

The Neo-Fuzzy Autoencoder for Adaptive Deep Neural Systems and its Learning

Yevgeniy Bodyanskiy, Iryna Pliss, Olena Vynokurova

Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, UKRAINE, Kharkiv, 14 Nauky Ave.,
email: yevgeniy.bodyanskiy@nure.ua, iryna.pliss@nure.ua, vynokurova@gmail.com

Abstract: In this paper the autoencoder based on the generalized neo-fuzzy neurons is proposed. Also its fast learning algorithm was proposed. Such system can be used as part of deep learning neural networks. The proposed neo-fuzzy autoencoder is characterized by high learning speed and less number of tuned parameters in comparison with well-known approaches. The efficiency of proposed approach has been examined based on different benchmarks and real data sets.

Keywords: neo-fuzzy autoencoder, deep learning network, neo-fuzzy neuron, fast learning algorithm, data compression.

I. INTRODUCTION

The task of compressing information that should be further processed, is one of main problem, which is solved in Data Mining. For solving this problem, a lot of approaches [1-5] are proposed, at that more of these methods comprehend the information processing in batch mode, when the fixed-sized data set is processed many times.

It is very important in order to the process of compression, the loss of information is minimal. Nowadays the approaches based on Deep neural networks (DNNs) [6-9] are widely used for solving the many tasks, which is connected with analysis of Big Data. As it can be seen from many researches the DNNs provide significantly better results than the conventional shallow neural networks.

The inherent part of DNN is, so-called autoencoder, which implements the compression of the input data and forms the input layers of the neural network.

As such autoencoders the multilayer associative “bottle-neck” perceptrons or restricted Boltzmann machines, in which nodes are the elementary Rosenblatt perceptrons with the sigmoidal activation functions, are often used.

Unfortunately, the learning process of such autoencoders demands a large time spending and cannot be implemented in online mode.

In the connection with the intensive development of Data Mining, Data Stream Mining, Web Mining over recent years the development of high speed information compression systems is an important problem. Such systems have to process data in sequential mode (perhaps in online mode) as the real information processing systems need.

II. THE ARCHITECTURE OF NEO-FUZZY AUTOENCODER

Fig. 1 shows the architecture of the proposed autoencoder, which is autoassociative “bottle-neck” modification of

Kolmogorov’s neuro-fuzzy network [10-14] that implements the multiresolution approach and is the universal approximator according to the theorem of Kolmogorov-Arnold and Yam-Nguen-Kreinovich.

It should be notice in [15, 16] the architecture, which nodes are the neo-fuzzy neurons (NFNs) [17], is considered. In spite of the simplicity of learning algorithm for synaptic weights, such system is abundant in the sense of the membership functions number.

Using the generalized neo-fuzzy neurons [18] instead of conventional NFN allows significantly to reduce the membership functions number and to introduce stacked NN [19]. Such stacked NN allows to simplify the architecture of autoencoder and in this way to speed up the learning process.

Therefore, autoencoder consists of two sequentially connected layers, which are implemented with the generalized neo-fuzzy neurons $\text{GNFN}^{[1]}$ and $\text{GNFN}^{[2]}$.

The sequence of input signals, which have to compress $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T \in R^n$, ($k = 1, 2, \dots$ is a number of the observation or a current instant of time), is fed to $\text{GNFN}^{[1]}$.

$\text{GNFN}^{[1]}$ consists of the n multidimensional nonlinear synapses $MNS_i^{[1]}$, $i = 1, 2, \dots, n$, where each of them has a one input, m outputs, h membership functions $\mu_{li}^{[1]}(x_i(k))$, $l = 1, 2, \dots, h$ and mh tuned synaptic weights $w_{ji}^{[1]}$, $j = 1, 2, \dots, m$.

The output of $\text{GNFN}^{[1]}$ is the compressed vector of the signals $y(k) = (y_1(k), \dots, y_j(k), \dots, y_m(k))^T \in R^m$, $m < n$, which simultaneously is output of the autoencoder. The signal $y(k)$ is fed to the inputs of $\text{GNFN}^{[2]}$, which contains m inputs, m multidimensional nonlinear synapses $MNS_j^{[2]}$, where each of them has one input, n outputs, h membership functions $\mu_{lj}^{[2]}(y_j(k))$, $l = 1, 2, \dots, h$ and nh synaptic weights $w_{ij}^{[2]}$.

Thus, considered autoencoder contains $2nmh$ tuned synaptic weights and $(n+m)h$ membership functions that is significantly fewer than in the architecture in [20].

