HYBRID GMDH-NEURO-FUZZY SYSTEM AND ITS TRAINING SCHEME

Yuriy Zaychenko, Yevgeniy Bodyanskiy, Oleksii Tyshchenko,
Olena Boiko, Galib Hamidov

Abstract: *A hybrid neuro-fuzzy system that combines different concepts like Deep Learning, Group Method of Data Handling and Evolving Systems is offered in this work. It's also proposed to adjust all parameters in an online way. As a node of the evolving multilayer system, there's an idea to utilize extended neo-fuzzy neurons which are exemplified by high approximating capabilities. During the learning stage, the proposed deep evolving system calculates its parameters and tweaks its architecture. The system's architecture can be evolving over time as synaptic weights, centers and widths' parameters of the neuro-fuzzy nodes are being adjusted for improvement of the system's approximation features. There's a high probability to process data sets much faster due to parallel tuning of parameters for the system. A key feature of the introduced system is that a large training set is not in demand for the system to be tuned.*

Keywords: *Evolving System; Neuro-Fuzzy System; Multilayer Neural Network; Extended Neo-Fuzzy Neuron; Group Method of Data Handling*

ITHEA keywords: *I.2 Artificial Intelligence; I.2.6 Learning, neural nets*

Introduction

An increasing number of various up-to-date applications produce non-stationary time series describing a behavior of some complex processes. That's why the generated model for predicting the system's dynamics and optimizing their presentation should be capable of adapting its parameters over time [1-6].

During the last few years, evolving intelligent systems have become widely spread and popular for handling any sort of dynamic modeling and training requirements in real-world (online) applications, especially under conditions of a growing effect of the dynamic data context, sequential video analysis, and web mining. This demand is justified by the growing dynamic and complexity of current problems as well as the ascending volumes of data storages, which lead to the fact that traditional batch training is not possible any more to be applied within some reasonable time period and tolerable accuracy. [7-15].

The evolving incremental learning systems should process huge amounts of data, analyze the data rapidly and extract data features on the fly. Since the data is transforming permanently, these systems must be capable of adapting their topology.

From the algorithmic point of view, the evolving system should be able to carry out some parametric adaptation. Stated in another way, it has to be contributed by a set of parameters along with adaptation of the required tweaks to be implemented effectively [16-21].

Let's dwell on the fact that deep neural networks (DNNs) have gained a high impact on data processing recently [22-25]. Although this class of networks is quite bulky when speaking of the computational implementation. And there's a high plausibility that the overfitting problem takes a place while dealing with a short training data set.

As an alternative view, it is also reasonable to generate DNN architectures on the grounds of the Group Method of Data Handling (GMDH) [26-39]. In this connection, various systems from the area of Computational Intelligence usually enhance automatically a number of their structure layers for information handling in order to obtain the precision on demand for results. That's a great deal of sense to separate an initial space somehow into a suite of subspaces in lower dimensions and combine the results obtained. The Group Method of Data Handling (GMDH) possesses an apparent benefit from a computational point of view. But its huge drawback is its rather poor adaptation for an online mode. That is a rather smart decision to apply GMDH characteristics to evolving cascade neural networks, although some of these systems might freeze their parameter values [40-43].

A specifically new subject of interest is a combination of hybrid systems of computational intelligence and the GMDH concepts with the general aim of new computational and theoretical results especially for Data Mining and Data Stream Mining [44, 45]. The GMDH-ANN topologies have been considered in [40] in terms of using specific two-input N-Adalines as structural elements. A main purpose of this topological element was to guarantee a quadratic approximation for recovering a non-linear mapping. Meanwhile, estimating the achieved quality could lead to a substantial quantity of hidden layers.

Authors already developed composite R-neurons as topological units for their online hybrid system (joining the paradigms of cascade neural networks and GMDH structures) [43] . A high operating speed and high approximating abilities are main performance indicators. Although both its parameters and framework are being adjusted in an online mode, but it still claims long enough training data sets.

That's a very challenging task when there's an obvious lack of incoming data (a short data set), and the system is not capable of tuning its parameters.

In this regard, it's highly important to offer a hybrid neuro-fuzzy system to be trained in an online fashion and to be able of altering its topology while being trained. That's also very topical to introduce the system that keeps in possession an appreciably lower number of attributes to be adjusted in comparison with other well-known compatible systems.

The Architecture of the Deep GMDH Neuro-Fuzzy System

A structure of the deep GMDH neuro-fuzzy system is given in Fig. 1. The receptive (zero) layer of the system contains a $(n\times 1)$-dimensional vector of input signals $x(k)=(x_1(k), x_2(k), \ldots, x_n(k))^{\mathrm{T}}$ ($k=1,2,\ldots,N$ denotes in this case either an observation in a training set or an index of the current discrete time). This vector is subsequently fed into the first hidden layer that comprises $n_1=c_n^2$ elements (every element owns only two inputs).

There is a special type of elements (the selection block) that accounts for choosing the best node in the strict sense of precision (in terms of an accepted criterion). For instance, the selection block in the first layer $SB^{[1]}$ selects n_1^* ($n_1^*\le n$) signals with the highest accuracy among the output signals $\hat{y}_m^{[1]}(k)$ ($m=1,2,\ldots,0{,}5n(n-1)=c_n^2$) of the first layer nodes $N^{[1]}$.

