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

The Autoencoder Based on Generalized Neo-Fuzzy Neuron and its Fast Learning for Deep Neural Networks

Yevgeniy Bodyanskiy
Control System Research Laboratory
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
yevgeniy.bodyanskiy@nure.ua

Yuriy Rashkevych
Ministry of Education and Science of Ukraine
Kyiv, Ukraine
rashkevyuriy@gmail.com

Dmytro Peleshko
IT Step University
Lviv, Ukraine
dpeleshko@gmail.com

Olena Vynokurova
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
IT Step University
Lviv, Ukraine
vynokurova@gmail.com

***Abstract*— In this paper the autoencoder based on the generalized neo-fuzzy neurons is proposed. Also its fast learning algorithm based on quadratic criterion was proposed. Such system can be used as part of deep learning systems. The proposed autoencoder is characterized by high learning speed and less number of tuned parameters in comparison with well-known autoencoders of "bottle neck" type. The efficiency of proposed approach has been justified based on different benchmarks and real data sets.**

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

I. Introduction

Nowadays the deep neural networks (DNN) [1-4] are becoming more widespread for solving the many type Data Mining tasks, first of all, due to the significantly higher quality of information processing in comparison with conventional shallow neural networks (SNN). But such higher quality is achieved at the cost of very slow learning speed. This fact doesn't allow to use existed DNN in the Data Stream Mining tasks when the information is fed sequentially in online mode. In the connection with that, the reducing the learning time is a very actual problem.

The important part of any DNN is the subsystem of input information compression, which is named an autoencoder that provides the dimensionality reduction of the input vectors-patterns without significant loss of an information. Such reduction process allows avoiding the undesirable effects, which are provided by «curse of dimensionality». One of most well-known autoencoders is the autoassociative multilayer perceptron "bottle-neck", which provides the optimal information compression, but also needs more learning time.

To overcome the difficulty, which is connected with low learning speed, we can use the hybrid systems of computational intelligence instead of the classical neural networks that are constructed based on the elementary perceptron of F. Rosenblatt. The neo-fuzzy neuron (NFN), which was proposed by T. Yamakawa and co-authors in [5-7] can be used as the structural block of such systems. The NFN is defined by high approximation properties and simplicity of learning process. It should be noticed, the learning process of the NFN can be optimized by speed [8] because the output signals of the NFNs are linearly dependent on the tuned synaptic weights.

The neuro-fuzzy Kolmogorov's network using the NFNs was introduced by authors in [9-13]. Such network has the two layers for information processing and is characterized by the universal approximation properties according to A. Kolmogorov - V. Arnold theorem. Using these networks, the authors in [14] have proposed the neuro-fuzzy model for the dimensionality reduction, which is learned using error backpropagation algorithm with gradient methods. In [15-17] the optimized learning algorithms for two-layer autoencoders based on the neo-fuzzy neurons were proposed, which allow significantly reducing the learning time.

In the same time the hybrid systems of computational intelligence with many inputs and many outputs, which are constructed based on neo-fuzzy neurons, have an abundant number of membership functions. It is possible significantly to reduce the number of these functions, using the so-called generalized neo-fuzzy neuron (GNFN) [18]. GNFN is the extension of NFN for a multidimension case and contains less number of membership functions.

Therefore, in the paper, the architecture of two-layer autoencoder based on GNFN and its optimized learning algorithms are proposed. Such approach allows reducing the time of information preprocessing in DNN.

II. The Architecture of Autoencoder Based on GNFN

The proposed autoencoder has the architecture, which is shown on Fig. 1. Such architecture consists of two sequentially connected layers, which are presented by the generalized neo-fuzzy neurons GNFN[1] and GNFN[2]. The input signals $x(k)=\left(x_1(k),\ldots,x_i(k),\ldots,x_n(k)\right)^T\in R^n$, $k=1,2,\ldots,N,\ldots$ are fed to the GNFN[1], which consists of n multidimensional nonlinear synapses $MNS_i^{[1]}$, $i=1,2,\ldots,n$. Each of them has the one input, m outputs, h membership functions $\mu_{li}^{[1]}(x_i(k)),\ l=1,2,\ldots,h$ and mh

tuned synaptic weights $w_{jli}^{[1]},\ j=1,2,\ldots,m$. The architecture of GNFN[1] is shown on Fig. 2.

