Kharkiv National University of Radio Electronics

Control Systems Research Laboratory

International Workshop on Inductive Modeling IWIM’2017

A Hybrid Growing ENFN-Based Neuro-Fuzzy System and its Rapid Deep Learning

Yevgeniy Bodyanskiy,
Dr. Sc.,
Prof. in Artificial Intelligence Department,
Scientific Head at CSRL

Oleksii Tyshchenko,
Ph.D., Senior Researcher at CSRL

Zhengbing Hu,
Ph.D., Associate Prof.,
Central China Normal University

September 5-8, 2017 Lviv, Ukraine

Hybrid systems of Computational Intelligence

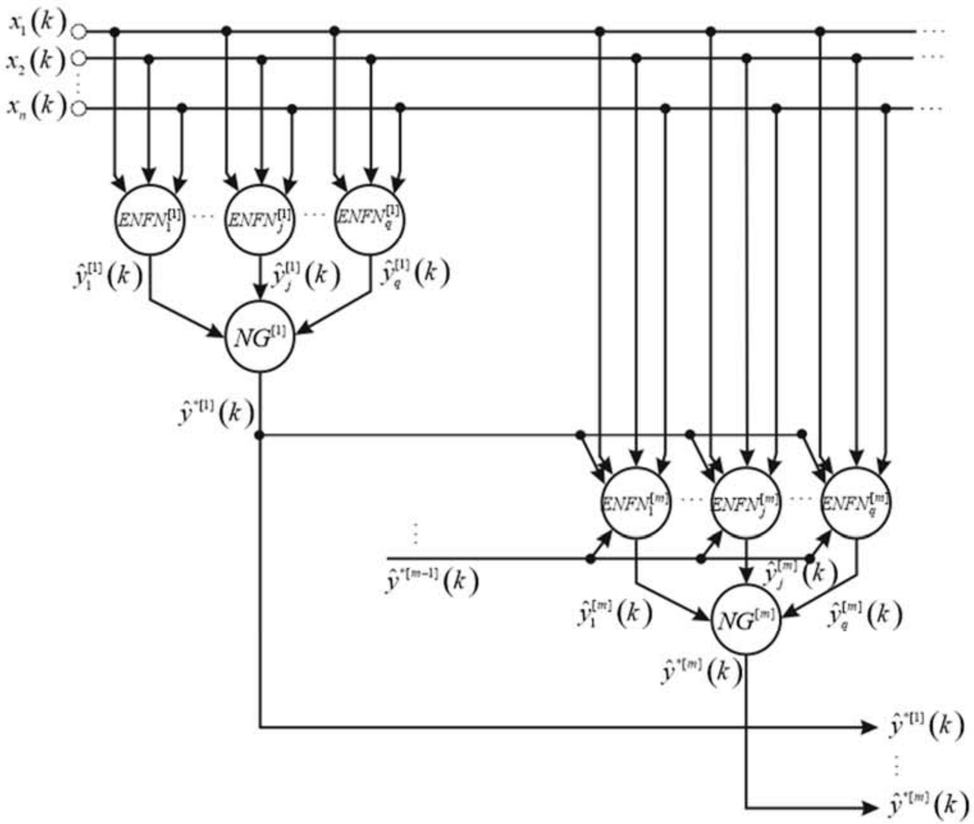
- Neural networks
- Wavelet theory
- Type-1 fuzzy systems
- Type-2 fuzzy systems
- Cascade neural networks
- Evolving methods

Hybrid evolving adaptive wavelet-neuro-fuzzy-systems

Advantages:

- identification of signals' local features;
- learning ability and universal approximation properties;
- linguistic interpretability;
- minimizing subjectivity for membership functions' selection;
- adaptive synthesis of an optimal architecture for a neural network;
- dimension reduction of an input space;
- selection of input variables.

An architecture of the hybrid growing system



An input signal

$$x(k) = \left(x_1(k), x_2(k), \ldots, x_n(k)\right)^T$$

These signals are later taken to an input of each unit $ENFN_j^{[m]}$ in every layer.