Afterwards, n_2 pairwise combinations $\hat{y}_l^{[1]*}(k), \hat{y}_p^{[1]*}(k)$ are composed (in most cases $n\le n_2\le 2n$) among the mentioned above n_1^* best outputs. The signals obtained are later propagated to the second hidden layer composed by nodes $N^{[2]}$ in a similar manner to the neurons $N^{[1]}$. Among the signals of this layer $\hat{y}_l^{[2]}$ the selection block $SB^{[2]}$ selects F best neurons by accuracy (e.g. by $\sigma^2_{y_l^{[2]}}$) if the best signal of the second layer is better than the best one of the first hidden layer $\hat{y}_1^{[1]}$ *.

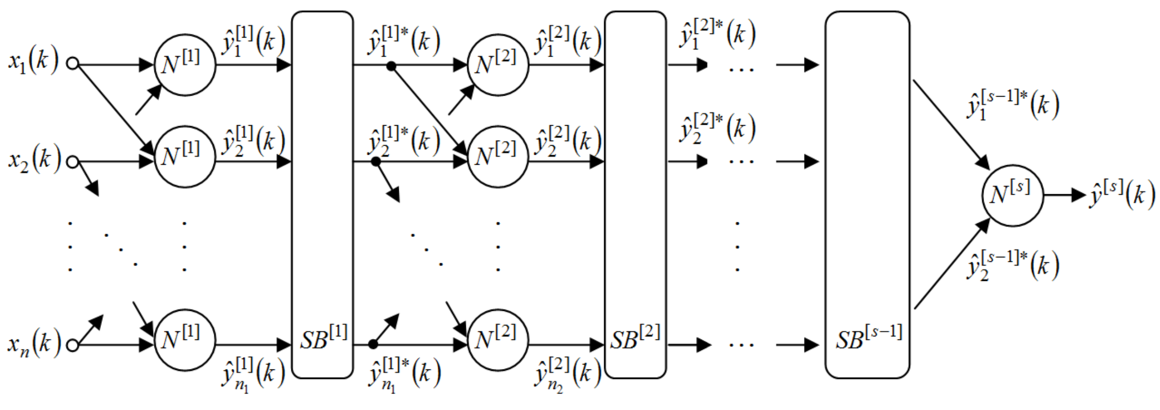

Figure 1. The structure of the deep GMDH neuro-fuzzy system

Other hidden layers forms signals similarly to the second hidden layer. The system evolution process continues until the best signal of the selection block $SB^{[s+1]}$ would be worse than the best signal of the previous sth layer, that is $\sigma^2_{y_i^{[s+1]}} > \sigma^2_{y_i^{[s]}}$. Then we return to the previous layer and choose its best node neuron $N^{[s]}$ in order to form the system output signal $\hat{y}^{[s]}$.

It should be stressed that we obtain not only optimal network structure but well-trained network as well due to GMDH algorithm. Besides, since the training is performed sequentially layer by layer the problems of high dimensionality as well as decaying or exploding gradient vanish. This is very important for deep learning networks.

The Extended Neo-Fuzzy Neuron

A model of the extended NFN was put forward in [46] as a further development and evolution of an ordinary neo-fuzzy neuron submitted by Yamakawa, Miki, and Uchino [47-49].

A traditional version of the neo-fuzzy neuron is a MISO (multiple inputs and a single output) non-linear system that accounts for the permutation

$$\hat{y} = \sum_{i=1}^{n} f_i\left(x_i\right) \tag{1}$$

where x_i signifies an i th component in the input vector $x = \left(x_1, ..., x_i, ..., x_n\right)^{\mathrm{T}} \in R^n$ (of the dimensionality n), $\hat{y}$ marks a scalar output of the neo-fuzzy neuron. In its usual form, NFN embodies multiple (non-linear) synapses NS_i . Their purpose is to modify the i th vector element in x_i into

$$f_i\left(x_i\right) = \sum_{l=1}^{h} w_{li}\mu_{li}\left(x_i\right) \tag{2}$$

where h is the number of membership functions, w_{li} defines a synaptic weight l in the i th non-linear synapse, $l = 1, 2, \ldots, h$, $i = 1, 2, \ldots, n$; $\mu_{li}\left(x_i\right)$ describes the l th membership function in the non-linear synapse i that performs fuzzification of a crisp element x_i . By such manner, the permutation ensured by the NFN could be noted down like

$$\hat{y} = \sum_{i=1}^{n} \sum_{l=1}^{h} w_{li} \mu_{li} \left(x_i \right). \tag{3}$$

The NFN provides the fuzzy inference rule implementation in the form

$$IF\ x_i\ IS\ X_{li}\ THEN\ THE\ OUTPUT\ IS\ w_{li},\ l = 1, 2, \ldots, h \tag{4}$$

which consequently infers that the synapse truthfully endows the 0th order fuzzy inference by Takagi–Sugeno.

As mentioned previously, the NFN’s synapse NS_i covers the 0th order inference by Takagi–Sugeno only producing the simplest Wang–Mendel neuro-fuzzy system [50–52]. It seems quite valid to expand approximating capabilities of this computational node by introducing a specified topological element to have been called an “extended nonlinear synapse” (ENS_i) and to develop the “extended neo-fuzzy neuron” (ENFN) that embraces ENS_i units instead of conventional synapses.

