IEEE Second International Conference on Data Stream Mining & Processing
August 21-25, 2018, Lviv, Ukraine

A Hybrid Neuro-Fuzzy Element: a New Structural Node for Evolving Neuro-Fuzzy Systems

Zhengbing Hu
School of Educational Information Technology
Central China Normal University
Wuhan, China
hzb@mail.ccnu.edu.cn

Oleksii K. Tyshchenko[1,2]
[1]*Institute for Research and Applications of Fuzzy Modeling, CE IT4Innovations, University of Ostrava*
Ostrava, Czech Republic
[2]*Control Systems Research Laboratory, Kharkiv National University of Radio Electronics*
Kharkiv, Ukraine
lehatish@gmail.com

***Abstract* — A modification of the structure for a neuro-fuzzy unit was offered which is generally a hybrid system that combines nonlinear synapses and an activation function to form the hybrid system's output value. The introduced neuro-fuzzy element is specifically an extension of the common neo-fuzzy neuron which is upgraded at the expense of application of an additional (contracting) activation function. A particular robust learning procedure is also considered for this case that makes it possible to reduce errors while processing data containing abnormal observations.**

Keywords* — *Learning Method; Evolving System; Computational Intelligence; Neuro-Fuzzy Unit; Data Stream Processing; Hybrid Neuro-Fuzzy System; Machine Learning.

I. INTRODUCTION

Multiple real-world systems and applications generate huge sequences of observations (data flows/streams) that arrive sequentially at a high speed. Data analysis should be brought into play in real time using limited storage and computing capabilities. As far as is known, Data Stream Mining [1-5] deals with extracting knowledge structures from continuous rapid data. Hybrid Computational Intelligence systems [6-10] like neuro-fuzzy systems, common neural networks, and wavelet neuro-fuzzy systems have proliferated extensively for taking decisions on a vast class of challenges [8-10] that come into existence within Data Mining in relation to their universal fitting properties and their aptitude of linguistic interpretation for obtained results. Most of these systems have proven their efficacy in problems of intensively developing Data Stream Mining [3-5], where information to be processed is fed sequentially (an observation by an observation), and a tuning process of the system's parameters should be carried out in an online mode by means of adaptive learning algorithms. First of all, we should mention here such systems where an output signal depends linearly on parameters to be tuned as radial-basis function neural networks [11-20], neuro-fuzzy systems by Takagi-Sugeno-Kang [21-22], hybrid systems that use neo-fuzzy neurons [23-30] as their nodes.

Experts' attention in the area of Computational Intelligence has been attracted to deep neural networks [31-39] recently. These networks considerably transcend conventional shallow neural networks regarding the quality of information processing. At the same time, they are instantiated by a low rate of learning that can be explained by a necessity of using the error backpropagation method for multiple hidden layers of the network. In view of this, it seems reasonable enough to synthesize/update a structure (a computational unit) and its learning methods to be later used as a part of some more complex computational systems like evolving cascade systems and deep neural networks. A point to be noted here about the novelty is the fact that a new type of membership functions is used in synapses to raise approximating abilities of a hybrid neuro-fuzzy element as well as different types of activation functions are considered for the introduced topology based on the fundamental properties of an issue being solved. The paper is composed in such a manner. Section 2 comprises complementary information concerning a neuro-fuzzy unit and its modification. Section 3 describes a robust learning procedure applied to the system and different activation functions to be used for practical applications. Section 4 embodies experimental results of the presented neuro-fuzzy system. Conclusions are presented in the last section.

II. A MODIFICATION OF THE NEURO-FUZZY UNIT'S STRUCTURE

Ye. Bodyanskiy and S. Popov proposed in broad brushstrokes further changes [40, 41] to a topology of the neo-fuzzy neuron (NFN) [23-25]. The proposed system is intrinsically a hybrid combination of a neuro-fuzzy system (more specifically the neo-fuzzy neuron) and the elementary neuron by McCulloch and Pitts (also known as the MCP neuron). The exploited architecture eliminates shortcomings (which are typical for the neo-fuzzy neuron) at the cost of induction into the system's structure of a tightening activation function that brings some additional nonlinear effect.

That means a nonlinear calculative framework (Fig.1) which embodies nonlinear synapses (Fig.2) succeeded by a summation block and a nonlinear activation function to calculate the system's output.

