existed a lot of algorithms that's permit to adjust RBM.

For solving tasks 3 and 4, RBM must be overviewed in more detailed way.

The deep belief NN contains a lot of hidden layers and performs a deep hierarchical transformation of input images as shown in Fig. 2.

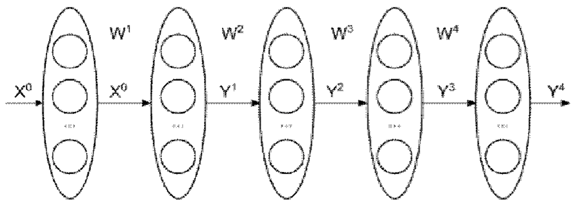

Fig. 2. Neural Network of Deep Learning

The initial value of the jth neuron of the kth layer is determined in the following way:

$$y_i^k = F\left(S_i^k\right),$$

$$s_j^k = \sum_{i=1} w_{ij}^k y_i^{k-1} + T_j^k,$$

where F is the activation function of the neural element; s_j^k is the weighed amount jth neuron of kth the layer; w_{ij}^k weight coefficient between ith neuron of (k–1)th layer and jth neuron of kth layer; T_j^k threshold value jth neuron of kth layer.

For the first layer:

$$y_i^0 = x_i.$$

In the matrix form, the output vector of the kth layer:

$$Y^k = F(S^k) = F\left(W^k Y^{k-1} + T^k\right),$$

where **W** matrix of weight coefficients; Y^{k-1} is the source vector of (k–1)th layer; T^k is the vector of threshold values for neurons of kth layer.

Preliminary learning of the NN by the method of layer training, starting with the first layer. This training is conducted without a teacher.

Configuring synaptic connections across the network using the error-return algorithm or the "wakefulness and sleep" algorithm.

A key stage in the training of neural networks of deep trust, which pretends to be superior to superficial models, is the prior learning. One of the main approaches to prior learning is a method based on providing each layer of the network in the form of a RBM, as well as the entire network – in the form of a set of such machines.

The restricted Boltzmann machine consists of two layers of stochastic binary neuronal elements, which are interconnected by bi-directional symmetric bonds (Fig. 3). The input layer of the neural elements is called visible (layer *X*), and the second layer is called hidden (layer *Y*). A neural network of deep trust can be imagined as a collection of limited Boltzmann machines. A restricted Boltzmann machine can generate (represent) any discrete distribution if a sufficient number of hidden neurons is used [10].

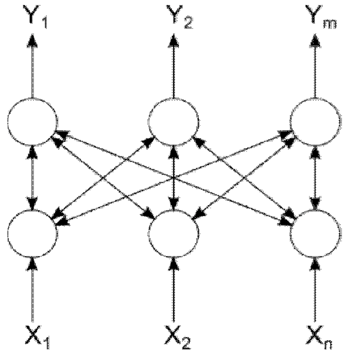

Fig. 3. Restricted Boltzmann Machine

This network is a stochastic neural network in which the state of visible and hidden neurons varies according to the probabilistic version of the sigmoid activation function of the form: 1

$$p\left(y_i = 1 \mid x\right) = \sigma\left(\sum_{i}^{n} w_{ij} x_i + T_j\right),$$

$$p\left(x_j = 1 \mid y\right) = \sigma\left(\sum_{i}^{n} w_{ij} y_i + T_j\right),$$

where $\sigma(x) = \dfrac{1}{1+e^{-x}}, w_{ij}$ weight coefficients; T_i, T_j are threshold items.

Such a structure is called binary RBM. In addition to the binary data-oriented model, there is a linear-binary RBM model in which a linear activation function is used on a visible layer. A similar RBM is recommended for use with non-binary (real) data.

The following rules are used to teach such an RBM:

$$p\left(y_i = 1 \mid x\right) = \sigma\left(\sum_{i}^{n} w_{ij} x_i + T_j, 1\right),$$

$$x_i = N\left(\sum_{j}^{m} w_{ij} y_i + T_j, 1\right),$$

where $N(m, s)$ is a normally distributed random value with an average value of m and a standard deviation s.