After introducing additional parameter values in extended nonlinear synapse

$$\varphi_{li} \left(x_i \right) = \mu_{li} \left(x_i \right) \left(w_{li}^0 + w_{li}^1 x_i + w_{li}^2 x_i^2 + \ldots + w_{li}^p x_i^p \right), \tag{5}$$

$$\begin{aligned} f_i \left(x_i \right) &= \sum_{l=1}^{h} \mu_{li} \left(x_i \right) \left(w_{li}^0 + w_{li}^1 x_i + w_{li}^2 x_i^2 + \ldots + w_{li}^p x_i^p \right) = \\ &= w_{1i}^0 \mu_{1i} \left(x_i \right) + w_{1i}^1 x_i \mu_{1i} \left(x_i \right) + \ldots + w_{1i}^p x_i^p \mu_{1i} \left(x_i \right) + \\ &+ w_{2i}^0 \mu_{2i} \left(x_i \right) + \ldots + w_{2i}^p x_i^p \mu_{2i} \left(x_i \right) + \ldots + w_{hi}^p x_i^p \mu_{hi} \left(x_i \right), \end{aligned} \tag{6}$$

$$w_i = \left(w_{1i}^0, w_{1i}^1, \ldots, w_{1i}^p, w_{2i}^0, \ldots, w_{2i}^p, \ldots, w_{hi}^p \right)^{\mathrm{T}}, \tag{7}$$

$$\tilde{\mu}_i(x_i) = \left(\mu_{1i}(x_i), x_i\mu_{1i}(x_i), \ldots, x_i^p\mu_{1i}(x_i), \mu_{2i}(x_i), \ldots, x_i^p\mu_{2i}(x_i), \ldots, x_i^p\mu_{hi}(x_i)\right)^{\mathrm{T}}, \qquad (8)$$

it can be marked down like

$$f_i(x_i) = w_i^T\tilde{\mu}_i(x_i), \qquad (9)$$

$$\hat{y} = \sum_{i=1}^{n} f_i(x_i) = \sum_{i=1}^{n} w_i^{\mathrm{T}}\tilde{\mu}(x_i) = \tilde{w}^{\mathrm{T}}\tilde{\mu}(x) \qquad (10)$$

where $\tilde{\mu}(x) = \left(\tilde{\mu}_1^T(x_1), \ldots, \tilde{\mu}_i^T(x_i), \ldots, \tilde{\mu}_n^T(x_n)\right)^{\mathrm{T}}$, $\tilde{w}^{\mathrm{T}} = \left(w_1^T, \ldots, w_i^T, \ldots, w_n^T\right)^{\mathrm{T}}$.

It can be noted easily that the ENFN holds $(p+1)hn$ parameters (synaptic weights) to be adjusted and the fuzzy inference realized by each ENS_i is

$$IF\ \ x_i\ \ IS\ \ X_{li}\ \ THEN\ \ THE\ \ OUTPUT\ \ IS\ \ w_{li}^0 + w_{li}^1 x_i + \ldots + w_{li}^p x_i^p,\ \ l = 1, 2, \ldots, h \qquad (11)$$

which ties up to the Takagi-Sugeno inference of the pth order.

The ENFN’s framework is not so complicated in comparison with the conventional neuro-fuzzy system. The architecture of the extended neo-fuzzy neuron and the extended neo-fuzzy synapse are given in Fig. 2 and Fig. 3.

The usage of the scatter partitioning of the input space [21] can cause the appearing of “gaps” in the fuzzified space. To avoid this problem one can use the bell-shaped constructions with non-strictly local receptive support as membership functions. Mostly the Gaussians are used as membership functions of the first layer

$$\mu_{li}\left(x_i\left(k\right)\right)=\exp\left(-\frac{\left(x_i\left(k\right)-c_{li}\left(k\right)\right)^2}{2\sigma_{li}^2\left(k\right)}\right) \tag{12}$$

where $c_{li}\left(k\right)$ is the parameter that defines the center of the membership function, $\sigma_{li}\left(k\right)$ is the width parameter of this function.

