UDC 004.891.3 **doi: 10.26906/SUNZ.2023.3.070**

Yevgeniy Bodyanskiy, Olha Chala

Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

ADAPTIVE DOUBLE NEO-FUZZY NEURON AND ITS COMBINED LEARNING

Abstract. The **subject of the study** in the article is the process of data classification under conditions of fuzziness and a limited volume of training sample. **The goal** is to enhance the double neo-fuzzy neuron within the framework of solving the data classification task with constraints on the training sample volume, processing time, as well as fuzziness and non-stationarity of input data. **The tasks** include improving the double neo-fuzzy neuron to enhance the system's approximation properties and developing a combined system learning method to ensure fast performance in an online mode. **The approaches** used are lazy learning, supervised learning, and self-learning. The following **results** have been obtained: the double neo-fuzzy neuron has been modified by introducing a compressive activation function at the output, creating conditions for building a neo-fuzzy network capable of adapting to non-stationary input data in an online mode and avoiding the vanishing gradient problem. **Conclusion**. A combined learning method for the double neo-fuzzy neuron has been proposed, involving parallel utilization of lazy learning, supervised learning, and self-learning with the "Winner Takes All" rule, followed by automatic formation of membership functions, enabling fast online classification in the presence of outliers in the input data.

Keywords: fast data classification, online classification, fuzzy classification, short sample, neo-fuzzy neuron, combined learning.

Introduction

Today, artificial neural networks (ANN) have been widely used to solve a wide range of problems related to Data Mining, primarily due to their universal approximating capabilities and their capacity to adjust their parameters (learn) by optimizing the adopted objective function (learning criterion). The main “building block” of ANN typically comprises elementary perceptrons by F. Rosenblatt, each using a particular nonlinear activation function (most referred to as the “squashing” activation function). Often, no more than three layers are required to achieve the necessary level of approximation accuracy from a different perspective. However, the training of ANN with a squashing activation function encounters significant computational challenges, often referred to as the “vanishing gradient” problem, which leads to a halt in the training process, causing network “paralysis”. Consequently, modern deep neural networks (DNN) [1–3] have abandoned squashing functions (such as sigmoid and tanh) in favour of piecewise linear functions (ReLU, PReLU, etc.) whose derivatives are commonly used as activation functions, most of which do not yield zero values.

Since such activation functions do not fulfil the conditions of G. Cybenko's approximation theorem [4], ensuring the required accuracy necessitates DNNs to encompass a substantial number of layers, neurons, and synoptic weights. Consequently, this demands an expansion in the volume of training samples and setup time. Although the application of adaptive piecewise linear activation functions can expedite the learning process [5], the need for substantial amounts of training data persists.

This situation can potentially be enhanced through the utilization of advanced nodes in lieu of traditional neurons, with one such alternative being the neo-fuzzy neuron (NFN) [6, 7]. The advantages of neo-fuzzy neurons have been showcased in addressing various problems [8, 9]. What sets a neo-fuzzy neuron apart is its integration of nonlinear synapses in place of traditional synaptic weights, with each synapse embodying the F-transform [10]. This approach facilitates universal approximating properties using a system of kernel membership functions, including conventional triangular functions that satisfy the conditions of unity partitioning. It's noteworthy that the neo-fuzzy neuron also offers piecewise linear approximation, the quality of which is significantly contingent on the count of membership functions within each nonlinear synapse.

A further progression in the evolution of the neo-fuzzy neuron is the Double Neo-Fuzzy Neuron (DNFN) [11, 12], which distinctively incorporates a non-linear synapse at its output (as opposed to an activation function), supplementing the non-linear synapses at the inputs. Importantly, the F-transform at the neuron's output can accommodate various forms of traditional activation functions, including squashing functions, without disrupting the vanishing gradient.

Both in NFN and DNFN, the number of membership functions in nonlinear synapses is predetermined (usually based on empirical considerations), evenly distributed along the abscissa axis. Enhanced approximation quality can be achieved by dynamically determining the requisite number of these functions and adapting the positioning of their centres. This necessitates an augmented training procedure that incorporates adjustments to the number and centres of membership functions. This innovation is anticipated to augment the precision of DNFN in comparison to systems with predetermined counts and shapes of activation functions - membership.