Output signals $\hat{y}_j^{[m]}(k)$ of these nodes form an ensemble and are handled by a node of generalization $NG^{[m]}$ which synthesizes an output signal $\hat{y}^{*[m]}(k)$

A signal for the first layer is $x(k) \in R^n$

A signal for the second layer is

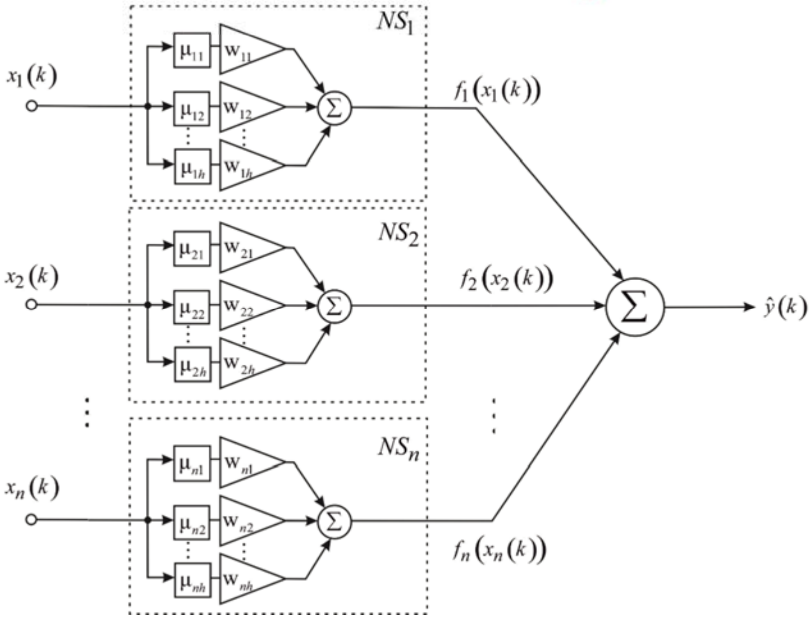
$$x^{[2]}(k) = \left(x^T(k), \hat{y}^{*[1]}(k)\right)^T \in R^{n+1}$$

A signal for the m-th layer is

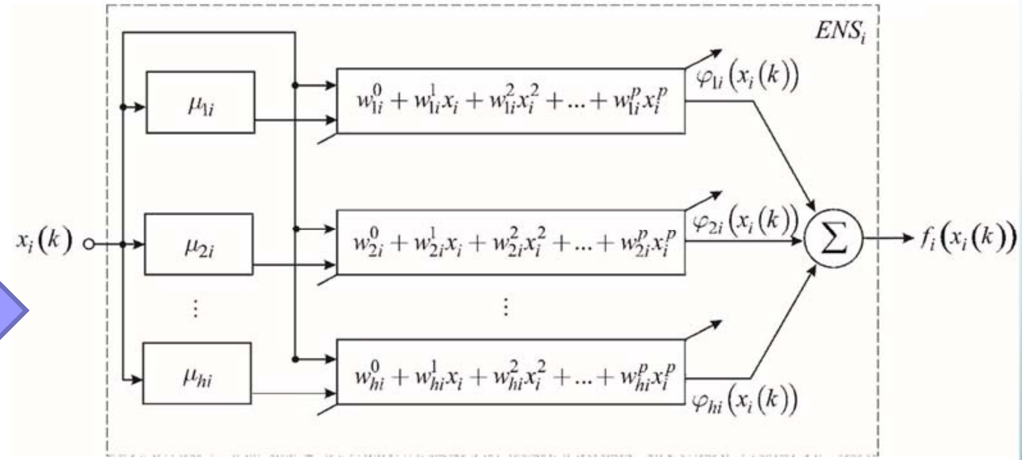
$$x^{[m]}(k) = \left(x^T(k), \hat{y}^{*[1]}(k), \hat{y}^{*[2]}(k), \ldots, \hat{y}^{*[m-1]}(k)\right)^T \in R^{n+m-1}$$