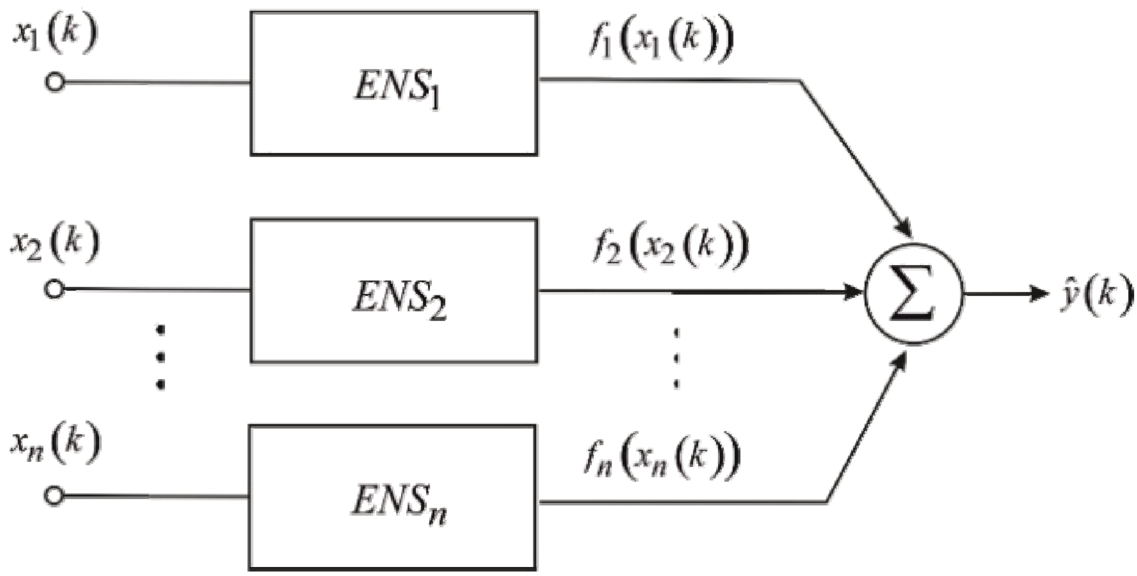

Figure 2. Extended neo-fuzzy neuron

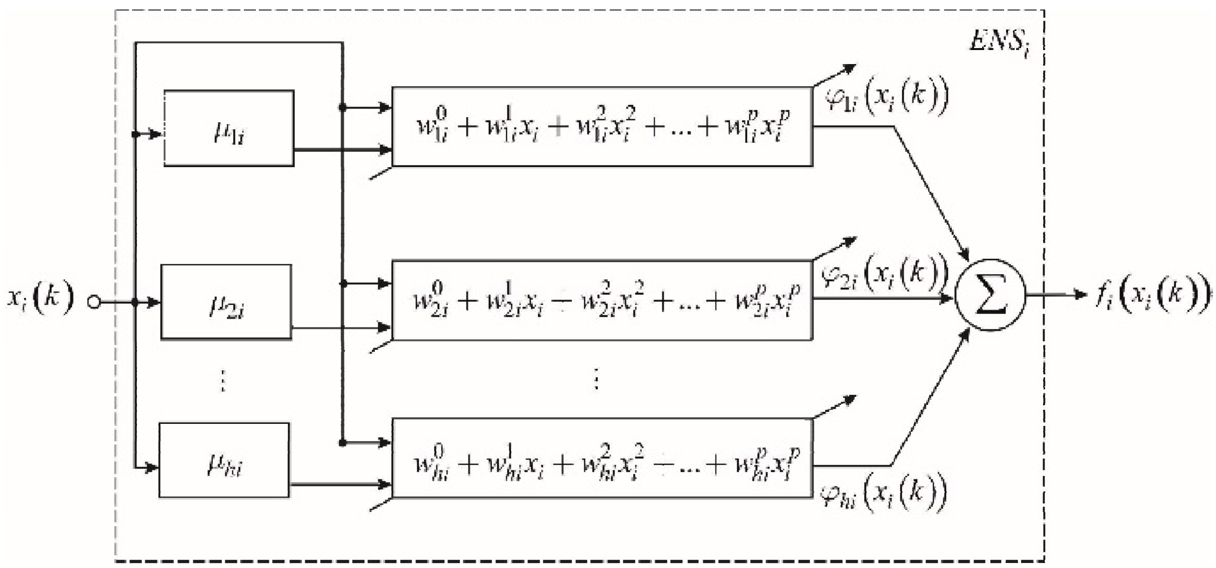

Figure 3. Extended neo-fuzzy synapse

The Adjustment Procedures for All System Parameters

With regard to the fact that the reference signal $\hat{y}_s^{[1]}(k)$ in every system node is in linear dependence on the configurable synaptic weights w_{li}, one can make use of both either the established least squares method or its recurrent version to tune them. If the data to be trained is not stationary, it is feasible enough to apply the exponentially weighted recurrent least squares algorithm to adjust the weights as represented by

$$\begin{cases} \tilde{w}(k) = \tilde{w}(k-1) + \dfrac{P(k-1)\left(y(k) - \left(\tilde{w}(k-1)\right)^{\mathrm{T}} \tilde{\mu}\left(x(k)\right)\right)\tilde{\mu}\left(x(k)\right)}{\alpha + \left(\tilde{\mu}\left(x(k)\right)\right)^{\mathrm{T}} P(k-1)\tilde{\mu}\left(x(k)\right)}, \\ P(k) = \dfrac{1}{\alpha}\left(P(k-1) - \dfrac{P(k-1)\tilde{\mu}\left(x(k)\right)\left(\tilde{\mu}\left(x(k)\right)\right)^{\mathrm{T}} P(k-1)}{\alpha + \left(\tilde{\mu}\left(x(k)\right)\right)^{\mathrm{T}} P(k-1)\tilde{\mu}\left(x(k)\right)}\right) \end{cases} \tag{13}$$

(where $0 < \alpha \le 1$ denotes a forgetting feature, and $y(k)$ implies the reference signal) or the exponentially weighted gradient learning procedure

$$\begin{cases} \tilde{w}(k) = \tilde{w}(k-1) + \dfrac{\left(y(k) - \tilde{w}^{\mathrm{T}}(k-1)\tilde{\mu}\left(x(k)\right)\right)\tilde{\mu}\left(x(k)\right)}{\beta(k)}, \\ \beta(k) = \alpha\beta(k-1) + \left\|\tilde{\mu}\left(x(k)\right)\right\|^2, 0 \le \alpha \le 1. \end{cases} \tag{14}$$