The primary objective of this article is to investigate and propose an alternative approach that overcomes the limitations encountered in traditional artificial neural networks (ANNs) and their training methodologies. In particular, we aim to address the challenges associated with the "vanishing gradient" problem, which impedes the training process of ANNs with certain activation functions. Additionally, we seek to enhance the accuracy and efficiency of the adaptive learning process within neural networks.

Architecture of the Double neo-fuzzy neuron

The DNFN architecture is shown in Fig. 1 and consists of $n+1$ nonlinear synapses NS_i, $i = 0,1,...,n$ the scheme of which is shown in Fig. 2.

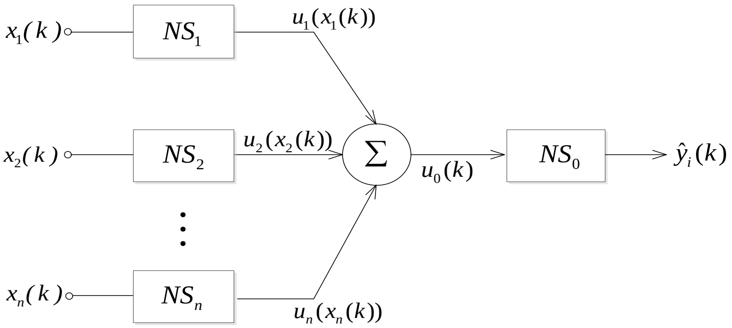

Fig. 1. Double neo-fuzzy neuron

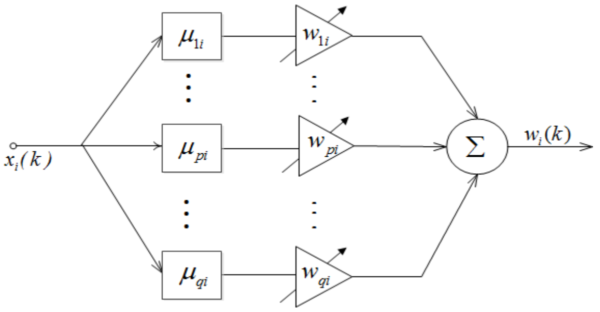

Fig. 2. Nonlinear synapse

The input information for training is a sample of vector observations $X = \{x(1), x(2),..., x(k),..., x(N)\}$, $x(k) = (x_1(k), \ ..., \ x_i(k), \ ... , \ x_n(k))^T \in R^n$, where k or the observation number in the sample, if it is specified a priori in the batch form, or the number of the moment of the current time, if the training is carried out online as the action information is received for processing. Each nonlinear synapse contains q membership functions $\mu_{ip}(u_i)$, $p = 1,...,q$; $i = 1,2,...,n$; and $\mu_{0p}(u_0)$ in the output layer and q tuned synaptic weights (one for each membership function) W_{ip} and W_{0p}. In this way, nonlinear synapses implement the transformation adaptive double neo-fuzzy neuron (DNFN) its combined learning.

$$\begin{cases} u_i(x_i(k)) = \sum_{p=1}^{q} \mu_{pi}(x_i(k)) w_{pi}(k), \\ \hat{y}(k) = \sum_{p=1}^{q} \mu_{p0}(u_0(k)) w_{pi}(k). \end{cases} \quad (1)$$

It is also easy to write down the transformations implemented by DNFN as a whole:

$$\begin{cases} u_{0i}(k) = \sum_{i=1}^{\Pi} \sum_{p=1}^{q} \mu_{pi}(x_i(k)) w_{pi}(k), \\ \hat{y}(k) = \sum_{p=1}^{q} \mu_{p0} \sum_{i=1}^{\Pi} \sum_{p=1}^{q} \mu_{pi}(x_i(k)) w_{pi}(k) w_{p0}(k)\Big). \end{cases} \quad (2)$$

Triangular ones satisfying the Ruspini conditions are most often used as the membership function in NFN, due primarily to numerical simplicity, although the use of other functions, such as B-Splines, is not excluded.