The states of the visible and hidden elements are taken independently:

$$P(x \mid y) = \prod_{i=1}^{n} P(x_i \mid y),$$

$$P(y \mid x) = \prod_{j=1}^{m} P(y_i \mid x).$$

Thus, the states of all elements of a RBM are determined by the probability distribution. In restricted Boltzmann machine, the hidden layer neurons are detector signs that keep the patterns of input data. The main task of learning is to reproduce the distribution of input data based on the state of the hidden layer neurons as accurately as possible. This is equivalent to maximizing the likelihood function by modifying the synaptic connections of the neural network.

The resulting RBM expressions are quite complex in computational terms, so Hinton suggested using the approximation that he called contrastive divergence (CD). This approximation is based on the Gibbs sampling (Gibbs sampling). In this case, the first terms in the expressions for the gradient characterize the data distribution at time $t = 0$, while the other components characterize the reconstruction or generate a state model at time $t = k$. Therefore, the CD-k procedure can be represented as follows:

$$x(0) \rightarrow y(0) \rightarrow x(1) \rightarrow y(1) \rightarrow \ldots \rightarrow x(k) \rightarrow y(k).$$

As a result, you can get the following rules for training RBMs:

$$w_{ij} = (t+1) = w_{ij}(t) + \alpha\left(x_i(0) y_i(0) - x_i(k) y_j(k)\right),$$

$$T_i(t+1) = T_i(t) + \alpha\left(x_i(0) - x_i(k)\right),$$

$$T_j(t+1) = T_j(t) + \alpha\left(x_j(0) - x_j(k)\right),$$

where $x_i(0) y_i(0)$ are characterizes the distribution of data at the time $t = 0$; $x_i(k) y_j(k))$ at time $t = k$, k is the parameter defining the number of "back and forth" iterations to perform the approximation of the corresponding gradients. In the case of using CD-1, $k = 1$, obtain the following training rules

$$w_{ij} = (t+1) = w_{ij}(t) + \alpha\left(x_i(0) y_i(0) - x_i(1) y_j(1)\right),$$

$$T_i(t+1) = T_i(t) + \alpha\left(x_i(0) - x_i(1)\right),$$

$$T_j(t+1) = T_j(t) + \alpha\left(x_j(0) - x_j(1)\right),$$

from the formulas it can be seen that the rules of a RBM minimize the difference between the original data and the data generated by the model. Data generating the model is obtained using the Gibbs sampling procedure.

IV. TSK AS FUZZY CLASSIFIER

The fuzzy TSK system offered by Takagi, Sugeno and Kang, it builds on its own unique interpretation, good learning ability and good approximation performance, is widely used in system identification, image recognition, image processing and intelligent data analysis and other areas. The fuzzy TSK system uses linguistic rules and fuzzy sets for processing data. TSK is simple and has good approximation performance. However, she still has no in-depth training. The rules for fuzzy output TSK are expressed as follows:

Fuzzy rule R^k:

$$x_1 \text{ is } A_1^k \wedge x_2 \text{ is } A_2^k \wedge ... \wedge x_d \text{ is } A_d^k \quad \text{then}$$

$$y^k = a_0^k + a_1^k x_1 + ... + a_d^k x_d = (a^k) \cdot [1, x^T]^T,$$

$$k = 1, ..., K,$$

where A_n^M is the meaning of a linguistic variable x_i for the rule R_M with membership function, which includes k the corresponding input x_i; d is the dimension of the sample; K is the number of fuzzy rules; $a^k = (a_0^k, ..., a_d^k)$ are consistent parameters; $\wedge$ is a fuzzy communication operator. After a series of operations, the fuzzy system is displayed

$$y^0 = \frac{\sum_{k=1}^{K} w^k y^k}{\sum_{k=1}^{K} w^k},$$

where w^k is the membership function. If the Gaussian function is given as a membership function, obtain next formulas:

$$w_{A_i^k}(x_i) = \exp\left(\frac{-(x_i - v_i^k)^2}{2\delta_i^k}\right),$$

$$w(x) = \frac{w^k(x)}{\sum_{k=1}^{K} w^k(x)},$$

$$w^k(x) = \prod_{i=1}^{d} w_{A_i^k}(x_i),$$

$$v_i^k = \frac{\sum_{j=1}^{N} \mu_{jk} x_{ji}}{\sum_{j=1}^{N} \mu_{jk}},$$

$$\delta_i^k = \frac{h \sum_{j=1}^{N} \mu_{jk} (x_{ji} - v_i^k)^2}{\sum_{j=1}^{N} \mu_{jk}},$$

where, v_i^k and δ_i^k the clustering center and the width, respectively, can be obtained by clustering algorithm or by other partitioning method. h is a scaling constant and can be set manually or determined by a specific training strategy.