The process of tuning both parameters of the centers and the synaptic weights may be implemented by means of the gradient procedures for minimization of the learning criterion

$$E(k) = \frac{1}{2}\left(y(k) - \hat{y}(k)\right)^2 = \frac{1}{2}\left(y(k) - \left(\tilde{w}(k)\right)^{\mathrm{T}} \tilde{\mu}\left(x(k)\right)\right)^2 \tag{15}$$

in the form of

$$\begin{cases} c_{ri}(k) = c_{ri}(k-1) - \eta_c \dfrac{\partial E(k)}{\partial c_{ri}}, \\ \tilde{\sigma}_{ri}^2(k) = \tilde{\sigma}_{ri}^2(k-1) - \eta_\sigma \dfrac{\partial E(k)}{\partial \tilde{\sigma}_{ri}^2} \end{cases} \tag{16}$$

where $r = 1, 2, ..., h$; η_c , η_σ signify learning rates for the centers' and the widths' parameters correspondingly, $\tilde{\sigma}_{ri}^2(k) = -0,5\sigma_{ri}^{-2}(k)$. Based on the previous expressions, the following expressions are obtained

$$\begin{cases} \dfrac{\partial E(k)}{\partial c_{ri}} = \left((\tilde{w}(k))^{\mathrm{T}} \tilde{\mu}(x(k)) - y(k) \right) \dfrac{\partial f_i(x_i(k))}{\partial c_{ri}}, \\ \dfrac{\partial E(k)}{\partial \tilde{\sigma}_{ri}^2} = \left((\tilde{w}(k))^{\mathrm{T}} \tilde{\mu}(x(k)) - y(k) \right) \dfrac{\partial f_i(x_i(k))}{\partial \tilde{\sigma}_{ri}^2}. \end{cases} \tag{17}$$

Following on from (17), the derivatives $\dfrac{\partial f_i(x_i(k))}{\partial c_{ri}}$ and $\dfrac{\partial f_i(x_i(k))}{\partial \tilde{\sigma}_{ri}^2}$ could be presented in terms of

$$\begin{cases} \dfrac{\partial f_i(x_i(k))}{\partial c_{ri}} = \dfrac{\partial \varphi_{ri}(x_i(k))}{\partial c_{ri}} = \sum_{t=0}^{p} w_{ri}^t x_i^t \dfrac{\partial \mu_{ri}(x_i(k))}{\partial c_{ri}}, \\ \dfrac{\partial f_i(x_i(k))}{\partial \tilde{\sigma}_{ri}^2} = \dfrac{\partial \varphi_{ri}(x_i(k))}{\partial \tilde{\sigma}_{ri}^2} = \sum_{t=0}^{p} w_{ri}^t x_i^t \dfrac{\partial \mu_{ri}(x_i(k))}{\partial \tilde{\sigma}_{ri}^2}, \end{cases} \tag{18}$$

Defined on the ground of (12), the derivatives $\dfrac{\partial \mu_{ri}(x_i(k))}{\partial c_{ri}}$ and $\dfrac{\partial \mu_{ri}(x_i(k))}{\partial \tilde{\sigma}_{ri}^2}$ can be represented as

$$\begin{cases} \dfrac{\partial \mu_{ri}\left(x_i\left(k\right)\right)}{\partial c_{ri}} = \dfrac{x_i\left(k\right) - c_{ri}\left(k\right)}{\sigma_{ri}^2\left(k\right)} \exp\left(-\dfrac{\left(x_i\left(k\right) - c_{ri}\left(k\right)\right)^2}{2\sigma_{ri}^2\left(k\right)}\right), \\ \dfrac{\partial \mu_{ri}\left(x_i\left(k\right)\right)}{\partial \tilde{\sigma}_{ri}^2} = \left(x_i\left(k\right) - c_{ri}\left(k\right)\right)^2 \exp\left(-\dfrac{\left(x_i\left(k\right) - c_{ri}\left(k\right)\right)^2}{2\sigma_{ri}^2\left(k\right)}\right). \end{cases} \quad (19)$$

In this fashion, all the system nodes' parameters (synaptic weights, centers and width parameters for the membership functions) may be adjusted. Concerning the successive layers, the nodes' parameters are usually tuned quite the same way as the nodes in the first hidden layer. It should be noted that inputs of the sth layer are a pairwise combination of the signals $\hat{y}_l^{[s-1]*}, \hat{y}_p^{[s-1]*}$ formed by the selection block $SB^{[s-1]}$. The reference signal $y(k)$ is the same one for all the blocks (layers) of the evolving complex system.

The Experimental Investigations

The Darwin sea level pressure data set was chosen from the Data Market data storage to investigate the efficiency of the offered deep GMDH-system and its learning schemes. It was mainly used for non-stationary signals' prediction. The data set presents chiefly a monthly sea level pressure for a period of more than a century (1882-1998). The general size of this data sample is 1400 observations. The system used 1100 observations for training and 300 observations for testing. To estimate the efficiency of the proposed neuro-fuzzy system a multilayer perceptron and ANFIS were also considered for solving the same task. The results obtained were estimated according to the MSE quality criterion.