If the input data is pre-encoded on the interval [0, 1], and their centers c_{pi} and c_{p0} evenly spaced on the abscissa axis $\Delta = c_{p+1,i} - c_{pi}$, it is easy to see that $\Delta = c_{p+1,i} - c_{pi} = (q-1)^{-1}$, namely, these functions can be written in the form

$$\mu_{1i}(x_i) = \begin{cases} (c_{2i} - x_i)c_{2i}^{-1} \ \ if \ \ x_i \in [0, c_{2i}], \\ 0 \ \ otherwise, \end{cases} \quad (3)$$

$$\mu_{pi}(x_i) \begin{cases} (x_i - c_{p-1,i})(c_{pi} - c_{p-1,i})^{-1} \\ \qquad if \ \ x_i \in [c_{p-1,i}, \ c_{pi}], \\ (c_{p+1,i} - x_i)(c_{p+1,i} - c_{pi})^{-1} \\ \qquad if \ \ x_i \in [c_{pi}, c_{p+1,i}], \\ 0 \ \ otherwise, \end{cases} \quad (4)$$

$$\mu_{qi}(x_i) \begin{cases} (x_i - c_{q-1,i})(1 - c_{q-1,i})^{-1} \\ \qquad if \ \ x_i \in [c_{q-1,i}, \ i,1], \\ 0 \ \ otherwise. \end{cases} \quad (5)$$

for the input layer of nonlinear synapses. For a nonlinear synapse DNFN can be written

$$\mu_{10}(u) \begin{cases} (c_{20} - u_0)c_{20}^{-1} \ \ if \ \ u_0 \in [0, c_{20}], \\ 0 \ \ otherwise, \end{cases} \quad (6)$$

$$\mu_{p0}(u) \begin{cases} (u_0 - c_{p-1,0})(c_{p0} - c_{p-1,0})^{-1} \\ \qquad if \ \ u_0 \in [c_{p-1,0}, \ c_{p0}], \\ (c_{p+1,0} - u_0)(c_{p+1,0} - c_{p0})^{-1} \\ \qquad if \ \ u_0 \in [c_{p0}, c_{p+1,0}], \\ 0 \ \ otherwise \end{cases} \quad (7)$$

$$\mu_{p0}(u)\begin{cases}(u_0-c_{p-1,0})(1-c_{p-1,0})^{-1} \\ \qquad if\ u_0\in[C_{p-1,0},1], \\ 0\ \ otherwise\end{cases} \quad (8)$$

a feature of the membership function used in DNFN is that at each time instant k only two neighbouring functions are-fired. Suppose that the component of the input vector *x(k)* belongs to the interval $C_{pi}<x_0(k)<C_{p+1,i}$, that is, only two adjacent functions are fired $\mu_{pi}(x_i(k))$ and $\mu_{p+1,i}(x_i(k))$.

It is easy to calculate that the signal at the input of the i-th input NS_i can be represented in the form:

$$\begin{aligned}&u_i(x_i(k))=\sum_{p=1}^{q}\mu_{pi}(x_i(k))w_{pi}(k)=\\&=\mu_{pi}(x_i(k))w_{pi}(k)+\mu_{p+1,i}(x_i(k))w_{p+1,i}(k)=\\&=(c_{p+1,i}-x_i(k))(c_{p+1,i}-c_{pi})^{-1}w_{pi}(k)+\\&+(x_i(k)-c_{pi})(c_{p+1,i}-c_{pi})^{-1}w_{p+1,i}(k)=\\&=a_i(k)x_i(k)+b_i(k)\end{aligned} \quad (9)$$

where

$$\begin{cases}a_i(k)=(w_{p+1,i}(k)-w_{pi}(k))(c_{p+1,i}-c_{pi})^{-1}\\ b_i(k)=\begin{pmatrix}c_{p+1,i}w_{pi}(k)-\\-c_{pi}w_{p+1,i}(k)\end{pmatrix}(c_{p+1,i}-c_{pi})^{-1}.\end{cases} \quad (10)$$