Said another way, input signals are transformed with the help of synapses into the signals $f_i(x_i)$. These signals are later joined into the internal activation signal $u=\sum_{i=1}^{n} f_i(x_i)$. The neuron's output signal is made up with a nonlinear activation function $y=\psi(u)$.

As a general matter, a NFU output signal executes a mathematical transformation (based on the quadratic criterion) in a certain way

$$y=\psi(u)=\psi\left(\sum_{i=1}^{n} f_i(x_i)\right)=\psi\left(\sum_{i=1}^{n}\sum_{j=1}^{h_i} w_{ij}\mu_{ij}(x_i)\right)$$

where $\psi(\bullet)$ notes a nonlinear activation function (either a sigmoid function or a hyperbolic tangent); x_i describes input signals; $\mu_{ij}(x_i)$ marks membership levels; w_{ij} determines synaptic weights; h describes a quantity of fuzzy spans; n stands for a plurality of inputs; y is an output value.

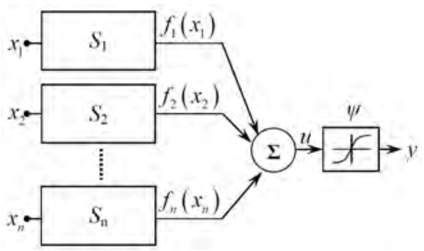

Fig. 1. A structure of the neuro-fuzzy unit

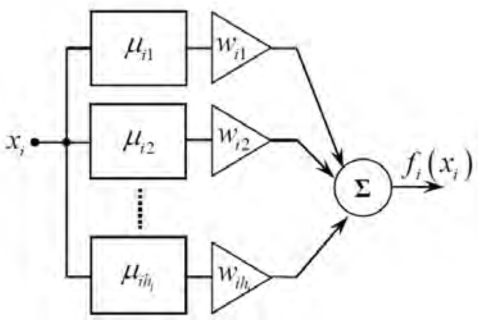

Fig. 2. A scheme of a nonlinear synapse in the neo-fuzzy neuron

On a large scale, triangular membership functions (Fig.3) stand on a distance between the input x_i and centers c_{ij}

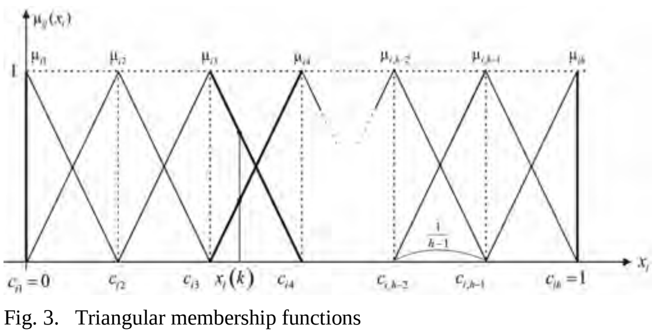

Fig. 3. Triangular membership functions

$$\mu_{i1}(x_i)=(c_{i1}-x_i)/c_{i2},$$

$$\mu_{ij}(x_i)=\begin{cases}(x_i-c_{i,j-1})/(c_{ij}-c_{i,j-1}),\ x_i\in[c_{i,j-1};c_{ij}],\\(c_{i,j+1}-x_i)/(c_{i,j+1}-c_{ij}),\ x_i\in[c_{ij};c_{i,j+1}],\\0\ \textit{otherwise.}\end{cases}\qquad(1)$$

$$\mu_{ih}(x_i)=(x_i-c_{i,h-1})/(1-c_{i,h-1}),$$
$$c_{i1}=0, c_{i2}=1/(h-1)\ c_{il}=(l-1)/(h-1), c_{ih}=1.$$

It's essentially taken all the initial data to be coded in the range [0;1]. It's crucial that this type of constructing membership functions makes automatically provision of the Ruspini (unity) partition

$$\sum_{j=1}^{h}\mu_{ij}(x_i)=1\ \ \forall i.$$

Let's hypothesize that a fuzzy interval p is currently active, an output of the nonlinear synapse may be presented in this fashion

$$f_i(x_i)=\sum_{j=1}^{h} w_{ij}\mu_{ij}(x_i)=w_{ip}\mu_{ip}(x_i)+w_{i,p+1}\mu_{i,p+1}(x_i)=$$
$$=\frac{c_{i,p+1}-x_i}{c_{i,p+1}-c_{ip}}w_{ip}+\frac{x_i-c_{ip}}{c_{i,p+1}-c_{ip}}w_{i,p+1}.$$