Table 1 gives a demonstration of the systems' performance. The proposed deep GMDH-system illustrated quite good results while handling the prediction task. It is worth mentioning that its training time was short enough compared to analogues. At the same time, its forecasting results were the best ones for this data set. Fig. 4 demonstrates a fragment of the learning process.

Table 1. Experimental results

System	Training error	Test error	Training time, sec
The proposed deep GMDH system	0.0146	0.0156	0.2067
MLP	0.0150	0.0168	0.2500
ANFIS	0.0157	0.0165	0.2031

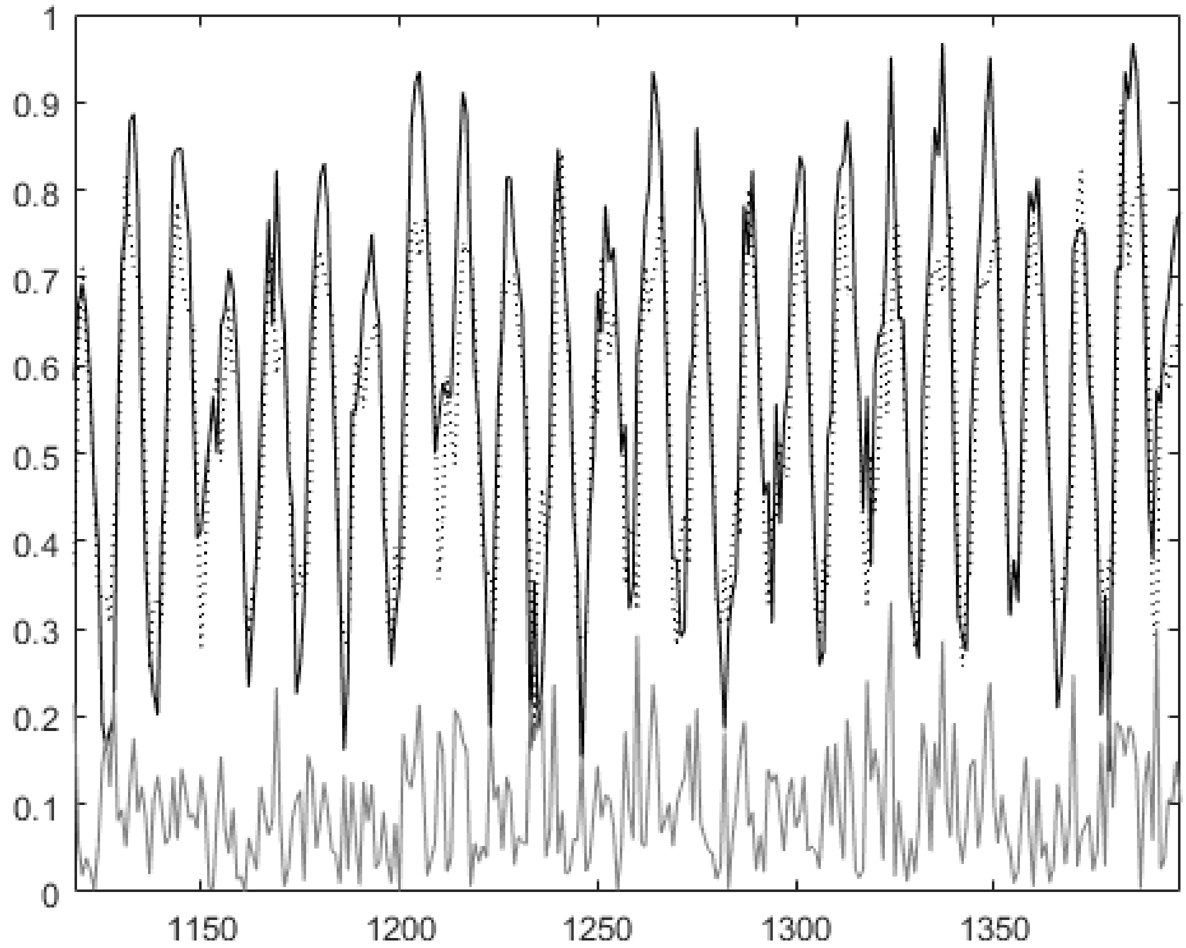

Figure 4. Prediction results

Conclusion

The deep evolving neuro-fuzzy system is suggested in this paper. The hybrid system is grounded on both the Group Method of Data Handling and the concept of evolving systems that makes it possible to define both optimal parameter values and the best structure in every specific case.

The important property of the suggested system is that it doesn’t require any high data volumes to get trained. Adjusting parameters in a parallel fashion gives an option of increasing a processing speed of data handling.

Bibliography

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

[2] Du, K.-L., Swamy, M. Neural Networks and Statistical Learning. Springer-Verlag, London, 2014.

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

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

[5] Kasabov, N. Evolving Connectionist Systems. Springer-Verlag London, 2007.

[6] Haykin, S. Neural Networks and Learning Machines. NJ: Prentice Hall, 2009.