Considering the above data, can obtain the following expressions

$$x_e = (1, x^T)^T,$$

$$x^k = \mu^k(x) x_e,$$

$$x_g = \left((x^1)^T, (x^2)^T, ..., (x^K)^T\right),$$

$$a^k = (a_0^k, a_1^k, ..., a_d^k)^T,$$

$$a_g = \left((a^1)^T, (a^2)^T, ..., (a^K)^T\right),$$

$$y^0 = a_g^T x_g.$$

Sequential parameters can be calculated using the least square [2] or least learning machine method.

V. TSK_DBN TOPOLOGY AND ALGORITHM

The architecture of a deep fuzzy classifier based on a RBM is shown in Fig. 4. Let's consider these parts of architecture in details.

The first part is part of the network with fuzzy logic (left part of Fig. 4), which is used to obtain fuzzy conclusions, as well as to obtain the final real values as the precursor of the fuzzy network. Using the mechanism of fuzzy conclusion, fuzzy rules can be linguistically interpreted with expert knowledge.

The second part is an adaptive deep neural network (right part from Fig. 4), which is introduced for in-depth analysis of features in TSK_DBN. This subsystem ensures that the result will be highly accurate.

The third part contains sequences TSK_DBN, which are studied from the first and second parts, respectively. Using the results of a fuzzy part and data from a deep network, the classifier can solve complex tasks of classification.

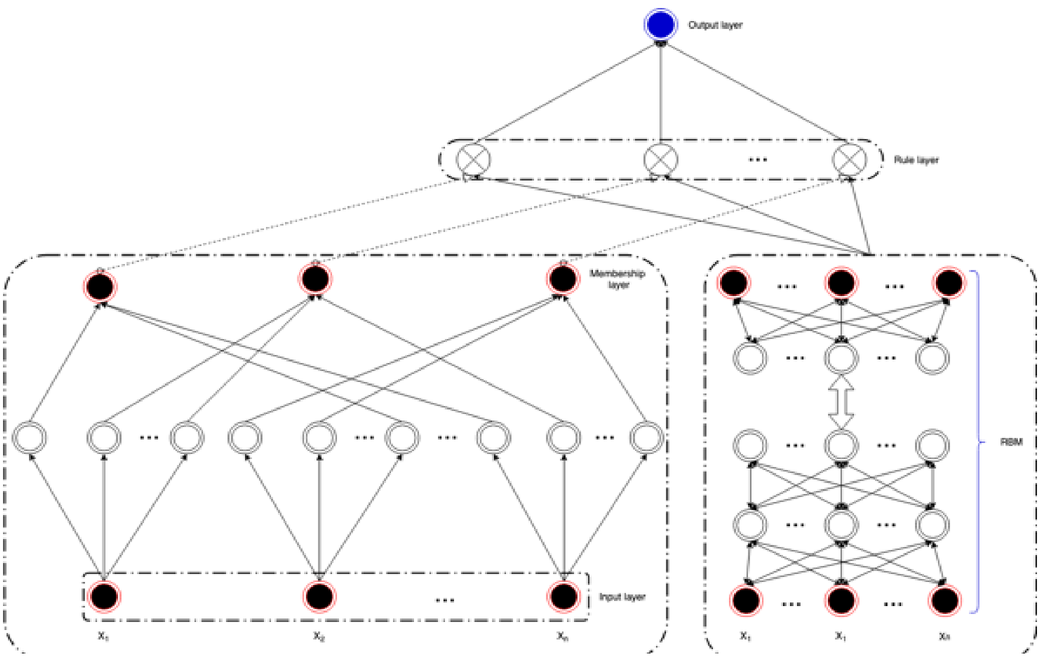

Fig. 4. Fuzzy classifier of deep learning

With reference to Zhou's D-TSK-FC [12], use the Least Learning Machine (LLM) [11] to compute sequential parameters, which means that TSK_DBN can also be expanded for large data sets with high accuracy and a brief interpretation.