and the signal at the adder input:

$$u_0(k)=\sum_{i=1}^{n}a_i(k)x_i(k)+b_i(k). \quad (11)$$

Then the DNFN output signal in general can be written as

$$\begin{aligned}&\hat{y}(k)=\sum_{p=1}^{q}\mu_{p0}(u_0(k))w_{p0}(k)=\\&=\mu_{p0}(u_0(k))w_{p0}(k)+\mu_{p+1,0}(u_0(k))w_{p+1,0}(k)=\\&=(c_{p+1,0}-u_0(k))(c_{p+1,0}-c_{p0})^{-1}w_{p0}(k)+\\&+(u_0(k)-c_{p0})(c_{p+1,0}-c_{p0})^{-1}w_{p+1,0}(k)=\\&=a_0(k)u_0(k)+b_0(k)\end{aligned} \quad (12)$$

where

$$\begin{cases}a_0(k)=(w_{p+1,0}(k)-w_{p0}(k))(c_{p+1,0}-c_{p0})^{-1}\\ b_0(k)=\begin{pmatrix}c_{p+1,0}w_{p0}(k)-\\-c_{p0}w_{p+1,0}(k)\end{pmatrix}(c_{p+1,0}-c_{pi})^{-1}.\end{cases} \quad (13)$$

It is easy to see that NS_0 at the DNFN output does not play the role of an activation function in F. Rosenblatt's neuron, and in some sense is close to the PReLU function popular in DNN, however, in our case, this activation function is configured, implements the F-transformation and can accept you are a complicated shape.

Finally, the transformation implemented by DNFN can be implemented in the form

$$\hat{y}(k)=a_0(k)(\sum_{i=1}^{n}a_i(k)x_i(k)+b_i(k))+b_0(k)\,, \quad (14)$$

that is, a piecewise linear approximation of some ringing function is provided, the quality of which depends on the number of readings (belonging functions) q, the location of the centres of these functions and *(n+1)q* synaptic weights, which is adjusted during the learning process.

A training procedure of the Double neo-fuzzy neuron

The tuning of DNFN synoptic weights is implemented by gradient minimization (tutored learning) of the objective function, which is commonly used in training and ANNs.

$$\begin{aligned}&E(k)=(y(k)-\hat{y}(k))^2=(y(k)-\\&-\sum_{p=1}^{q}\mu_{p0}\left(\sum_{i=1}^{n}\sum_{p=1}^{q}\mu_{pi}(x_i(k))w_{pi}\right)w_{p0})^2=\\&=(y(k)-a_0\left(\sum_{i=1}^{n}a_ix_i(k)+b_i)-b_0\right)\end{aligned} \quad (15)$$

where $y(k)$ -external reference signal.

The learning process at each moment k is implemented in two stages: adjustment of the synoptic weights of the original nonlinear synapse NS_0 setting the weights of input signals NS_i, $i=1,2,...,n.$

At the same time, it is noticeable that only the weights corresponding to the membership functions are adjusted μ_{pi} and $\mu_{p+1,i}$, which are currently k have zero values.

At the same time, the learning process can be left as it is

$$\begin{cases}w_{l0}(k+1)=w_{l0}(k)-\eta_p(k)l(k)\mu_{l0}(u(k)),\\ l=p,p+1\\ w_{l0}(k+1)=w_{l0}(k)\ \ \forall l\neq p\neq p+1,\end{cases} \quad (16)$$

where $l(k)=y(k)-\hat{y}(k)$ – learning error, $\eta_0(k)>0$ – learning rate parameter, which means speed of convergence.