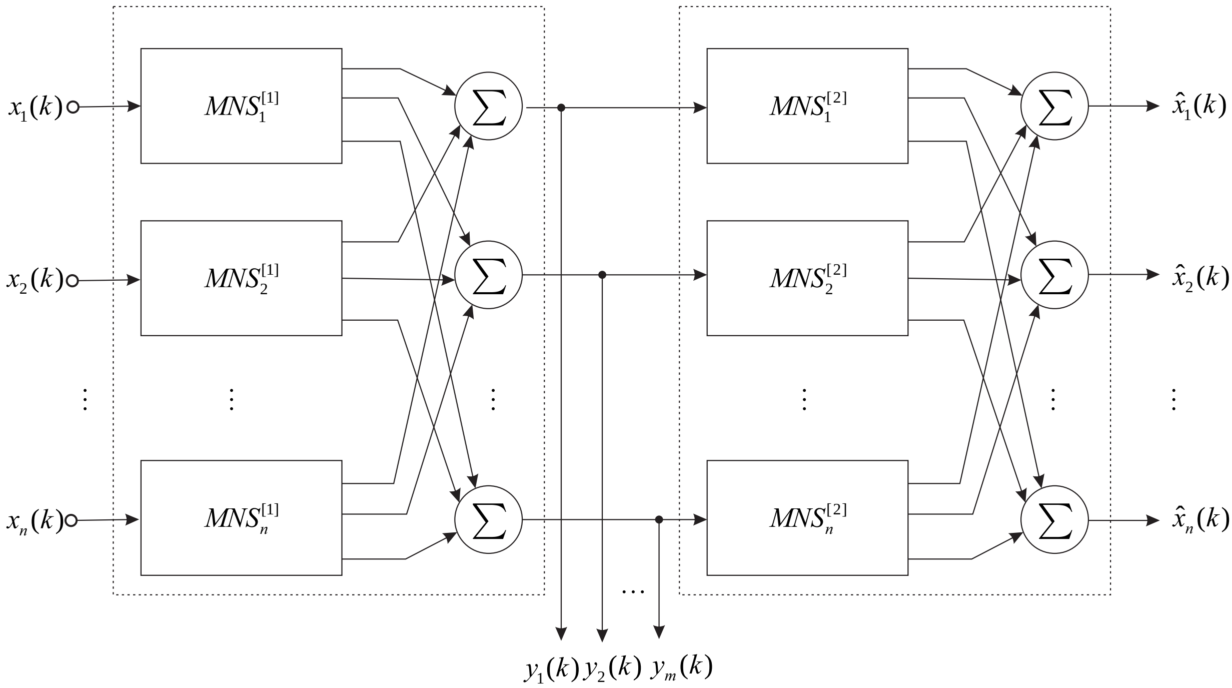


Fig. 1. Autoencoder based on GNFNs

The output of the first autoencoder layer is the compressed vector of signals $y(k)=\left(y_1(k),\ldots,y_j(k),\ldots,y_m(k)\right)^T\in R^m$, $m<n$, which at the same time is the output of system in whole. This signal is fed to the inputs of GNFN[2], which differs from GNFN[1] only in that it has m inputs, m multidimensional nonlinear synapses $MNS_j^{[2]}$, each of them has one input, n outputs, h membership functions $\mu_{lj}^{[2]}(y_j(k)),\ l=1,2,\ldots,h$ and nh tuned synaptic weights $w_{ilj}^{[2]}$. In total, the autoencoder contains $2nmh$ tuned synaptic weights and $(n+m)h$ membership functions that is significantly less than in the systems, which were described in [14-17].