The model of the rules of this classifier is as follows:

$$x_1 \text{ is } A_1^k \wedge x_2 \text{ is } A_2^k \wedge \ldots \wedge x_d \text{ is } A_d^k \quad \text{then}$$

$$y^k = a_0^k + a_1^k x_1 + \ldots + a_d^k x_d = (a^k) * [1, \varphi(x)^T]^T, \quad k = 1, \ldots, K. \tag{1}$$

Input data: Training sample $x = [x_1, x_2, \ldots, x_N]^T$ and appropriate for it set of labels, plural $y = [y_1, y_2, \ldots, y_N]^T \in R^N$, where $x_i \in R^d$, d – length of sample (number of elements in the plural), $y_i \in R, i = 1, 2, \ldots, N$; L – number of layers in system TSK_DBN, initial value $L = 2$; K – number of rules.

Input: Structure TSK_DBN with all relevant parameters.

Algorithm

Step 1. Calculate the matrix of fuzzy membership function by the FCM algorithm, the clustering center and width are obtained by the formulas:

$$v_i^k = \frac{\sum_{j=1}^{N} \mu_{jk} x_{ji}}{\sum_{j=1}^{N} \mu_{jk}},$$

$$\delta_i^k = \frac{h \sum_{j=1}^{N} \mu_{jk} (x_{ji} - v_i^k)^2}{\sum_{j=1}^{N} \mu_{jk}},$$

$$w_{A_i^k}(x_i) = \exp\left(\frac{-(x_i - v_i^k)^2}{2\delta_i^k} \right), \tag{2}$$

$$w(x) = \frac{w^k(x)}{\sum_{k=1}^{K} w^k(x)}. \tag{3}$$

The degree of fuzzy access is obtained using the (1), and (2), then determined.

Step 2. Study samples are entered into an adaptive submodule. The RBM training algorithm is used to extract the implicit function.

Step 3. The output of the last hidden layer in RBM is introduced into the fuzzy TSK system as an implicit function. The form of fuzzy rules TSK_DBN is considered in the formula (2).

$$x^k = \mu(x) x_e.$$

By formula (3) can be constructed matrix **W.**

Step 4. In accordance with the theory of LLM (less learning machine), the successive parameters of the fuzzy system are calculated:

$$a_g = \left(\frac{1}{C} \mathbf{I} + W^T W \right)^{-1} W^T y,$$

where C is the regularization parameter; **I** is the matrix of identity.

Step 5. Calculate the output layer using $y^0 = a_g x_g$.

Step 6. $L = L + 1$ increase the number of layers in the system

Step 7. If the performance of the system does not change or decreases after adding the layer, then stop adding the layers and proceed to step 8; otherwise go to step 2.

Step 8. Return to the structure TSK_DBN with all the parameters received.

Experiments were carried out on the sets of data presented in Table I. Experiments result given in Table II.

TABLE I. DATASETS

DATASETS	SAMP-LES	ATTRIBU-TES	CLASSES
Breast (BRE)	699	10	2
Vote (VOT)	435	16	2
Seismic_bumps (SEI)	2584	19	2
Segment (SEG)	2310	18	7

TABLE II. RESULTS

DATASETS	TSK	DBN	TSK_DBN
Breast (BRE)	0.9693	0.9703	0.9714
	0.9689	0.9643	**0.9704**
Vote (VOT)	0.9620	0.9515	0.9604
	0.9413	0.9367	**0.9532**
Seismic_bumps (SEI)	0.9356	0.9350	0.9344
	0.9310	0.9319	**0.9337**
Segment (SEG)	0.7763	0.9262	0.8530
	0.7565	**0.9207**	0.8364

VI. CONCLUSIONS

As was shown, the integration of the fuzzy TSK network and deep RBM-based network yields results. The new network is better than the usual fuzzy classifier, compared with DBN, the results may be either better or worse. The accuracy of the results depends on the sample provided to the network.