Having said stated above, triangular membership constructions (1) are conventionally made use of as activation functions in the neo-fuzzy neuron. It may bring some obstruction for processes' simulation exemplified by differentiable (smooth) functions. The piecewise linear fitting appeared in this case by the neo-fuzzy neuron can account for a diminished accuracy level of the results obtained. A quantity of membership functions could be increased to lessen this negative effect. But finally, it results in enlargement of a number of weight coefficients, and the structure's complexity is growing along with the learning time required. The announced drawback may be avoided by means of cubic spline membership functions to be represented as follows

$$\mu_{ij}(x_i)=\begin{cases}0.25\left(2+3\frac{2x_i-c_{ij}-c_{i,j-1}}{c_{ij}-c_{i,j-1}}-\left(\frac{2x_i-c_{ij}-c_{i,j-1}}{c_{ij}-c_{i,j-1}}\right)^3\right),\\x_i\in[c_{i,j-1};c_{ij}];\\0.25\left(2-3\frac{2x_i-c_{i,j+1}-c_{ij}}{c_{i,j+1}-c_{ij}}+\left(\frac{2x_i-c_{i,j+1}-c_{ij}}{c_{i,j+1}-c_{ij}}\right)^3\right),\\x_i\in(c_{ij};c_{i,j+1}].\end{cases}$$

At some time moment, an input signal engages only two adjoining functions (Fig.4) contemporaneously (this case is rather similar to the triangular membership functions). But the provided set of functions doesn't cater to the needs of the Ruspini partition. Conversely, application of the cubic spline functional relations actualizes smooth polynomial fitting as a substitute for piecewise linear approximation and raises the possibility of carrying out the top grade simulation of substantively nonstationary and nonlinear signals.

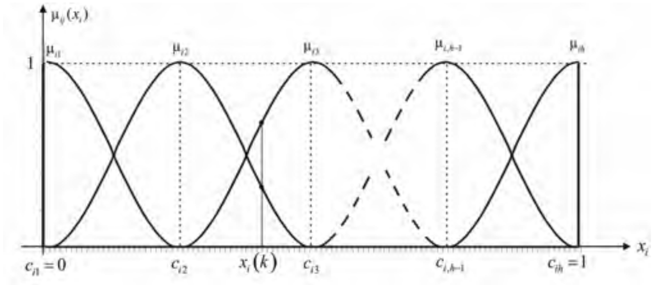

Fig. 4. Cubic spline membership functions

Giving effect to additional nonlinearity (compared to the conventional NFN) at the neuron's output quantity culminates in automatic containment of the element's output amplitude (which is especially relevant for construction of complex multilayer networks). A procedure of the NFE weights' tuning is exploited with reference to the quadratic criterion

$$\begin{aligned} E(k) &= \frac{1}{2}\big(d(k)-y(k)\big)^2 = \frac{1}{2}e^2(k) = \\ &= \frac{1}{2}\big(d(k)-\psi(u(k))\big)^2 = \\ &= \frac{1}{2}\left(d(k)-\psi\left(\sum_{i=1}^{n}\sum_{j=1}^{h_i} w_{ij}\mu_{ij}\left(x_i(k)\right)\right)\right)^2 \end{aligned} \qquad (2)$$

where k denominates a unit of discrete time; $d(k)$ marks a reference signal; $e(k)$ stands for a learning error;

$$w_i = \left(w_{i1}, w_{i2}, ..., w_{ih}\right)^T;$$
$$\mu_i\left(x_i(k)\right) = \left(\mu_{i1}\left(x_i(k)\right), ..., \mu_{ih}\left(x_i(k)\right)\right)^T.$$

In the interest of minimization of the equation (2), the gradient descent learning procedure should be applied

$$\begin{aligned} w(k+1) &= w(k) - \eta(k)\nabla_w E(k) = \\ &= w(k) + \eta(k)e(k)\frac{\partial e(k)}{\partial u(k)}\nabla_w u(k) = \\ &= w(k) + \eta(k)e(k)\psi'\big(u(k)\big)\mu\big(x(k)\big) \end{aligned} \qquad (3)$$

where $\eta(k)$ designates a learning rate.