The output of GNFN[2] is the recovered vector of input signals $\hat{x}(k)=\left(\hat{x}_1(k),\ldots,\hat{x}_i(k),\ldots,\hat{x}_n(k)\right)^T$, at that the less mismatch between $x(k)$ and $\hat{x}(k)$, the higher the quality of the information compression by the autoencoder.

Therefore the autoencoder under consideration is the autoassociative hybrid neo-fuzzy system like «bottle-neck» system.

In general, the proposed system implements the mapping of “input-output” in the form

$$\hat{x}_i(k)=\sum_{j=1}^{m}\varphi_{ij}^{[2]}(y_j(k))=\sum_{j=1}^{m}\varphi_{ij}^{[2]}\left(\sum_{i=1}^{n}\varphi_{ji}^{[1]}(x_i(k))\right),$$
$$\forall i=1,2,\ldots n$$

where $\varphi_{ji}^{[1]}(\bullet)$, $\varphi_{ij}^{[2]}(\bullet)$ are the nonlinear transformations, which are implemented by the multivariate nonlinear synapses of the autoencoder layers.

These transformations can be written in the form

$$y_j(k)=\sum_{i=1}^{n}\sum_{l=1}^{h}w_{jli}^{[1]}\mu_{li}^{[1]}(x_i(k)),\ \forall j=1,2,\ldots,m,$$

$$\hat{x}_i(k)=\sum_{j=1}^{m}\sum_{l=1}^{h}w_{ilj}^{[2]}\mu_{lj}^{[2]}(y_j(k)),\ \forall i=1,2,\ldots n$$

or finally

$$\hat{x}_i(k)=\sum_{j=1}^{m}\sum_{l=1}^{h}w_{ilj}^{[2]}\mu_{lj}^{[2]}\left(\sum_{i=1}^{n}\sum_{l=1}^{h}w_{jli}^{[1]}\mu_{li}^{[1]}(x_i(k))\right). \qquad (1)$$

It should be noted the multidimensional nonlinear synapses in general case describe the Takagi-Sugeno-Kang (zero order) neuro-fuzzy system (Wang-Mendel system), i.e. has high approximation properties, and the transformation (1) describes the autoassociative variation of the neuro-fuzzy Kolmogorov's network, i.e. is the universal approximator.

In the simplest case as the membership function conventional triangular functions can be used, which have been used by the authors of the neo-fuzzy neuron [5-7] and can be written in the form