Analyzing these results, generate next conclusions that the result of the TSK_DBN network depends on the number of rules it generates and the source classes that defines the sample. If the number of classes is large, the network begins to create a large number of rules and if it is not regulated, the network's accuracy becomes worse, so the maximum number of rules for each dataset is regulated by the network user. If the set does not have a large number of output classes, the network shows itself better than its predecessors, which allows it to be used in applied spheres of science.

REFERENCES

[1] G. E. Hinton, and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science,* 28 July 2006, vol. 313, pp. 504–507. www.sciencemag.org

[2] G. E. Hinton, S. Osindero, and Y. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, 18, pp. 1527–1554, 2006.

[3] Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, 2012.

[4] Gers, Schmidhuber, and Cummins, “Learning to Forget: Continual Prediction with LSTM,” 1999.

[5] Hochreiter and Schmidhuber. “Long short-term memory,” 1997.

[6] M. Sugeno and G. T. Kang, “Structure identification of Fuzzy Model,” *Fuzzy Sets and Systems*, vol. 28, Issue 1, October 1988, pp. 15–33.

[7] Goodfellow et al. “Generative Adversarial Networks,” 2014.

[8] Gehring, et al. “A Convolutional Encoder Model for Neural Machine Translation,” 2016.

[9] Nogueira dos Santos and Gatti, “Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts,” 2014.

[10] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems 19 (NIPS'06)*, (B. Schölkopf, J. Platt, and T. Hoffman, eds.), pp. 153–160, MIT Press, 2007.

[11] Wang Shitong and Korris Fu-Lai Chung, “On Least Learning Machine,” *Journal of Jiangnan University (Natual Science Edition)*, vol. 9, pp. 505–510, Feb. 2010.

[12] Ta Zhou, Fu-lai Chung, and Shitong Wang, “Deep TSK Fuzzy Classifier with Stacked Generalization and Triplely Concise Interpretability Guarantee for Large Data,” *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 813–826, August 2016.

[13] D. Rutkowska, “Neuro-fuzzy Architectures and Hybrid Learning,” *Physica Verlag*, New York, 2002.

[14] Andrew Hoblitzell, Meghna Babbar-Sebens, Snehasis Mukhopadhyay, “Fuzzy and deep learning approaches for user modeling in wetland design,” 2016, *IEEE International Conference on Systems, Man, and Cybernetics(SMC).*

[15] G. E. Hinton, “A Practical Guide to Training Restricted Boltzmann Machines,” http://learning.cs.toronto.edu, Aug. 2010.

[16] G. E. Hinton, S. Osindero and Y. W. The, "A faster learning algorithm for deep belief nets," *Neural Comput*, vol.1, no.7, pp.1527–1544, 2006

[17] R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, M. Steinbrecher, and P. Held, *Computational Intelligence. A Methodological Introduction*. Berlin: Springer, 2013.

[18] Zhaohong Deng, Longbing Cao, Yizhang Jiang and Shitong Wang, "Minimax Probability TSK Fuzzy System Classifier: A More Transparent and Highly Interpretable Classification Model," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 813–826, Apr. 2015.

[19] Simon Haykin: *Adaptive Filter Theory,* Prentice Hall, 2002.

[20] J.-S.R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man and Cybernetics*, 1993.

Received April 28, 2019

Sineglazov Victor. orcid.org/0000-0002-3297-9060
Doctor of Engineering Science. Professor. Head of the Department.
Aviation Computer-Integrated Complexes Department, Faculty of Air Navigation Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine.
Education: Kyiv Polytechnic Institute, Kyiv, Ukraine, (1973).
Research area: Air Navigation, Air Traffic Control, Identification of Complex Systems, Wind/Solar power plant.
Publications: more than 600 papers.
E-mail: svm@nau.edu.ua

Roman Koniushenko, Bachalour.
Aviation Computer-Integrated Complexes Department, Faculty of Air Navigation Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine.
Education: National Aviation University, Kyiv, Ukraine, (2019).
Research area: neural networks.
E-mail: romankonyushenko@gamil.com