In the outputs of $\text{GNFN}^{[2]}$ the recovered signal $\hat{x}(k) = (\hat{x}_1(k), \dots, \hat{x}_i(k), \dots, \hat{x}_n(k))^T$ is formed. In such manner the autoencoder is the autoassociative hybrid neo-fuzzy system of computational intelligence.

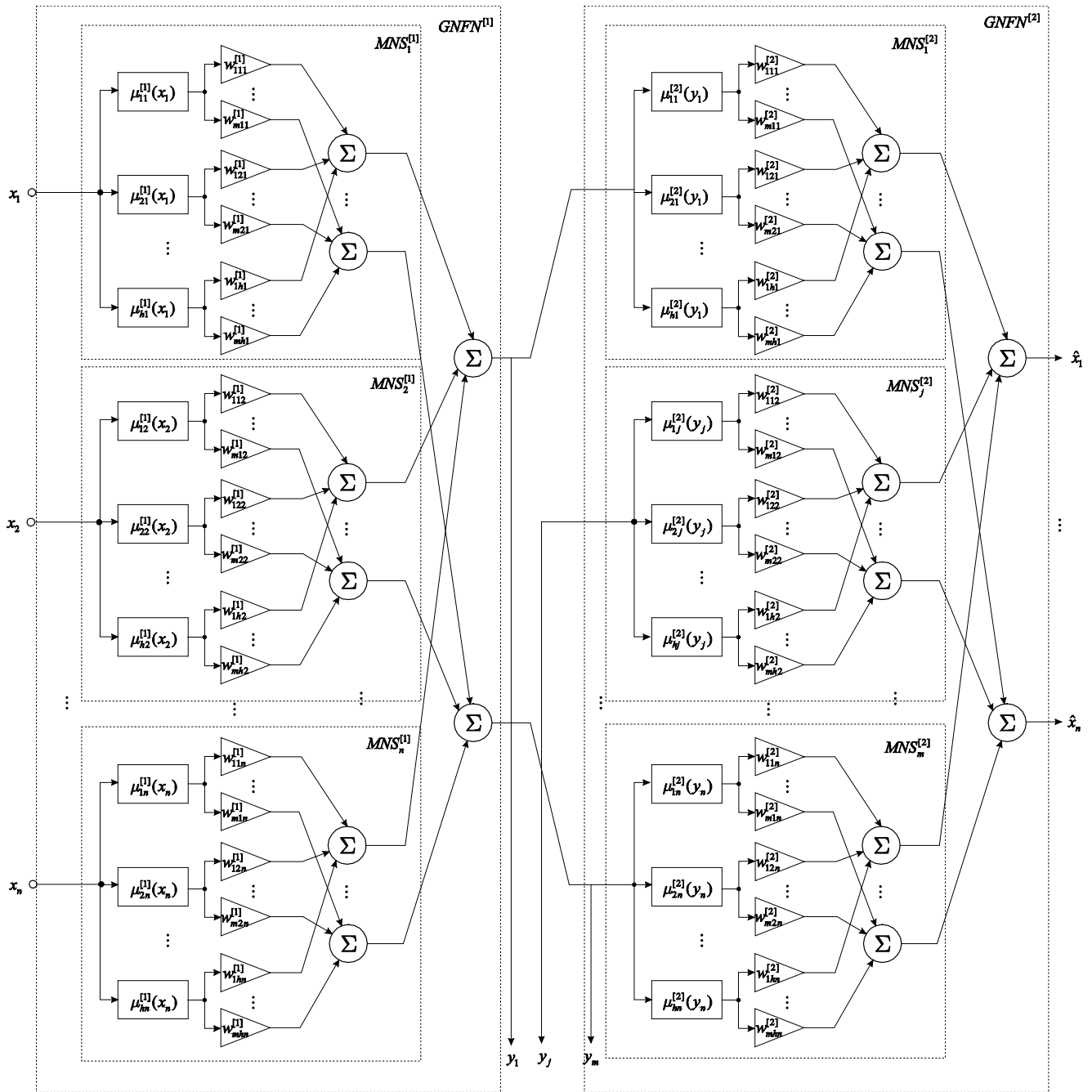


Fig 1. The architecture of the proposed neo-fuzzy autoencoder.