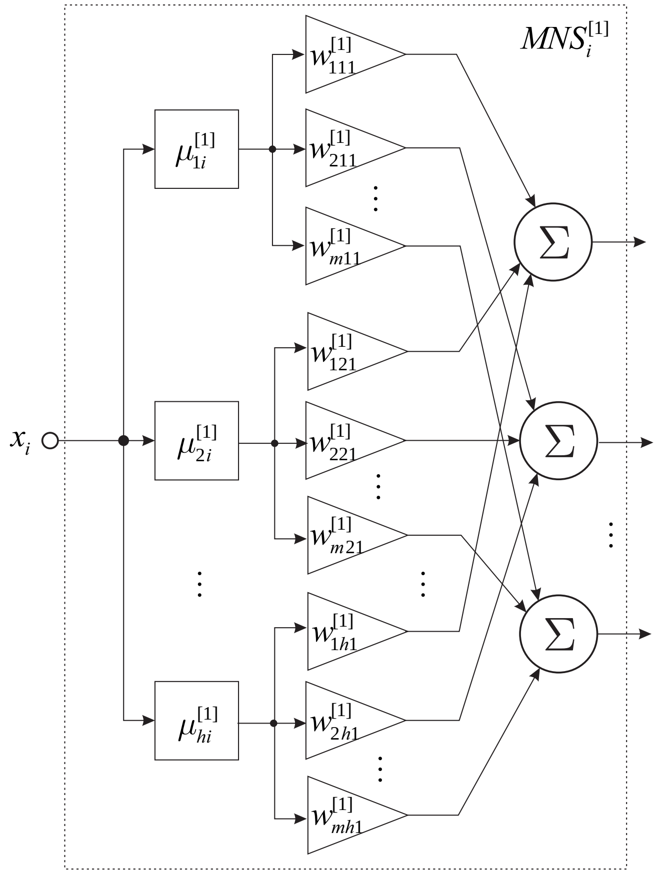


Fig. 2. Multidimensional nonlinear synapce of first layer

$$\mu_{li}^{[1]}(x_i)=\begin{cases}\dfrac{x_i-\overline{x}_{l-1,i}^{[1]}}{\overline{x}_{l,i}^{[1]}-\overline{x}_{l-1,i}^{[1]}}, \ if \ \ x_i\in[\overline{x}_{l-1,i}^{[1]},\overline{x}_{l,i}^{[1]}],\\ \dfrac{\overline{x}_{l+1,i}^{[1]}-x_i}{\overline{x}_{l+1,i}^{[1]}-\overline{x}_{l,i}^{[1]}}, \ if \ \ x_i\in[\overline{x}_{l,i}^{[1]},\overline{x}_{l+1,i}^{[1]}],\\ 0 \ \ otherwise\end{cases}$$

and

$$\mu_{lj}^{[2]}(y_j)=\begin{cases}\dfrac{y_j-y_{l-1,j}^{[2]}}{\overline{y}_{l,j}^{[2]}-\overline{y}_{l-1,j}^{[2]}}, \ if \ \ y_j\in[\overline{y}_{l-1,j}^{[2]},\overline{y}_{l,j}^{[2]}],\\ \dfrac{y_{l+1,j}^{[2]}-y_j}{\overline{y}_{l+1,j}^{[2]}-\overline{y}_{l,j}^{[2]}}, \ if \ \ y_j\in[\overline{y}_{l,j}^{[2]},\overline{y}_{l+1,j}^{[2]}],\\ 0 \ \ otherwise\end{cases} \qquad (3)$$

where $\overline{x}_{l,i}^{[1]}$, $\overline{y}_{l,j}^{[2]}$, $l=1,2,\ldots,h$ are the centers of the activation functions, in simplest case they are equidistributed along the axes x_i, y_j.

The membership functions (2), (3) fulfil Ruspini conditions:

$$\begin{cases}\mu_{l-1,i}^{[1]}(x_i)+\mu_{l,i}^{[1]}(x_i)=1, \ if \ \ x_i\in[\overline{x}_{l-1,i}^{[1]},\overline{x}_{l,i}^{[1]}],\\ \mu_{l,i}^{[1]}(x_i)+\mu_{l+1,i}^{[1]}(x_i)=1, \ if \ \ x_i\in[\overline{x}_{l,i}^{[1]},\overline{x}_{l+1,i}^{[1]}],\\ \mu_{l-1,j}^{[2]}(y_j)+\mu_{l,j}^{[2]}(y_j)=1, \ if \ \ y_j\in[\overline{y}_{l-1,j}^{[2]},\overline{y}_{l,j}^{[2]}],\\ \mu_{l,j}^{[2]}(y_j)+\mu_{l+1,j}^{[2]}(y_j)=1, \ if \ \ y_j\in[\overline{y}_{l,j}^{[2]},\overline{y}_{l+1,j}^{[2]}].\end{cases}$$

This fact significantly reduces the learning process, because in each instant of time k only two nearest-neighbor membership functions are fired and accordingly that not all synaptic weights are adjusted, but only $4nm$ ones of them.

III. THE LEARNING OF THE GNFN AUTOENCODER

As a rule, the supervised learning process of the hybrid systems of machine learning is reduced to tuning the synaptic weights set with the goal of minimizing the accepted (usually quadratic) learning criterion.