[7] Hu, Zh., Bodyanskiy, Ye. V., Tyshchenko, O.K. A Deep Cascade Neuro-Fuzzy System for High-Dimensional Online Fuzzy Clustering. Proc. of the 2016 IEEE First Int. Conf. on Data Stream Mining & Processing (DSMP), 2016, pp.318-322. DOI: 10.1109/DSMP.2016.7583567.

[8] Bodyanskiy, Ye., Tyshchenko, O., Deineko, A. An Evolving Radial Basis Neural Network with Adaptive Learning of Its Parameters and Architecture. Automatic Control and Computer Sciences, Vol. 49, No. 5, 2015, pp. 255-260. DOI: 10.3103/S0146411615050028.

[9] Bodyanskiy, Ye., Tyshchenko, O., Kopaliani, D. Adaptive learning of an evolving cascade neo-fuzzy system in data stream mining tasks. Evolving Systems, Vol. 7, No. 2, 2016, pp.107-116. DOI: 10.1007/s12530-016-9149-5.

[10] Angelov, P., Filev, D., Kasabov, N. Evolving Intelligent Systems: Methodology and Applications. John Willey and Sons, 2010.

[11] Hu, Zh., Bodyanskiy, Ye. V., Tyshchenko, O. K. A Cascade Deep Neuro-Fuzzy System for High-Dimensional Online Possibilistic Fuzzy Clustering. Proc. of the XI-th International Scientific and Technical Conference “Computer Science and Information Technologies” (CSIT 2016), 2016, pp. 119-122. DOI: 10.1109/STC-CSIT.2016.7589884.

[12] Lughofer, E. Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications. Springer Berlin, 2011.

[13] Bodyanskiy, Ye., Tyshchenko, O., Kopaliani, D. An evolving neuro-fuzzy system for online fuzzy clustering. Proc. Xth Int. Scientific and Technical Conf. “Computer Sciences and Information Technologies (CSIT’2015)”, 2015, pp. 158-161. DOI: 10.1109/STC-CSIT.2015.7325456.

[14] Bodyanskiy, Ye., Tyshchenko, O., Kopaliani, D. A hybrid cascade neural network with an optimized pool in each cascade. Soft Computing. A Fusion of Foundations, Methodologies and Applications, Vol.19, No.12, 2015, pp. 3445-3454. DOI: 10.1007/s00500-014-1344-3.

[15] Hu, Zh., Bodyanskiy, Ye. V., Tyshchenko, O. K., Boiko, O. O. An Evolving Cascade System Based on a Set of Neo-Fuzzy Nodes. International Journal of Intelligent Systems and Applications (IJISA), Vol. 8(9), 2016, pp. 1-7. DOI: 10.5815/ijisa.2016.09.01.

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

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

[17a] Eds. Kacprzyk, J., Pedrycz, W. Springer Handbook of Computational Intelligence. Springer-Verlag Berlin Heidelberg, 2015.

[18] Eds. Pedrycz, W., Chen, S.-M., Information Granularity, Big Data, and Computational Intelligence. Springer In-ternational Publishing Switzerland, 2015.

[19] Takagi, T., Sugeno, M. Fuzzy identification of systems and its application to modeling and control. IEEE Trans. on Systems, Man, and Cybernetics, No. 15, 1985, pp. 116-132.

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

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

[22] Hinton, G., Osindero, S., Teh, Y.-W. A fast learning algorithm for deep belief nets. Neural Computation, vol. 18, no. 7, 2006, pp. 1527-1554.

[23] Bengio, Y., LeCun, Y., Hinton, G. Deep learning. Nature, no. 521, 2015, pp. 436-444.

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

[25] Ivakhnenko, A. The group method of data handling – a rival of the method of stochastic approximation. Soviet Automatic Control, vol. 13, no. 3, 1968, pp. 43-55.

[26] Ivakhnenko, A. Polynomial theory of complex systems. IEEE Transaction on Systems, Man, Cybernetics, vol. 1, no. 4, 1971, pp. 364-378.

[27] Ivakhnenko, A., Ivakhnenko, G., Muller, A. Self-organization of the neural networks with active neurons. Pattern Recognition and Image Analysis, vol. 4, no. 2, 1994, pp. 177-178.

[28] Zaychenko, Yu. The fuzzy Group Method of Data Handling and its application for economical processes forecasting, Scientific Inquiry, 7, No.1, 2006, pp. 83-96.

[29] Kondo, T. GMDH neural network algorithm using the heuristic self-organization method and its application to the pattern identification problem. Proceedings of the 37th SICE Annual Conference, Japan, Tokyo, Jul. 1998, pp. 1143–1148. doi: 10.1109/SICE.1998.742993

[30] Kondo, T. Identification of radial basis function networks by using revised GMDH-type neural networks with a feedback loop. Proceedings of the 41st SICE Annual Conference, Japan, Osaka, vol. 5, Aug. 2002, pp. 2882-2887. doi: 10.1109/SICE.2002.1195514

[31] Kondo, T., Pandya, A. S., Nagashino, H. GMDH-type neural network algorithm with a feedback loop for structural identification of RBF neural network. KES Journal, Vol. 11, No. 3, 2007, pp. 157-168.

[32] Kondo, T., Ueno, J., Pandya, A. S. Multilayered GMDH-Type Neural Network with Radial Basis Functions and its Application to 3-Dimensional Medical Image Recognition of the Liver. JACIII, Vol. 11, No. 1, 2007, pp. 96-104.