The proposed system is implemented a nonlinear mapping in the form

$$\hat{x}_i(k) = \sum_{j=1}^m \sum_{l=1}^h w_{ij}^{[2]} \mu_{ij}^{[2]} \left(\sum_{i=1}^n \sum_{l=1}^h w_{ji}^{[1]} \mu_{li}^{[1]} (x_i(k)) \right),$$

or in the matrix form

$$\hat{x}(k) = W^{[2]} \mu^{[2]} (W^{[1]} \mu^{[1]} (x(k))),$$

where

$$W^{[1]} = \begin{pmatrix} w_{111}^{[1]} & w_{121}^{[1]} & \cdots & w_{1hn}^{[1]} \\ w_{211}^{[1]} & w_{221}^{[1]} & \cdots & w_{2hn}^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m11}^{[1]} & w_{m21}^{[1]} & \cdots & w_{mhn}^{[1]} \end{pmatrix},$$

$$W^{[2]} = \begin{pmatrix} w_{111}^{[2]} & w_{121}^{[2]} & \cdots & w_{1hm}^{[2]} \\ w_{211}^{[2]} & w_{221}^{[2]} & \cdots & w_{2hm}^{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n11}^{[2]} & w_{n21}^{[2]} & \cdots & w_{nhm}^{[2]} \end{pmatrix},$$

$$\begin{aligned} \mu^{[1]}(x(k)) &= (\mu_{11}^{[1]}(x_1(k)), \mu_{21}^{[1]}(x_1(k)), \dots, \mu_{h1}^{[1]}(x_1(k)), \\ &\mu_{12}^{[1]}(x_2(k)), \dots, \mu_{i1}^{[1]}(x_i(k)), \dots, \mu_{hm}^{[1]}(x_n(k)))^T, \\ \mu^{[2]}(y(k)) &= (\mu_{11}^{[2]}(y_1(k)), \mu_{21}^{[2]}(y_1(k)), \dots, \mu_{h1}^{[2]}(y_1(k)), \\ &\mu_{12}^{[2]}(y_2(k)), \dots, \mu_{ij}^{[2]}(y_j(k)), \dots, \mu_{hm}^{[2]}(y_m(k)))^T. \end{aligned}$$

III. THE LEARNING ALGORITHM FOR SYNAPTIC WEIGHTS OF NEO-FUZZY AUTOENCODER

For the tuning the synaptic weights of GNFN^[2] we can use the gradient procedure of minimizing the quadratic criterion in the form

$$\begin{aligned} w_{ij}^{[2]}(k) &= w_{ij}^{[2]}(k-1) - \eta^{[2]}(k) \frac{\partial e_i^2(k)}{\partial w_{ij}^{[2]}} = \\ &= w_{ij}^{[2]}(k-1) + \eta^{[2]}(k) e_i(k) \mu_{ij}^{[2]}(y_j(k)) \end{aligned}$$

where $\eta^{[2]}(k)$ is learning rate parameter of the output layer, which is chosen accordingly to the condition in [20, 21]

$$\eta^{[2]}(k) = (r^{[2]}(k))^{-1}; r^{[2]}(k) = \alpha r^{[2]}(k-1) + \|\mu^{[2]}(y(k))\|^2$$

where $0 \leq \alpha \leq 1$ is forgetting factor.

For tuning the synaptic weights GNFN^[1] the optimized backpropagation error procedure, which for uniformly distributed in the line of X-axis the triangular membership functions with centers $\bar{x}_i^{[1]}$ $\bar{y}_{ij}^{[2]}$ can be write in the form

$$w_{ji}^{[1]}(k) = w_{ji}^{[1]}(k-1) + \eta^{[1]}(k) e_i(k) \mu_{ji}^{[1]}(x_i(k)) \tilde{w}_{ij}^{[2]}(k)$$

where

$$\begin{aligned} \eta^{[1]}(k) &= (r^{[1]}(k))^{-1}; r^{[1]}(k) = \alpha r^{[1]}(k-1) + \|\mu^{[1]}(x(k))\|^2, \\ \tilde{w}_{ij}^{[2]}(k) &= \sum_{l=1}^h w_{ij}^{[2]}(k) \begin{cases} (\bar{y}_{l,j}^{[2]} - \bar{y}_{l-1,j}^{[2]})^{-1}, & \text{if } y_j(k) \in [\bar{y}_{l-1,j}^{[2]}, \bar{y}_{l,j}^{[2]}], \\ (\bar{y}_{l,j}^{[2]} - \bar{y}_{l+1,j}^{[2]})^{-1}, & \text{if } y_j(k) \in [\bar{y}_{l,j}^{[2]}, \bar{y}_{l+1,j}^{[2]}], \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

The proposed learning algorithm for synaptic weights of autoencoder is characterized by high speed and adjugate following and filtering properties.