Introducing into consideration the vector of membership functions

$$\mu^{[1]}(x(k))=\left(\mu_{11}^{[1]}(x_1(k)),\mu_{21}^{[1]}(x_1(k)),\ldots,\mu_{h1}^{[1]}(x_1(k)),\mu_{12}^{[1]}(x_2(k)),\right.$$

$$\left.\ldots,\mu_{li}^{[1]}(x_i(k)),\ldots,\mu_{hn}^{[1]}(x_n(k))\right)^T, \ \mu^{[2]}(y(k))=\left(\mu_{11}^{[2]}(y_1(k)),\right.$$
$$\mu_{21}^{[2]}(y_1(k)),\ldots,\mu_{h1}^{[2]}(y_1(k)),\mu_{12}^{[2]}(y_2(k)),\ldots,\mu_{lj}^{[2]}(y_j(k)),\ldots,$$
$\left.\mu_{hm}^{[2]}(y_m(k))\right)^T$ with dimensions $(hn\times 1)$, $(hm\times 1)$ respectively and synaptic weights matrices

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

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

of dimensions $(m\times hn)$, $(n\times hm)$ respectively. Then the mapping, which realized by GNFN[1], can be written in the form

$$y(k)=W^{[1]}\mu^{[1]}(x(k))$$

and GNFN[2]

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

In general, the autoencoder realizes a mapping in the form

$$\hat{x}(k)=W^{[2]}\mu^{[2]}(W^{[1]}\mu^{[1]}(x(k))) \ , \qquad (4)$$

which is the generalization of the expression (1). During the learning process the matrix $W^{[1]}$, $W^{[2]}$ have to be obtain, which provide an optimal compression of the initial data set.

For tuning synaptic weights of GNFN[2] we can introduce into consideration the error of the recovering $i-$th element of input signal $x_i(k)$ in the form

$$\begin{aligned} e_i(k) &= x_i(k) - \hat{x}_i(k) = \\ &= x_i(k) - \sum_{l=1}^{h}\sum_{j=1}^{m} w_{ilj}^{[2]}(k-1)\mu_{lj}^{[2]}(y_j(k)) = \\ &= x_i(k) - w_i^{[2]}(k-1)\mu^{[2]}(y(k)) \end{aligned}$$

(here $w_i^{[2]}(k-1)$ is the $i-$th row of the weights matrix $W^{[2]}$) and standard learning criterion for $i-$th output

$$E_i(k) = \sum_k e_i^2(k) = \sum_k \left(x_i(k) - w_i^{[2]}(k-1)\mu^{[2]}(y(k))\right)^2 .$$

The gradient procedure for minimization of the criterion $E_i(k)$ can be written in the general form

$$\begin{aligned} w_i^{[2]}(k) &= w_i^{[2]}(k-1) - \eta^{[2]}(k)\nabla_{w_i^{[2]}} E_i(k) = \\ &= w_i^{[2]}(k-1) - \eta^{[2]}(k)\nabla_{w_i^{[2]}} e_i^2(k) = \\ &= w_i^{[2]}(k-1) + \eta^{[2]}(k)e_i(k)\mu^{[2]T}(y(k)) = \\ &= w_i^{[2]}(k-1) + \\ &\quad + \eta^{[2]}(k)(x_i(k) - w_i^{[2]}(k-1)\mu(y(k)))\mu^{[2]T}(y(k)), \end{aligned} \qquad (5)$$

where $\eta^{[2]}(k)$ is a learning rate coefficient of the output layer.