III. A Robust Learning procedure and activation functions used

Utilizing the neuro-fuzzy element is quite challenging for signal processing. The remarkable thing is that its nonlinear properties can be set up by means of membership functions' parameters inside the nonlinear synapses. From this perspective, outliers may be put out, and an impact of less limitative input terms should get diminished noticeably. Learning methods on the grounds of the quadratic criteria (2) are highly exposed to data distribution's deviations. In the context of various types of irregular observations, learning methods based on the quadratic criteria don't demonstrate high efficiency due to obstacles with the "heavy tail" distribution and massive errors. In these cases, robust estimation methods [42] seem to be the most effective and appropriate ones [43-44]. The criterion mentioned below is very popular in the theory of robust estimation

$$E^R(k) = \beta \ln\left(\cosh\frac{e(k)}{\beta}\right) \qquad (4)$$

where $e(k)$ stands for a learning error; β denotes a scalar parameter magnitude to be chosen commonly in terms of a posteriori knowledge in order to appoint susceptibility for anomalous faults.

According to the initial paper [41], the improved learning procedure for the hybrid neuro-fuzzy element provides an opportunity to reduce processing errors for irregular samples by introducing the robust learning criterion. Through the lens of the learning procedure (3) grounded in this criterion (4) for the neuro-fuzzy element, the tweakage process of the system's weights may be represented in this view

$$\begin{aligned} w(k+1) &= w(k) - \eta(k)\nabla_w E^R(k) = \\ &= w(k) + \eta(k)\beta\tanh\frac{e(k)}{\beta}\psi'\big(u(k)\big)\mu(k) = \\ &= w(k) - \eta(k)\delta^R(k)\mu(k) \end{aligned} \qquad (5)$$

where

$$\frac{\partial E^R(k)}{\partial e} = \beta\tanh\frac{e(k)}{\beta}; \quad \delta^R(k) = \beta\tanh\frac{e(k)}{\beta}\psi'\big(u(k)\big).$$

Besides that, the sigmoid is smooth and dependent on x. The fault is suitable for backpropagation, and weights should be updated subsequently. But anyway, there several issues to be addressed with that. The curve is quite plane beyond the $[-3;3]$ interval which means that once the relationship finds the way in that bracket, its gradients start descending (the gradient is verging to zero, and the network doesn't receive any training in the actual circumstances). An alternative issue that has to do with the logistic function (a sigmoid curve) is that its meanings only gauge in $(0;1)$. This means that the sigmoid curve is not symmetric around the reference point, and the implications gained are positive. But what if there's no need to send permanently to a subsequent neuron the values to be all of the same sign. One of the possible solutions for this case is scaling the sigmoid curve. The tanh function is of the nature of the logistic function and is in sober fact just a scaled version of it. Tanh works in the same fashion compared to the sigmoid relation, but it is symmetric over the initial point and sites from -1 to 1. It principally addresses the challenge of the meanings all being of the same sign. The rest of features are identical to the logistic curve. It is continuous and differentiable at all points. The functional relation is nonlinear and may be applied easily to backpropagating errors. Speaking of the tanh gradient, it's steeper in comparison with the sigmoid function. A selection between sigmoid and tanh essentially is stipulated by the gradient precondition for a problem statement. But there's also the vanishing gradient problem (the tanh graph is plane, and the gradients obtained are close to zero). The softmax function (the normalized exponential function) is some sort of the logistic function, but it's favorable when it comes to handling classification issues. The softmax distribution would compress output implications for every group between 0 and 1 and divide by the outputs' sum. In order to get a probability distribution of outputs, the softmax function is usually put in requisition to impart probabilities when there is more than one output. It's chiefly advantageous when there is a need to find the most probable occurrence of an output with respect to other ones. Several more words should be said about a sigmoid-weighted linear unit (SiLU)

$$f(x) = \frac{x}{1+e^{-x}} = x\sigma(x)$$

where $\sigma(x)$ signifies the sigmoid function. Its derivative is

$$f'(x) = f(x) + \sigma(x)\left(1 - f(x)\right)$$

and it's generally used as a function approximator for neural networks in reinforcement learning.