An extended neo-fuzzy neuron

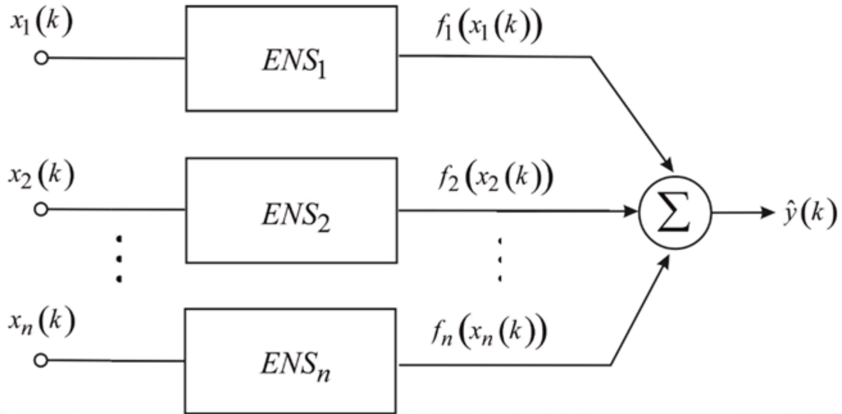
A conventional neo-fuzzy neuron



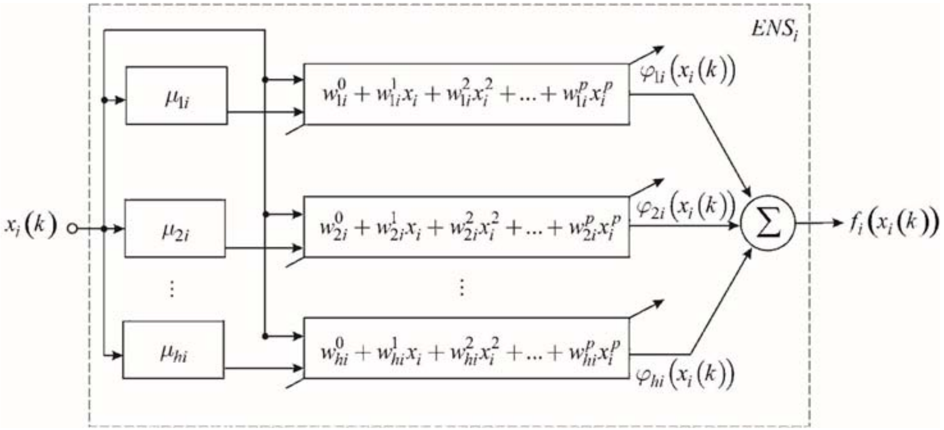
An extended neo-fuzzy synapse



An extended neo-fuzzy neuron



An extended neo-fuzzy neuron



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

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

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

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

$$\tilde{w}^T = \left(w_1^T, \ldots, w_i^T, \ldots, w_n^T\right)^T$$

$$\tilde{\mu}(x) = \left(\tilde{\mu}_1^T(x_1), \ldots, \tilde{\mu}_i^T(x_i), \ldots, \tilde{\mu}_n^T(x_n)\right)^T$$

$$f_i(x_i) = \sum_{l=1}^{h} \mu_{li}(x_i)\left(w_{li}^0 + w_{li}^1 x_i + w_{li}^2 x_i^2 + \ldots + w_{li}^p x_i^p\right) =$$
$$= w_{1i}^0\mu_{1i}(x_i) + w_{1i}^1 x_i\mu_{1i}(x_i) + \ldots + w_{1i}^p x_i^p\mu_{1i}(x_i) +$$
$$+ w_{2i}^0\mu_{2i}(x_i) + \ldots + w_{2i}^p x_i^p\mu_{2i}(x_i) + \ldots + w_{hi}^p x_i^p\mu_{hi}(x_i),$$
$$w_i = \left(w_{1i}^0, w_{1i}^1, \ldots, w_{1i}^p, w_{2i}^0, \ldots, w_{2i}^p, \ldots, w_{hi}^p\right)^T,$$

IF x_i *IS* X_{li} *THEN THE OUTPUT IS*

$$w_{li}^0 + w_{li}^1 x_i + \ldots + w_{li}^p x_i^p, \; l = 1, 2, \ldots, h$$

A learning procedure

A learning criterion for the node ***j*** in the cascade ***m*** is calculated according to

$$E_j^{[\mathrm{m}]}(k)=\tfrac{1}{2}\left(e_j^{[\mathrm{m}]}(k)\right)^2=\tfrac{1}{2}\left(y(k)-\tilde{w}_j^{[\mathrm{m}]\mathrm{T}}(k-1)\,\tilde{\mu}^{[m]}\left(x^{[m]}(k)\right)\right)^2 \quad (1)$$

where denominates a learning error $e_j^{[\mathrm{m}]}(k)$ at the ***k***-th time step, $y(k)$ marks in an external reference signal.