To increase learning process speed based on algorithm in the form (5) we can use either the standard recurrent least square method in the form

$$\begin{cases} w_i^{[2]}(k) = w_i^{[2]}(k-1) + \dfrac{e_i(k)\mu^{[2]T}(y(k))P^{[2]}(k-1)}{1+\mu^{[2]T}(y(k))P^{[2]}(k-1)\mu^{[2]}(y(k))}, \\ P^{[2]}(k) = P^{[2]}(k-1) - \\ \qquad - \dfrac{P^{[2]}(k-1)\mu^{[2]}(y(k))\mu^{[2]T}(y(k))P^{[2]}(k-1)}{1+\mu^{[2]T}(y(k))P^{[2]}(k-1)\mu^{[2]}(y(k-1))}, \end{cases}$$

or one-step optimal algorithm in the form [19]:

$$w_i^{[2]}(k) = w_i^{[2]}(k-1) + e_i(k)\mu^{[2]+}(y(k)) \qquad (6)$$

(here $\mu^{[2]+}(y(k)) = \mu^{[2]T}(y(k))\left\|\mu^{[2]}(y(k))\right\|^{-2}$) or the algorithm, which has both tracking and smoothing properties [20]:

$$\begin{cases} w_i^{[2]}(k) = w_i^{[2]}(k-1) + (r^2(k))^{-1}e_i(k)\mu^{[2]T}(y(k)), \\ r^{[2]}(k) = \alpha r^{[2]}(k-1) + \left\|\mu^{[2]}(y(k))\right\|^2 \end{cases} \qquad (7)$$

where $0 \le \alpha \le 1$ - forgetting factor.

It can be seen that if $\alpha = 0$, then the algorithm (7) coincides with expression (6).

The tuning of synaptic weights matrix is performed based on the error backpropagation procedure [21]. At that, we can write similarly to (5)

$$\begin{aligned} w_j^{[1]}(k) &= w_j^{[1]}(k-1) - \eta^{[1]}(k)\nabla_{w_j^{[1]}} E_i(k) = \\ &= w_j^{[1]}(k-1) - \eta^{[1]}(k)\nabla_{w_j^{[1]}} e_i^2(k) \end{aligned}$$

or element-wise

$$\begin{aligned} w_{jli}^{[1]}(k) &= w_{jli}^{[1]}(k-1) - \eta^{[1]}(k)\frac{\partial e_i^2(k)}{\partial w_{jli}^{[1]}} = \\ &= w_{jli}^{[1]}(k-1) - \eta^{[1]}(k)\frac{\partial e_i^2(k)}{\partial \hat{x}_i(k)}\frac{\partial \hat{x}_i(k)}{\partial y_j(k)}\frac{\partial y_j(k)}{\partial w_{jli}^{[1]}} = \\ &= w_{jli}^{[1]}(k-1) + \eta^{[1]}(k)e_i(k)\mu_{li}^{[1]}(x_i(k))\sum_{l=1}^{h} w_{ilj}^{[2]}(k)\frac{\partial \mu_{lj}^{[2]}(y_j(k))}{\partial y_j}. \end{aligned}$$

In the case, if the membership functions' centers in the output layer are uniformly distributed in the line of X-axis we can write

$$\frac{\partial \mu_{lj}^{[2]}(y_j(k))}{\partial y_j} = \begin{cases} (\bar{y}_{l,j}^{[2]} - \bar{y}_{l-1,j}^{[2]})^{-1}, \; if \;\; y_j(k) \in [\bar{y}_{l-1,j}^{[2]}, \bar{y}_{l,j}^{[2]}], \\ (\bar{y}_{l,j}^{[2]} - \bar{y}_{l+1,j}^{[2]})^{-1}, \; if \;\; y_j(k) \in [\bar{y}_{l,j}^{[2]}, \bar{y}_{l+1,j}^{[2]}], \\ 0 \; otherwise \end{cases}$$

or introducing the notations

$$(\bar{y}_{l,j}^{[2]} - \bar{y}_{l-1,j}^{[2]})^{-1} = \Delta\bar{y} ,$$

$$(\bar{y}_{l,j}^{[2]} - \bar{y}_{l+1,j}^{[2]})^{-1} = -\Delta\bar{y} ,$$

we can obtain the compacted expression

$$\frac{\partial \mu_{lj}^{[2]}(y_j(k))}{\partial y_j} = \begin{cases} \Delta\bar{y}, \; if \;\; y_j(k) \in [\bar{y}_{l-1,j}^{[2]}, \bar{y}_{l,j}^{[2]}], \\ -\Delta\bar{y}, \; if \;\; y_j(k) \in [\bar{y}_{l,j}^{[2]}, \bar{y}_{l+1,j}^{[2]}], \\ 0 \; otherwise. \end{cases}$$