The last functional relation to be mentioned is SoftPlus

$$f(x) = \ln\left(1+e^{x}\right).$$

And its derivative is the logistic (sigmoid) function. That's naturally a smooth approximation of a rectified linear unit which is broadly exploited in deep learning, computer vision, and speech recognition. All of these fore mentioned functions could be utilized as activation functions in the hybrid neuro-fuzzy element specifically from the perspective of a task type under consideration and some initial conditions of the problem.

IV. EXPERIMENTS

Theoretical aspects of our research were validated with the help of an experimental study depicting the forecasting challenge of electric loads. An available data sample comprised 6380 values documenting 6 months of electric power consumption in 2012 in Kharkiv region (Ukraine). A number of experiments were conducted to compare performance and prediction results. In our experimental part, we used two learning criteria (the criterion (2) and the robust criterion (5)), a different number of membership functions as well as such activation functions as the hyperbolic tangent and the sigmoid-weighted linear unit (SiLU). The data set was split to training and test data arrays. In our experimental research, for the purpose of simplicity, a quite unsophisticated model embodying only a single NFU was ample. Plots of the data array in Figs.5-6 illustrate apparently footprints of outliers stipulated by peak loads, measuring faults, and other factors. The outliers' fortuitous character is almost unpredictable and results in high prediction errors. It can be seen from Figs.5 and 6 that a prediction quality is growing (RMSE and SMAPE are gradually falling down). It should be noted that in case the outliers' assessed values are used straightforward to manage the learning flow, all in all, that could lead to the model parameters' distortion and consequently to a low rate of a prediction quality. Experimental results demonstrate a dependency between a forecasting accuracy and a number of membership functions (Figs.5-6). There's also a dependence between a forecasting error and an amount of membership functions illustrated in Fig.7.

This scientific work was financially subvented by self-determined research funds of CCNU from the colleges' basic research and operation of MOE (CCNU16A02015).

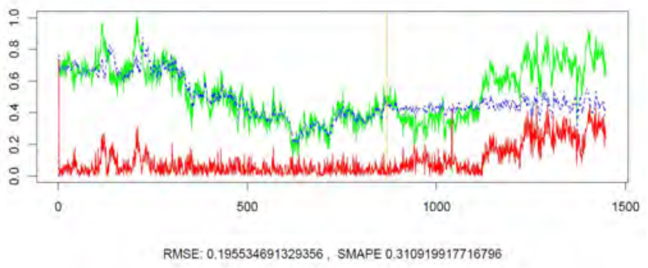

Fig. 5. A forecast performed by the hybrid neuro-fuzzy element (2 membership functions; the tanh activation function)

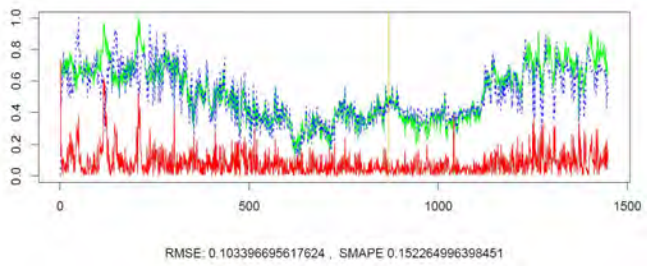

Fig. 6. A forecast performed by the hybrid neuro-fuzzy element (5 membership functions; the SiLU activation function)

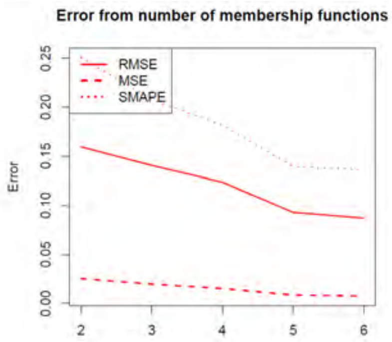

Fig. 7. A forecasting error for the hybrid neuro-fuzzy element depending on a number of membership functions

V. CONCLUSION

The described modification for the hybrid neuro-fuzzy element has been developed as a structural node for more complex computational systems like evolving cascade neuro-fuzzy systems and deep learning systems. Specifications of the membership functions can be set up in a rather straightforward manner to restrict large input values and contract extreme values. A few activation functions were considered for the offered modification in the neuro-fuzzy node in dependence to the nature of a task at hand. It was also recommended to give rise to approximation qualities of the system by applying the cubic splines as the membership functions. Having said that, it should be highlighted that the developed element is quite simple from the actualization point of view and keeps in possession approximating properties and a high processing speed.