While training the proposed system, it turns out to be more effective to use a criterion of a more general form

$$E_j^{[\mathrm{m}]}(k)=\sum_{\tau=1}^{k}\alpha^{k-\tau}\left(e_j^{[\mathrm{m}]}(k)\right)^2 \quad (2)$$

A traditional exponentially weighted recurrent method of the least squares (EWRLSM) is

$$\begin{cases}\tilde{w}_j^{[\mathrm{m}]}(k)=\tilde{w}_j^{[\mathrm{m}]}(k-1)+\dfrac{P_j^{[\mathrm{m}]}(k-1)e_j^{[\mathrm{m}]}(k)\,\tilde{\mu}^{[m]}\left(x^{[\mathrm{m}]}(k)\right)}{\alpha_j+\tilde{\mu}^{[m]\mathrm{T}}\left(x^{[\mathrm{m}]}(k)\right)P_j^{[\mathrm{m}]}(k-1)\,\tilde{\mu}^{[m]}\left(x^{[\mathrm{m}]}(k)\right)},\\[2ex] P_j^{[\mathrm{m}]}(k)=\dfrac{1}{\alpha_j}\left(P_j^{[\mathrm{m}]}(k-1)-\dfrac{P_j^{[\mathrm{m}]}(k-1)\,\tilde{\mu}^{[m]}\left(x^{[\mathrm{m}]}(k)\right)\tilde{\mu}^{[m]\mathrm{T}}\left(x^{[\mathrm{m}]}(k)\right)P_j^{[\mathrm{m}]}(k-1)}{\alpha_j+\tilde{\mu}^{[m]\mathrm{T}}\left(x^{[\mathrm{m}]}(k)\right)P_j^{[\mathrm{m}]}(k-1)\,\tilde{\mu}^{[m]}\left(x^{[\mathrm{m}]}(k)\right)}\right)\end{cases} \quad (3)$$

where its own distinctive parameter α_j is used for each unit in the cascade.

A learning procedure

Along with that, it should be also kept in mind that a practical application of the algorithm **(3)** is inhibited by the fact that a so-called effect of the "parameters' blast" for a covariance matrix may take place during the learning process which means an exponential grow of its elements. This unwished phenomenon can be eliminated by choosing values large enough of the forgetting factor , but tracking properties of the algorithm are getting dropped during the learning procedure.

$$\begin{cases} P(k+1)=P(k)-P(k)y^T(k+1)\left(I_m+y(k+1)P(k)y^T(k+1)\right)^{-1}y(k+1)P(k)= \\ =\left(I_m-P(k)y^T(k+1)y(k+1)\right)^{-1}P(k), \\ \mathrm{r}(k+1)=r(k)+y^T(k+1)d(k+1), \\ w^*(k+1)=P(k+1)\mathrm{r}(k+1), \\ w(k+1)=w^*(k+1)+P(k+1)\left(E_h^TP(k+1)E_h\right)^{-1}\left(1-E_h^Tw^*(k+1)\right)E_h. \end{cases}$$

This system of equations is a generalizing view for a multidimensional case of the learning algorithm by Kaczmarz- Widrow-Hoff.

A detailed description of the learning method

Step 1. Several ENFNs should be generated which are different according to their characteristics.

Step 2. Set initial weights' values.

Step 3. Make a setup for all neurons.

Step 4. Output signals of each neuron are fed to a node of generalization.

Step 5. Values of weight coefficients should be adjusted.

Step 6. Synthesize an optimal output of cascades using the expression **(3)**.

Step 7. Set a minimal volume of a data sample.

Step 8. Estimate a prediction accuracy according to the criterion **(1)**.

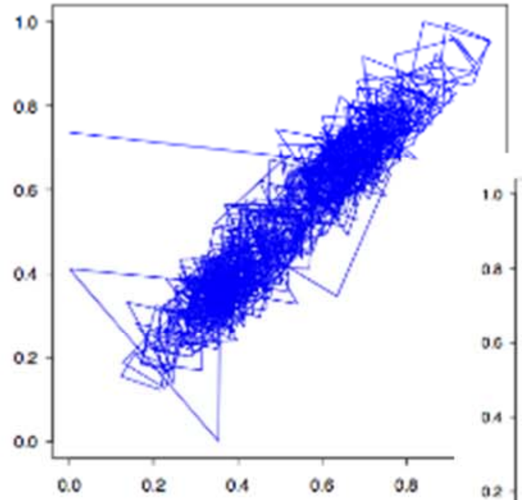
Step 9. If the quality achieved meets the previously formulated demands, the system stops processing.