Further, introducing the new notations

$$\sum_{l=1}^{h} w_{ilj}^{[2]}(k)\begin{pmatrix} \Delta\bar{y},\ if\ \ y_j(k)\in[\bar{y}_{l-1,j}^{[2]},\bar{y}_{l,j}^{[2]}], \\ -\Delta\bar{y},\ if\ \ y_j(k)\in[\bar{y}_{l,j}^{[2]},\bar{y}_{l+1,j}^{[2]}], \\ 0 \qquad\qquad otherwise \end{pmatrix} = \tilde{w}_{ij}^{[2]}(k),$$

we can write the procedure for adjusting the synaptic weights of the first layer

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

Choice of the learning rate parameter can be provided like (7), at that

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

In distinction from autoencoders, which are described in [14-17] the proposed system contains less number of the membership functions (it reduces its computational implementation) and has a high speed of learning algorithm due to the optimized choice of learning rate parameters.

IV. EXPERIMENTS

The effectiveness of proposed approach has been performed using data sets from UCI Repository of machine learning databases [20]. We take three data set: Iris, Parkinsons,Wine. Iris data set consists of 150 observations with 4 attributes and 3 classes, Wine data set consists of 178 observations with 13 attributes and 3 classes, Parkinsons data set consists of 197 observations with 23 attributes and 3 classes. The obtained results based on proposed autoencoder have been compared with the results based on autoassociative autoencoder "Bottle Neck" The data dimension after compression was 3 components for simplicity of visualization. The results were averaged after 20 times simulation with a different start condition for learning algorithm.

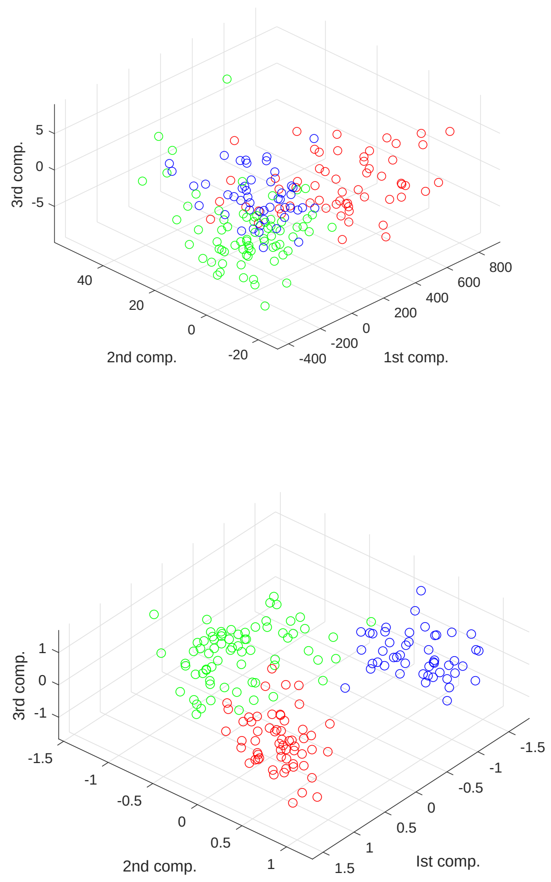
TABLE I. RESULTS OF SIMULATION BASED ON PROPOSED AUTOENCODER

Autoencoders	**Data Sets**	**Error**	**Learning time, sec.**	
			Min	***Max***
Proposed autoencoder	Iris	0.19	2.31	3.91
	Wine	0.51	2.60	4.12
	Parkinsons	0.83	6.45	7.32
Autoassociative autoencoder "Bottle Neck"	Iris	0.486	4.12	6.22
	Wine	0.903	6.44	8.26
	Parkinsons	0.593	9.21	10.98

It should be noticed, data, which are compressed based on proposed autoencoder, are more compact clusters than data, which are compressed using the autoassociative autoencoder "Bottle Neck".