В. М. Синєглазов, Р. С. Конюшенко. Нечіткий класифікатор глибокого навчання

В даній роботі розглянуто рішення проблеми класифікаці. Показано, що нейронні мережі мають важливі переваги поряд з іншими методами, такими як: класифікація з використанням методу найближчого сусіда, класифікація за допомогою векторів підтримки, класифікація з використанням дерев рішень, тощо. Рішення може бути розширено за допомогою глибокого підходу до навчання, що передбачає використання додаткової нейронної мережі (глибокі мережі) для вирішення задачі попереднього навчання. Запропонована нова тофологія складається з: нечітких класифікаторів Такагі-Сугено-Канга і нейронної мережі обмеженої машини Больцмана. Незважаючи на те, що ця топологія була запропонована раніше, в цій статті було проведено достатньо досліджень, які дозволили створити алгоритм навчання. Наведено приклад використання запропонованого алгоритму.

Ключові слова: нейронна мережа; нечітка нейронна мережа; глибоке навчання.

Синєглазов Віктор Михайлович. orcid.org/0000-0002-3297-9060
Доктор технічних наук. Професор. Завідувач кафедрою.
Кафедра авіаційних комп'ютерно-інтегрованих комплексів, Факультет аеронавігації електроніки і телекомунікацій, Національний авіаційний університет, Київ, Україна.
Освіта: Київський політехнічний інститут, Київ, Україна, (1973).
Напрям наукової діяльності: аеронавігація, управління повітряним рухом, ідентифікація складних систем, вітроенергетичні установки.
Кількість публікацій: більше 600 наукових робіт.
E-mail: svm@nau.edu.ua

Конюшенко Роман Сергійович. Бакалавр.
Кафедра авіаційних комп'ютерно-інтегрованих комплексів, Факультет аеронавігації електроніки і телекомунікацій, Національний авіаційний університет, Київ, Україна.
Освіта: Національний авіаційний університет, Київ, Україна, (2019).
Напрям наукової діяльності: нейронні мережі.
E-mail: romankonyushenko@gamil.com

В. М. Синєглазов, Р. С. Конюшенко. Нечеткий классификатор глубокого обучения

Рассмотрено решение проблемы классификации на основе анализа представленного обзора. Показано, что нейронные сети обладают важными преимуществами по сравнению с другими методами, такими как:

классификация с использованием метода ближайшего соседа, классификация векторов поддержки, классификация с использованием деревьев решений и т. д. Существует множество искусственных нейронных сетей, которые имеют простейшую структуру, но точность решения может быть увеличена с помощью подхода глубокого обучения, который предполагает использование дополнительной нейронной сети для решения задач предварительной подготовки (сети глубокого обучения). Предложена новая топология, которая состоит из: нечеткого классификатора Такаги–Сугено–Канга и нейронной сети ограниченной машины Больцмана. Несмотря на то, что эта топология была предложена в этой статье, было проведено достаточно исследований, которые позволили создать новый алгоритм обучения. Приведен пример использования предложенного алгоритма.

Ключевые слова: нейронная сеть; нечеткая нейронная сеть; глубокое обучение.

Синеглазов Виктор Михайлович. orcid.org/0000-0002-3297-9060

Доктор технических наук. Профессор. Заведующий кафедрой.

Кафедра авиационных компьютерно-интегрированных комплексов, Факультет аэронавигации электроники и телекоммуникаций, Национальный авиационный университет, Киев, Украина.

Образование: Киевский политехнический институт, Киев, Украина, (1973).

Направление научной деятельности: аэронавигация, управление воздушным движением, идентификация сложных систем, ветроэнергетические установки.

Количество публикаций: более 600 научных работ.

E-mail: svm@nau.edu.ua

Конюшенко Роман Сергеевич. Бакалавр.

Кафедра авиационных компьютерно-интегрированных комплексов, Факультет аэронавигации электроники и телекоммуникаций, Национальный авиационный университет, Киев, Украина.

Образование: Национальный авиационный университет, Киев, Украина, (2019).

Направление научной деятельности: нейронные сети.

E-mail: romankonyushenko@gamil.com