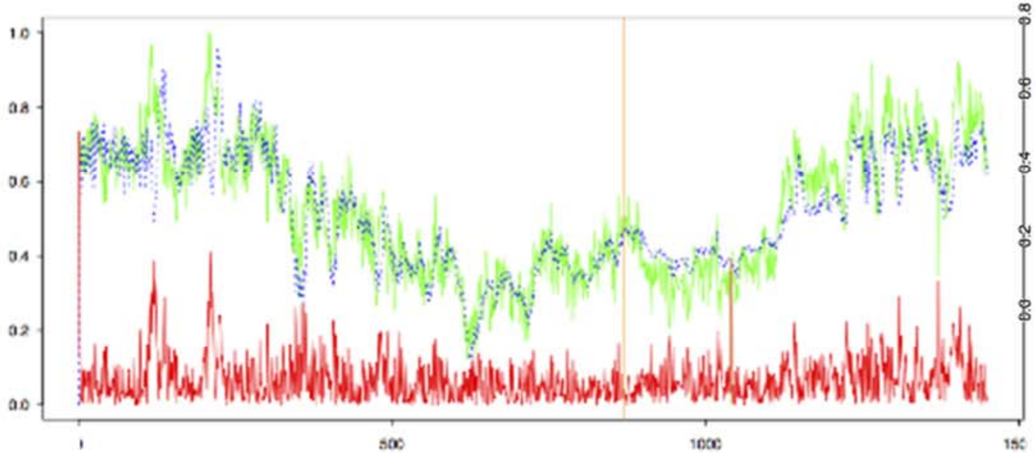
Step 10. Otherwise, the second cascade should be introduced.

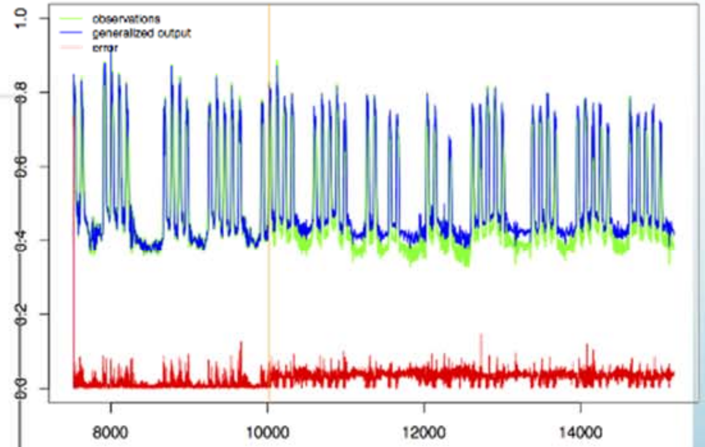
Step 11. A signal $x^{[2]}(k)=\left(x^T(k),\hat{y}^{*[1]}(k)\right)^T \in R^{n+1}$ is fed to an input of the second cascade.

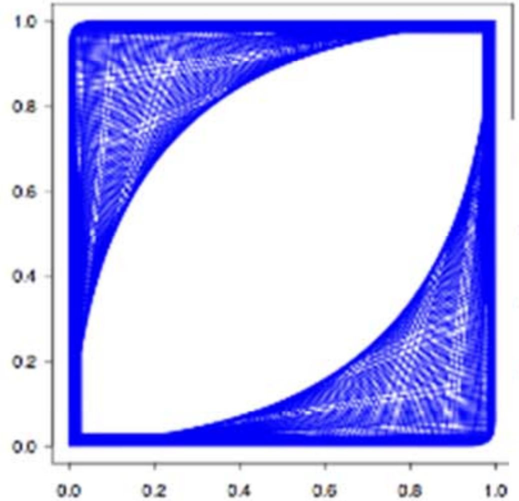
Step 12. The learning procedure should be performed. If upon reaching some threshold on the data volume the required prediction quality is not met, one more cascade should be added.