IV. EXPERIMENTS

For effectiveness verification of the proposed neo-fuzzy autoencoder, the data sets were taken from UCI Repository [22]: Iris, Wine, Hayes-roth. Data set ‘‘Iris’’ contains 150 observations (Number of Attributes: 4) of 3 classes, Data set ‘‘Wine’’ contains 178 observations (Number of Attributes: 13) of 3 classes, data set ‘‘Hayes-roth’’ contains 160 observations (Number of Attributes: 5) of 3 classes.

It is seen from Fig.2 data, which are compressed using neo-fuzzy autoencoder, are more compact clusters than data, which

are compressed based on the autoassociative multilayer neural network ‘‘Bottle Neck’’.

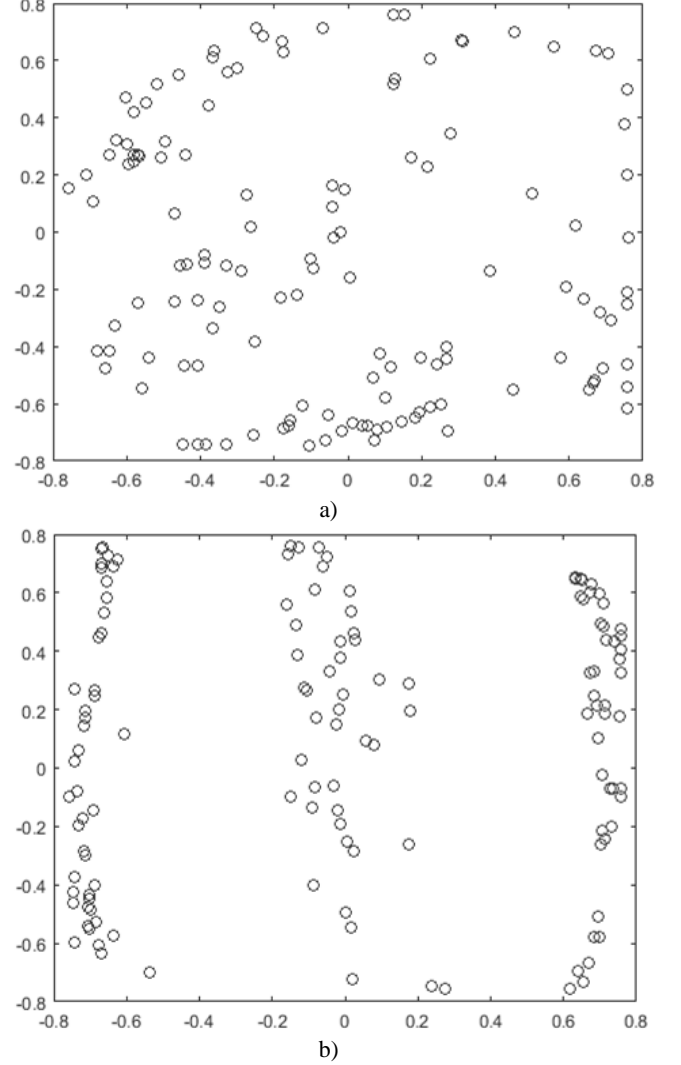


Fig. 2. Data set Hayes-roth after compression based on the autoassociative multilayer neural network ‘‘Bottle Neck’’ (a) and neo-fuzzy autoencoder (b)

The results, which were obtained using the proposed neo-fuzzy autoencoder, were compared with the results of autoassociative multilayer neural network ‘‘Bottle Neck’’ (Table I). The dimension of compression data was 2 components. The simulation was performed 20 times with different initial condition and the results were averaged.

TABLE I. RESULTS OF SIMULATION

AUTOENCODERS	DATA SETS	MSE
Neo-fuzzy autoencoder	Iris	0.199
	Wine	0.499
	Hayes-roth	0.312
Autoassociative three layer neural network ‘‘Bottle Neck’’	Iris	0.486
	Wine	0.903
	Hayes-roth	0.593

IV. CONCLUSIONS

The architecture of «bottle-neck» two-layer autoencoder and its learning algorithm are proposed. Such system is based on generalized neo-fuzzy neurons and is autoassociative «bottle-neck» modification of Kolmogorov's neuro-fuzzy network. The proposed hybrid neo-fuzzy system of computational intelligence provides high quality of information compression, which are fed sequentially for processing. It is characterized by computational simplicity and high speed of the learning process.