This process can be optimized for speed by using a modification of the Kaczmarz-Widrow-Hoff algorithm [13, 14]:

$$\begin{cases}w_{p0}(k+1)=w_{p0}(k)+\\ \qquad +\dfrac{l(k)\mu_{p0}(u(k))}{\mu^2{}_{p0}(u(k))+\mu^2{}_{p+1,0}(u(k))},\\ w_{p+1,0}(k+1)=w_{p+1,0}(k)+\\ \qquad +\dfrac{l(k)\mu_{p+1,0}(u(k))}{\mu^2{}_{p0}(u(k))+\mu^2{}_{p+1,0}(u(k))}.\end{cases} \quad (17)$$

In the case when the input data is disturbed by interference, an algorithm with additional filtering properties can be used [15]

$$\begin{cases} w_{p0}(k+1) = w_{p0}(k) + \\ \quad + \eta_0^*(k) l(k) \mu_{p0}(u(k)), \\ w_{p+1,0}(k+1) = w_{p+1,0}(k) + \\ \quad + \eta_0(k) l(k) \mu_{p+1,0}(u(k)), \\ \eta_0^*(k) = (\eta_0 \eta_0^*(k-1) + \mu^2{}_{p0}(u(k)) + \\ \quad + \mu^2{}_{p+1,0}(u(k)))^{-1}. \end{cases} \tag{18}$$

where $0 \le \eta_0 \le 1$ is a forgetting factor.

To adjust the synaptic weights of the first layer of non-linear synapses NS_i, a standard S-rule training method can be used.

Let us consider the derivatives of the objective function.

$$\frac{\partial E(k)}{\partial w_{li}} = -l(k) \frac{\partial \hat{y}(k)}{\partial u_0(k)} \cdot \frac{\partial u_0(k)}{\partial w_{li}} = = -l(k) a_0(k) \frac{\partial u_0(k)}{\partial w_{li}} \tag{19}$$

and δ is an error.

$$\delta(k) = l(k) a_0(k). \tag{20}$$

It is easy to write down the gradient learning procedure

$$\begin{cases} w_{li}(k+1) = w_i(k) + \eta_i\ (k) \delta(k) \mu_{li}(x_i(k)), \\ l = p, p+1, \\ w_{li}(k+1) = w_{li}(k)\, \forall l \ne p \ne p+1. \end{cases} \tag{21}$$

Optimized versions of this procedure take shape accordingly

$$\begin{cases} w_{pi}(k+1) = w_{pi}(k) + \\ + \dfrac{\delta(k) \mu_{pi}(x_i(k))}{\mu^2{}_{pi}(x_i(k)) + \mu^2{}_{p+1,i}(x_i(k))}, \\ w_{p+1,i}(k+1) = w_{p+1,i}(k) + \\ + \dfrac{\delta(k) \mu_{p+1,i}(x_i(k))}{\mu^2{}_{pi}(x_i(k)) + \mu^2{}_{p+1,i}(x_i(k))}. \end{cases} \tag{22}$$

and

$$\begin{cases} w_{pi}(k+1) = w_{pi}(k) + \\ \quad + \eta_i^*(k) \delta(k) \mu_{pi}(x_i(k)), \\ w_{p+1,i}(k+1) = w_{p+1,i}(k) + \\ \quad + \eta_i^*(k) \delta(k) \mu_{p+1,i}(x_i(k)), \\ \eta_i^*(k) = (\eta_i\, \eta_i^*(k-1) + \mu_{pi}^2(x_i(k)) + \\ \quad + \mu_{p+1,i}^2(x_i(k))^{-1}, \\ 0 \le \eta_i \le 1. \end{cases} \tag{23}$$

It is possible to improve the approximate properties of DNFN by adjusting not only the synoptic weights of linear synapses, but also the number of membership functions and the location of their centres. Self-learning methods without a teacher and lazy learning based on the principle of "Neurons at data points" [16] can be used for this.