ACKNOWLEDGMENT

This scientific work was financially assisted by self-determined research funds of CCNU from the colleges' basic research and operation of MOE (CCNU16A02015).

Oleksii K. Tyshchenko is kindly grateful for the financial assistance of the Visegrad Scholarship Program—EaP #51700967 funded by the International Visegrad Fund (IVF).

REFERENCES

[1] J. Gama, Knowledge Discovery from Data Streams. Boca Raton: Chapman and Hall/CRC, 2010.

[2] A. Bifet, R. Gavaldà, G. Holmes, and B. Pfahringer, Machine Learning for Data Streams with Practical Examples in MOA. The MIT Press, 2018.

[3] A. Bifet, Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams. Amsterdam: IOS Press, 2010.

[4] Ye.V. Bodyanskiy, O.K. Tyshchenko, and D.S. Kopaliani, "An Evolving Connectionist System for Data Stream Fuzzy Clustering and Its Online Learning", Neurocomputing, vol. 262, pp.41-56, 2017.

[5] M. Garofalakis, J. Gehrke, and R. Rastogi (eds.), Data Stream Management. Processing High-Speed Data Streams. Berlin Heidelberg: Springer-Verlag, 2016.

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

[7] B.M. Wilamowski and J.D. Irwin, Intelligent Systems. Boca Raton: CRC Press, 2017.

[8] J. Kacprzyk and W. Pedrycz (eds.), Springer Handbook of Computational Intelligence. Berlin Heidelberg: Springer-Verlag, 2015.

[9] C.L. Mumford and L.C. Jain, Computational Intelligence. Berlin: Springer-Verlag, 2009.

[10] Ye. Bodyanskiy, I. Pliss, D. Peleshko, Yu. Rashkevych, and O. Vynokurova, "Hybrid Generalized Additive Wavelet-Neuro-Fuzzy-System and its Adaptive Learning", Dependability Engineering and Complex Systems: Proc. of the Eleventh International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, Brunow, Poland, pp. 51-61, 2016.

[11] D. Graupe, Principles of Artificial Neural Networks (Advanced Series in Circuits and Systems). Singapore: World Scientific Publishing Co. Pte. Ltd., 2007.

[12] K.-L. Du and M.N.S. Swamy, Neural Networks and Statistical Learning. London: Springer, 2014.

[13] K. Suzuki, Artificial Neural Networks: Architectures and Applications. NY: InTech, 2013.

[14] R. Tkachenko and I. Izonin, "Model and Principles for the Implementation of Neural-Like Structures based on Geometric Data Transformations". Advances in Computer Science for Engineering and Education. ICCSEEA2018. Advances in Intelligent Systems and Computing, 2018, in press.

[15] G. Hanrahan, Artificial Neural Networks in Biological and Environmental Analysis. NW: CRC Press, 2011.

[16] Ye. Bodyanskiy, O. Tyshchenko, and A. Deineko, "An Evolving Radial Basis Neural Network with Adaptive Learning of Its Parameters and Architecture", Automatic Control and Computer Sciences, Vol. 49, No. 5, pp. 255-260, 2015.

[17] S. Haykin, Neural Networks and Learning Machines (3rd Edition). NJ: Prentice Hall, 2009.

[18] R. Tkachenko, P. Tkachenko, I. Izonin, and Y. Tsymbal, "Learning-based image scaling using neural-like structure of geometric transformation paradigm", Studies in Computational Intelligence, vol. 730, pp. 537–565, 2018.

[19] S. Bassis, A. Esposito, F. C. Morabito, Advances in Neural Networks: Computational and Theoretical Issues. Springer International Publishing, 2016.

[20] I. Izonin, R. Tkachenko, D. Peleshko, T. Rak and D. Batyuk, "Learning-based image super-resolution using weight coefficients of synaptic connections", Proc. Xth International Scientific and Technical Conference "Computer Sciences and Information Technologies" (CSIT), Lviv, Ukraine, pp. 25-29, 2015.

[21] J-S.R. Jang, C.T. Sun and E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, New Jersey: Prentice Hall, 1997.

[22] L.X. Wang, Adaptive Fuzzy Systems and Control. Design and Stability Analysis. Upper Saddle River, New Jersey: Prentice Hall, 1994.