[33] Kondo, T. Revised GMDH-type neural network using artificial intelligence and its application to medical image diagnosis. 2011 IEEE Workshop On Hybrid Intelligent Models And Applications (HIMA 2011), 2011, pp. 76-83.

[33a] Kondo, T., Ueno, J., Takao, Sh. Feedback GMDH-type Neural Network Self-Selecting Various Functions and Its Application to Medical Image Diagnosis of Lung Cancer. 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2012), 2012, pp. 203-208.

[34] Kondo, T., Ueno, J., Takao, Sh. Medical Image Diagnosis of Liver Cancer by Revised GMDH-type Neural Network Using Feedback Loop Calculation. 2012 Sixth International Conference on Genetic and Evolutionary Computing (ICGEC 2012), 2012, pp. 237-240.

[35] Kondo, T., Ueno, J., Takao, Sh. Hybrid Multi-layered GMDH-type Neural Network Using Principal Component Regression Analysis and its Application to Medical Image Diagnosis of Liver Cancer. 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems (KES 2013), 2013, pp. 172-181.

[36] Kondo, T., Ueno, J., Takao, Sh. Medical Image Recognition of Abdominal Multi-organs by Hybrid Multi-layered GMDH-type Neural Network Using Principal Component-Regression Analysis. Second International Symposium on Computing and Networking (CANDAR 2014), 2014, pp. 157-163.

[37] Bodyanskiy, Ye., Vynokurova, O., Teslenko, N. Cascade GMDH-wavelet-neuro-fuzzy network. Proceedings of International Workshop Inductive Modelling, Ukraine, Kyiv, Sep. 2011, pp. 22-30.

[38] Bodyanskiy, Ye., Vynokurova, O., Dolotov, A., Kharchenko, O. Wavelet-neuro-fuzzy-network structure optimization using GMDH for the solving forecasting tasks. Proceedings of International Workshop Inductive Modelling, Ukraine, Kyiv, Sep. 2013, pp. 61-67.

[39] Bodyanskiy, Ye., Kharchenko, O., Vynokurova, O. Hybrid cascade neural network based on wavelet-neuron. Information Theories and Applications, Vol. 18, Issue 4, 2011, pp. 335-343.

[40] Bodyanskiy, Ye., Zaychenko, Yu., Pavlikovskaya, E., Samarina, M., Viktorov, Ye. The neo-fuzzy neural network structure optimization using the GMDH for the solving forecasting and classification problems, Proc. Int. Workshop on Inductive Modeling, Krynica, Poland, 2009, pp. 77-89.

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

[42] Aggarwal, C. C. Data Streams: Models and Algorithms (advances in database systems). New York: Springer, 2007.

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

[44] Yamakawa, T., Uchino, E., Miki, T., Kusanagi, H. A neo-fuzzy neuron and its applications to system identification and prediction of the system behavior. Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks, 1992, pp. 477-483.

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

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

[47] Wang, L. X., Mendel, J. M. Fuzzy basis functions, universal approximation and orthogonal least squares learning. IEEE Trans. on Neural Networks, No. 3, 1993, pp. 807-814.

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

[49] Osowski, S. Sieci neuronowe do przetwarzania informacji. Warszawa: Oficijna Wydawnicza Politechniki Warszawskiej, 2006. (in Polish)

Acknowledgement

"The paper is published with financial support by the project ITHEA XXI of the Institute of Information Theories and Applications FOI ITHEA (www.ithea.org) and the Association of Developers and Users of Intelligent Systems ADUIS Ukraine (www.aduis.com.ua)."

Authors' Information

Yuri Zaychenko *– Professor, doctor of technical sciences, Institute for applied system analysis, NTUU “KPI”, 03056, Ukraine, Kyiv, Peremogi pr. 37, Corpus 35; e-mail: zaychenkoyuri@ukr.net*

Major Fields of Scientific Research: Information systems, Fuzzy logic, Decision making theory

Yevgeniy Bodyanskiy *– Professor, D.Sc., Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, 61000, Ukraine, Kharkiv, Nauky ave., 14; e-mail: yevgeniy.bodyanskiy@nure.ua*

Major Fields of Scientific Research: Hybrid systems, Deep Learning, Computational Intelligence

Oleksii Tyshchenko *– Ph.D., Senior Researcher, Control Systems Research Laboratory, Kharkiv National University of Radio Electronics. 61000, Ukraine, Kharkiv, Nauky ave., 14; e-mail: lehatish@gmail.com*

Major Fields of Scientific Research: Cascade Neuro-Fuzzy Systems, Computational Intelligence, Machine Learning.

Olena Boiko – Ph.D., Junior Researcher *Control Systems Research Laboratory, Kharkiv National University of Radio Electronics. 61000, Ukraine, Kharkiv, Nauky ave., 14; e-mail: olena.boiko@ukr.net*

Major Fields of Scientific Research: Deep Learning, Hybrid Systems, Machine Learning

Galib Hamidov – Ph.D, *Direktor Information Tecnologies Department of Azerisiq JSC, Baku, Azerbaijan, Kazimzade-20; e-mail: galib..hamidov@gmail.com*

Major Fields of Scientific Research: Information technologies, Data Mining