V. CONCLUSIONS

In this paper, the autoencoder based on the generalized neo-fuzzy neuron and its learning algorithm are proposed. Such system can be used as part of deep learning systems or as separated autoencoder for solving compression tasks in the machine learning problems. The proposed autoencoder is co-called autoassociative "bottle-neck" system of computational intelligence but is characterized by high learning speed and less number of tuned parameters in comparison with well-known autoencoders, that allow using such system in Data Stream Mining. The efficiency of proposed approach has been justified based on different benchmark and real data set, obtained results have confirmed the advantages of the proposed autoencoder based on generalized neo-fuzzy neuron.



a)

b)

Fig. 3. Results of compression based on Wine data set using autoassociative autoencoder "Bottle Neck" (a) and proposed autoencoder (b)

REFERENCES

[1] Y. LeCun, Y. Bengio, and G.E. Hinton, "Deep Learning". Nature, 2015, v. 521, pp. 436-444.

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

[3] J. Schmidhuber, "Deep learning in neural networks: An overview." Neural Networks, 2015, v. 61, pp. 85-117.

[4] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT Press, 2016.

[5] T. Miki, and T. Yamakawa, "Analog implementation of neo-fuzzy neuron and its on-board learning." In: Mastorakis NE (eds) Computational Intelligence and Application, Piraeus: WSES Press, 1999, pp. 144–149.

[6] 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: Proceedings 2-nd International Conference on Fuzzy Logic and Neural Networks (IIZUKA-92), Iizuka, Japan, 17-22 July 1992, pp. 477–483.

[7] E. Uchino, and T. Yamakawa, “Soft computing based signal prediction, restoration and filtering.” In: Ruan D. (eds) Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks and Genetic Algorithms, Boston: Kluwer Academic Publishers, 1997, pp. 331–349.

[8] Ye. Bodyanskiy, I. Kokshenev, and V. Kolodyazhniy, “An adaptive learning algorithm for a neo-fuzzy neuron.” In: Proceedings of 3-rd International Conference of European Union Society for Fuzzy Logic and Technology (EUSFLAT), Zittau, Germany, 10-12 September 2003, pp. 375–379

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Ye. Bodyanskiy, I. Pliss, and O. Vynokurova, “On-line neuro-fuzzy big data autoencoder for deep neural networks and its rapid learning” In: Proc. of XXX Int. Conference Problems of Decision making under uncertainties (PDMU 2017) August 14-19, 2017, Vilnius, Lithuania, P. 20.

[16] Ye. Bodyanskiy, I. Pliss, D. Peleshko, Yu. Rashkevych, and O. Vynokurova, “Hybrid Generalized Additive Wavelet-Neuro-Fuzzy-System and its Adaptive Learning”. In: Eds. Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J., Dependability Engineering and Complex Systems: Proceedings of the Eleventh International Conference on Dependability and Complex Systems DepCoS-RELCOMEX. June 27-July 1, 2016, Brunow, Poland, 2016, pp. 51-61.

[17] O. Vynokurova, Ye. Bodyanskiy, I. Pliss, D. Peleshko, and Yu. Rashkevych. “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.

[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] Ye. Bodyanskiy, D. Peleshko, I. Pliss, and O. Vynokurova. Hybrid adaptive systems of computational intelligence and their on-line learning in IT energy management tasks Green IT Engineering: Concepts, Models, Complex Systems Architectures, Eds. By V. Kharchenko, Yu. P Kondratenko, J. Kacprzyk, Series: Studies in Systems, Decision and Control, Book 74, Publisher: Springer; 2017, pp 229-244.

[20] UCI Repository of machine learning databases. CA: University of California, Department of Information and Computer Science. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html