[23] E. Uchino and T. Yamakawa, "Soft computing based signal prediction, restoration and filtering", Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks and Genetic Algorithms, Boston: Kluwer Academic Publisher, pp. 331-349, 1997.

[24] Ye. Bodyanskiy, O. Tyshchenko, and D. Kopaliani, "An Extended Neo-Fuzzy Neuron and its Adaptive Learning Algorithm", I.J. Intelligent Systems and Applications (IJISA), Vol.7(2), pp.21-26, 2015.

[25] T. Miki and T. Yamakawa, "Analog implementation of neo-fuzzy neuron and its on-board learning", Computational Intelligence and Applications, Piraeus: WSES Press, pp. 144-149, 1999.

[26] Ye. Bodyanskiy, O. Tyshchenko, and D. Kopaliani, "A hybrid cascade neural network with an optimized pool in each cascade", Soft Computing. A Fusion of Foundations, Methodologies and Applications (Soft Comput), Vol.19, No.12, pp.3445-3454, 2015.

[27] Zh. Hu, Ye.V. Bodyanskiy, and O.K. Tyshchenko, "A Deep Cascade Neural Network Based on Extended Neo-Fuzzy Neurons and its Adaptive Learning Algorithm", Proc. of 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kyiv, Ukraine, pp.801-805, 2017.

[28] Zh. Hu, Ye.V. Bodyanskiy, O.K. Tyshchenko, and O.O. Boiko, "An Evolving Cascade System Based on a Set of Neo-Fuzzy Nodes", International Journal of Intelligent Systems and Applications (IJISA), Vol. 8(9), pp.1-7, 2016.

[29] Zh. Hu, Ye.V. Bodyanskiy, and O.K. Tyshchenko, "A Hybrid Growing ENFN-Based Neuro-Fuzzy System and its Rapid Deep Learning", Proc. of 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT'2017), Lviv, Ukraine, pp.514-519, 2017.

[30] Ye. Bodyanskiy, O. Tyshchenko, and D. Kopaliani, "Adaptive learning of an evolving cascade neo-fuzzy system in data stream mining tasks", Evolving Systems, Vol.7, No.2, pp.107-116, 2016.

[31] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[32] Ye. Bodyanskiy, O. Vynokurova, P. Mulesa, G. Setlak, and I. Pliss, "Fast Learning Algorithm for Deep Evolving GMDH-SVM Neural Network in Data Stream Mining Tasks", Proc. of 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), pp. 257-262, 2016.

[33] A. L. Caterini and D. E. Chang, Deep Neural Networks in a Mathematical Framework. Springer, 2018.

[34] J. Heaton, Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks. CreateSpace Independent Publishing Platform, 2015.

[35] A. Menshawy, Deep Learning By Example: A hands-on guide to implementing advanced machine learning algorithms and neural networks. Packt Publishing Limited, 2018.

[36] M. Fullan, J. Quinn, and J. McEachen, Deep Learning: Engage the World Change the World. Corwin, 2017.

[37] D. Graupe, Deep Learning Neural Networks: Design and Case Studies. World Scientific, 2016.

[38] Y. LeCun, Y. Bengio, G.E. Hinton, "Deep learning", Nature, vol. 521, pp. 436–444, 2015.

[39] J. Schmidhuber, "Deep learning in neural networks: an overview," Neural Networks, no. 61, pp. 85–117, 2015.

[40] Y. Bodyanskiy and S. Popov, "Neuro-fuzzy unit for real-time signal processing," Proc. IEEE East-West Design & Test Workshop (EWDTW'06), Sochi, Russia, pp. 403-406, 2006.

[41] Ye. Bodyanskiy, S. Popov, and M. Titov, "Robust learning algorithm for networks of neuro-fuzzy units", Innovations and Advances in Computer Sciences and Engineering, pp. 343-346, 2010.

[42] W. J. J. Rey, Robust Statistical Methods. Berlin-Heidelberg-New York: Springer, 1978.

[43] D. S. Chen and R. C. Jain, "A Robust Back Propagation Learning Algorithm for Function Approximation", IEEE Trans. Neural Networks, vol. 5, pp. 467-479, 1994.

[44] A. Cichocki and R. Unbehauen, Neural Networks for Optimization and Signal Processing. Stuttgart: Teubner, 1993.