Usually, the number of membership functions in non-linear synapses q is given by p empirical considerations and the weights are placed uniformly with an interval Δ. Let us consider the threshold of indistinguishability of two neighbouring centres $\delta << \Delta$ let's start the learning process with the entry into the system of the first observation of the training sample $x(1) = x_1(1),\ \ldots\ ,\ x_i(1),\ \ldots\ x_n(k))^T$. The first function of belonging x $\mu_{1,i}(x_i(1))$ is formed so that its centre $c_{1,i}(x_i(1))$. Then with the arrival of the second observation $x(2)$ the condition is checked $|\, x_i(2) - c_{1i}(2)\,| \le \delta$ and if it is performed for some components of the input signals, then new centres are not formed at these inputs.

If the condition is fulfilled (on several inputs) $\delta <|\, x_i(2) - c_{1i}\,|< 2\delta$, coefficients of centers $c_{1,i}$ adjusted according to the self-learning rule "Winner Takes All" [17], introduced by T. Kohonen

$$c_{1i}(2) = c_{1i}(1) + \eta_0(2)(x_i(2) - c_{il}(1)). \tag{24}$$

With the learning threshold parameter $\eta_0(2) < 1$ if the inequality holds for some components $2\delta < x_i(2) - c_{1i}(2)$, then the second membership function is formed at the i-th input μ_{qi} with centre $c_{2i} = x_i(2)$. During the second learning cycle, the number of membership functions in synapses NS_i may be different. On the N-th self-learning step with input signal $x(N)$ membership functions are solved for each nonlinear synapsee the membership function- winner is sought $\mu_{pi}^*(x_i(N-1))$ for which the distance $|\, x_i(N) - c_{pi}(N-1)\,|$ is minimal and the following rules are checked:

$$\begin{cases} |\, x_i(N) - c_{pi}^*(N-1)\,| \le \delta, \\ \delta <|\, x_i(N) - c_{pi}^*(N-1)\,| \le 2\delta, \\ 2\delta <|\, x_i(N) - c_{pi}^*(N-1)\,| \end{cases} \tag{25}$$

after which at the i-th input in the linear synapse NS_i or is not formed or adjusted or a new (and last) membership function is created $\mu_{qi}(x_i)$ with the centre C_{qi}. Thus, in each nonlinear synapse, the number of membership functions can vary from two (in the case of binary input signals) to N (in the case of a small indistinguishability threshold δ. Similarly, the membership functions of the nonlinear synapse can be configured NS_0 at the output where DNFN.

Conclusion

To enhance the widely known neo-fuzzy neuron (NFN) referred to as the Double neo-fuzzy neuron, which exhibits enhanced approximative capabilities when compared to its prototype, a novel combined learning approach is introduced. This approach is grounded in tutored learning, self-learning, and lazy learning principles. Through this methodology, the aim is to achieve optimal rapidity in configuring the synoptic weights of nonlinear synapses, along with the automatic formation of membership functions, all implemented in real-time online mode.

Importantly, the proposed method doesn't necessitate substantial amounts of training data. Instead, it's characterized by its computational simplicity, rendering it applicable in diverse scenarios. This method holds promise in creating a double neo-fuzzy system capable of effectively adapting to phasing systems, even within the constraints of a limited training dataset.