REFERENCES

- [1] J. Han and M. Kamber, "Data Mining: Concepts and Techniques". Amsterdam: Morgan Kaufman Publ., 2006.
- [2] C.C. Aggarwal, "Data Mining", N.Y.: Springer, 2015.
- [3] A. Bifet, R. Gavaldà, G. Holmes, and B. Pfahringer, Machine Learning for Data Streams with Practical Examples in MOA. The MIT Press, 2018.
- [4] A. Bifet, Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams. Amsterdam: IOS Press, 2010.
- [5] A. Menshawy, Deep Learning By Example: A hands-on guide to implementing advanced machine learning algorithms and neural networks. Packt Publishing Limited, 2018.
- [6] M. Fullan, J. Quinn, and J. McEachen, Deep Learning: Engage the World Change the World. Corwin, 2017.
- [7] A. L. Caterini and D. E. Chang, Deep Neural Networks in a Mathematical Framework. Springer, 2018.
- [8] Y. LeCun, Y. Bengio, and G.E. Hinton, "Deep Learning". *Nature*, 2015, v. 521, pp. 436-444.
- [9] D. Graupe, "Deep Learning Neural Networks: Design and Case Studies". World Scientific Publishing Company, 2016.
- [10] V. Kolodyazhniy and Ye. Bodyanskiy, "Fuzzy Kolmogorov's Network," in *Lecture Notes in Computer Science*, vol. 3214, M.G. Negoita et al., Eds., Springer-Verlag, 2004, pp.764-771.
- [11] Ye. Bodyanskiy, V. Kolodyazhniy and P. Otto, "Neuro-fuzzy Kolmogorov's network for time-series prediction and pattern classification," in *Lecture Notes in Artificial Intelligence*, vol. 3698, U. Furbach, Ed., Heidelberg: Springer-Verlag, 2005, pp. 191-202.
- [12] V. Kolodyazhniy, Ye. Bodyanskiy and P. Otto, "Universal approximator employing neo-fuzzy neurons," in *Computational Intelligence Theory and Applications*, Ed. B. Reusch, Ed., Berlin-Heidelberg: Springer, 2005, pp. 631-640.
- [13] V. Kolodyazhniy, Ye. Bodyanskiy, V. Poyedyntseva, and A. Stephan "Neuro-fuzzy Kolmogorov's network with a modified perceptron learning rule for classification problems," in *Advances in Soft Computing*, vol. 38, B. Reuch, Ed., Berlin-Heidelberg: Springer-Verlag, 2006, pp. 41-49.
- [14] Ye. Bodyanskiy, Ye. Gorshkov, V. Kolodyazhniy, and V. Poyedyntseva "Neuro-fuzzy Kolmogorov's network," in *Lecture Notes in Computer Science*, vol.3697, W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny, Eds., Berlin-Heidelberg: Springer-Verlag, 2005, pp.1-6.
- [15] V. Kolodyazhniy, F. Klawonn, and K. Tschumitschew, "A neuro-fuzzy model for dimensionality reduction and its application" *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* vol. 15, is. 05, October 2007, pp. 571-593.
- [16] Vynokurova O., Bodyanskiy Ye., Pliss I., Peleshko D., Rashkevych Yu. "Neo-fuzzy encoder and its adaptive learning for Big Data processing." *Scientific Journal of RTU, Series "Computer Science"* Volume "Information Technology and Management Science" 2017, vol. 20, pp. 6-11.
- [17] T. Yamakawa, E. Uchino, T. Miki and H. Kusanagi, "A neo-fuzzy neuron and its applications to system identification and prediction of the system behavior," in *Proc. of 2-nd Int. Conf. on Fuzzy Logic and Neural Networks "IIZUKA-92"*, Iizuka, Japan, pp. 477-483, 1992.
- [18] R.P.Landim, B. Rodrigues, S.R. Silva, and W.M. Caminhas, "A neo-fuzzy-neuron with real time training applied to flux observer for an induction motor". In: *Proceedings of IEEE Vth Brazilian Symposium on Neural Networks, Belo Horizonte*, 9-11 Dec 1998, pp. 67-72.
- [19] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85-117, Jan. 2015. (doi: 10.1016/j.neunet.2014.09.003)
- [20] Ye. Bodyanskiy, I. Kokshenev, V. Kolodyazhniy, "An adaptive learning algorithm for a neo fuzzy neuron," in *Proc. 3rd Int. Conf. of European Union Society for Fuzzy Logic and Technology (EUSFLAT 2003)*, Zittau, 2003, pp. 375-379.
- [21] P. Otto, Ye. Bodyanskiy, V. Kolodyazhniy, "A new learning algorithm for a forecasting neuro-fuzzy network," *Integrated Computer-Aided Engineering*, vol. 10, pp. 399-409, Dec. 2003
- [22] UCI Repository of machine learning databases. CA: University of California, Department of Information and Computer Science. [Online]. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>