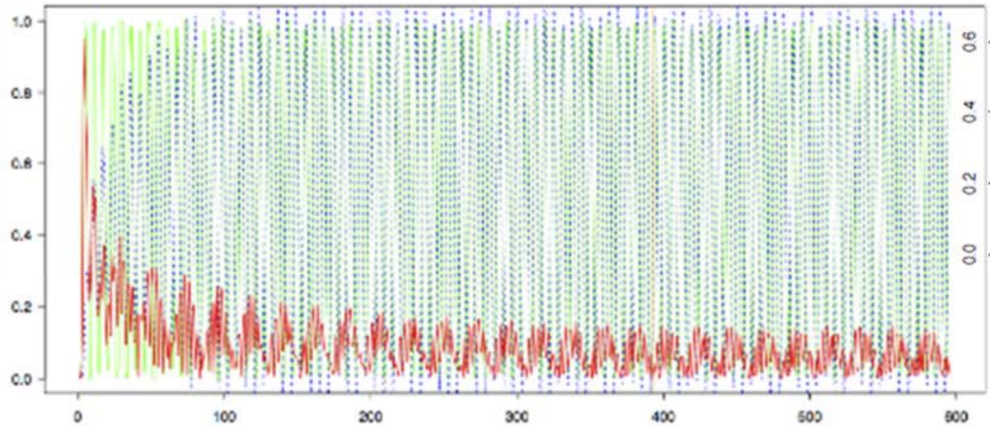
Experiments

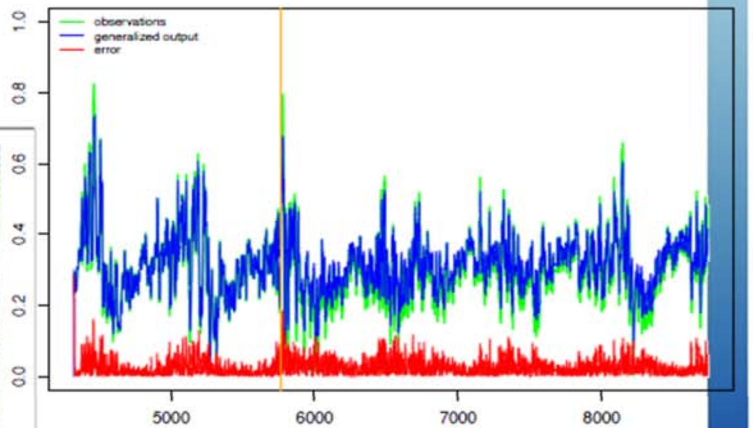












Conclusion

The hybrid growing neuro-fuzzy architecture and its deep learning algorithms were considered and proposed in this report.

The architecture of this hybrid system of computational intelligence was developed by means of the GMDH paradigm and the ideas of evolving systems.

A main achievement of the developed hybrid neuro-fuzzy system is its ability to perform parallel calculations for a data stream based on special-type elements with enhanced approximating properties.

The developed system is rather simple from the effectuation viewpoint, it holds a high processing speed and approximating features.

It can be described by a rather high training speed which makes it possible to process online sequential data. It may be applied to various practical issues in the area of Data Stream Mining.

References

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

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

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

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

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

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

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

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

[9] 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.

[10] Ye. Bodyanskiy, O. Tyshchenko, and D. Kopaliani, "A hybrid cascade neural network with an optimized pool in each cascade", Soft Computing, Vol.19, No.12, pp.3445-3454, 2015.

[11] 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.

[12] 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.

References

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

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

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

[16] Y. Bengio, Y. LeCun, and G. Hinton, "Deep learning," Nature, no. 521, pp. 436–444, May 2015.

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

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

[19] A. Ivakhnenko, "The group method of data handling - a rival of the method of stochastic approximation," Automatica, vol. 6, no. 2, pp. 207–219, 1970.

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

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

[22] A. Ivakhnenko, "Self-organization of neuro net with active neurons for effects of nuclear test explosions forecasting," System Analysis Modeling Simulation, vol. 20, no. 1-2, pp. 107–116, 1995.

[23] A.G. Ivakhnenko and V.S. Stepashko, Noise immunity of modeling. Kiev: Naukova dumka, 1985.

NURE

Kharkiv National University
of Radioelectronics

CSRL

Thanks for
your attention and patience