References

1. Aggarwal CC (2018) Neural networks and deep learning: a textbook. Springer, Cham, Switzerland
2. Bengio Y, Cun YL, Hinton G (2015) Deep learning. Nature 521:436–444
3. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press
4. Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signal Systems 2:303–314. https://doi.org/10.1007/BF02551274
5. Bodyanskiy YV, Tyshchenko OK (2020) A hybrid cascade neural network with ensembles of extended neo-fuzzy neurons and its deep learning. Advances in Intelligent Systems and Computing 945:164–174
6. Yamakawa J, Uchino E, Miki J, Kusanagi H (1992) A neo-fuzzy neuron and its application to system identification and prediction of the system behavior. In: Proceedings of the 2nd International Conference on Fuzzy Logic & Neural Networks. Iizuka, Japan
7. Miki T (1999) Analog Implementation of Neo-Fuzzy Neuron and Its On-board Learning
8. Zurita D, Delgado M, Carino JA, et al (2016) Industrial Time Series Modelling by Means of the Neo-Fuzzy Neuron. IEEE Access 4:6151–6160. https://doi.org/10.1109/ACCESS.2016.2611649
9. Bodyanskiy YV, Kokshenev I, Kolodyazhniy V (2003) An adaptive learning algorithm for a neo fuzzy neuron. In: Wagenknecht M, Hampel R (eds) Proceedings of the 3rd Conference of the European Society for Fuzzy Logic and Technology, Zittau, Germany, September 10-12, 2003. University of Applied Sciences at Zittau/Görlitz, Germany, pp 375–379
10. Perfilieva I (2006) Fuzzy transforms: Theory and applications. Fuzzy Sets and Systems 157:993–1023. https://doi.org/10.1016/j.fss.2005.11.012
11. Bodyanskiy Dr.-Ing. habil. Y, Gorshkov M.Sc. Y, Kolodyazhniy Dr.-Ing. V, et al (2007) Evolving network based on double neo-fuzzy neurons. Computer science meets automation: 52 IWK, Internationales Wissenschaftliches Kolloquium ; proceedings ; 10 - 13 September 2007; Volume I 52, 2007, Vol. 1:35–40
12. Hu Z, Bodyanskiy YV, Tyshchenko OK, Samitova VO (2017) Fuzzy clustering data given on the ordinal scale based on membership and likelihood functions sharing. International Journal of Intelligent Systems and Applications 9:1–9
13. Kaczmarz S (1993) Approximate solution of systems of linear equations. International Journal of Control 57:1269–1271. https://doi.org/10.1080/00207179308934446
14. Widrow B, Hoff ME (1960) Adaptive Switching Circuits. IRE WESCON Convention Record
15. Bodyanskiy Y, Kolodyazhniy V, Stephan A (2001) An Adaptive Learning Algorithm for a Neuro-fuzzy Network. Computational Intelligence Theory and Applications Fuzzy Days 2206:68–75
16. Zahirniak DR, Chapman R, Rogers SK, et al (1990) Pattern recognition using radial basis function network. Aerospace Application of Artificial Intelligence 249–260

Received (Надійшла) 16.07.2023
Accepted for publication (Прийнята до друку) 06.09.2023

Адаптивний подвійний нео-фаззі нейрон та його комбіноване навчання

Є. В. Бодянський, О. С. Чала

Анотація. Предметом вивчення в статті є процес класифікації даних за умов нечіткості та обмеженого об'єму навчальної вибірки. **Метою** є удосконалення подвійного нео-фаззі нейрона в рамках вирішення задачі класифікації даних із обмеженнями щодо об'єму тренувальної вибірки, часу обробки, а також нечіткості та не стаціонарності вхідних даних. **Завдання**: удосконалення подвійного нео-фаззі нейрона для покращення апроксимаційних властивостей системи, а також розробка комбінованого методу навчання системи для забезпечення швидкої в онлайн режимі. Використовуваними **підходами** є: ліниве навчання, навчання з учителем та самонавчання. Отримані наступні **результати**. Модифіковано подвійний нео-фаззі нейрон, запропоновано метод комбінованого навчання, що забезпечує оптимальну швидкість при налаштуванні синоптичних ваг та автоматичне формування функцій належності в онлайн-режимі за умов обмеженої навчальної вибірки. **Висновки**. Удосконалено подвійний нео-фаззі нейрон шляхом введенням стискаючої активаційної функції на виході, що створює умови для побудови нео-фаззі мережі з можливістю адаптації до нестаціонарних вхідних даних за умови роботи в онлайн режимі, а також уникнути проблеми зникаючого градієнту. Запропоновано комбінований метод навчання подвійного нео-фаззі нейрону, який передбачає паралельне використання лінивого навчання, навчання з учителем та самонавчання за правилом «Переможець забирає все» з подальшим автоматичним формуванням функцій належності, що дає можливість швидкої класифікації в режимі онлайн за умови наявності викидів у вхідних даних.

Ключові слова: швидка класифікація даних, класифікація в онлайн режимі, нечітка класифікація, коротка вибірка, нео-фаззі нейрон, комбіноване навчання.