THE CASCADE ORTHOGONAL NEURAL NETWORK

Yevgeniy Bodyanskiy, Artem Dolotov, Iryna Pliss, Yevgen Viktorov

Abstract: in the paper new non-conventional growing neural network is proposed. It coincides with the Cascade-Correlation Learning Architecture structurally, but uses ortho-neurons as basic structure units, which can be adjusted using linear tuning procedures. As compared with conventional approximating neural networks proposed approach allows significantly to reduce time required for weight coefficients adjustment and the training dataset size.

Keywords: orthogonal activation functions, ortho-synapse, ortho-neuron, cascade orthogonal neural network.

ACM Classification Keywords: I.2.6 Learning – Connectionism and neural nets

Conference: The paper is selected from Sixth International Conference on Information Research and Applications – i.Tech 2008, Varna, Bulgaria, June-July 2008

Introduction

Nowadays artificial neural networks (ANNs) are widely applied for solving large class of problems related to processing information given as time-series or numerical data arrays generated by nonstationary stochastic or chaotic systems. The most attractive properties of the ANNs are their approximating possibilities and learning capabilities.

Traditionally by learning we understand a process of the net's synaptic weights adjustment accordingly to selected optimization procedure of accepted learning criterion [1, 2]. Quality of received result can be improved not only by adjusting weight coefficients but also by adjusting of the neural network architecture (the number of nodes). There are two basic approaches of the neural network architecture adjustment: 1) 'constructive approach' [3 - 5] – starts with simple architecture and gradually adds new nodes during learning; 2) 'destructive approach' [6 - 8] – starts with an initially redundant network and simplifies it throughout learning process.

Obviously, constructive approach needs less computational resources and within the bounds of this approach cascade neural networks (CNNs) [9 - 11] can be marked out. The most efficient representative of CNNs is the Cascade-Correlation Learning Architecture (CasCorLA) [9]. This network begins with the simplest architecture which consists of a single neuron. Throughout learning procedure new neurons are added to the network, producing multilayered structure. It is important that during each learning epoch only one neuron of the last cascade is adjusted. All pre-existing neurons process information with "frozen" weights. CasCorLA authors, S.E. Fahlman and C. Lebiere, point out high speed of learning procedure and good approximation properties of this network. But it should be observed that elementary Rosenblatt perceptrons with hyperbolic tangent activation functions are used in this architecture as nodes. Thus output signal of each neuron is nonlinearly depended from its weight coefficients. Therefore it is necessary to use gradient learning methods such as delta-rule or its modifications, and speed of operation optimization becomes impossible. In connection with the above it seems to be reasonable to synthesize cascade architecture based on elementary nodes with linear dependence of output signal from synaptic weights. It allows to increase speed of synaptic weight adjustment and to reduce minimally required size of training set.

Ortho-neuron

Within the variety of the functional structures, used for approximation of nonlinear dependences, orthogonal polynomials [12, 13] deserve a special attention. They possess quite attractive properties, which make it possible to reduce computational complexity and increase precision of received results. At the present time we can

observe more and more realizations of the orthogonal polynomials theory in the field of neural networks [14 - 21], which demonstrate impressive effectiveness.

Elementary one-dimensional system described in "input-output" space of some unknown functional dependence $y(x)$ can be expressed by the following sum:

$$\hat{y} = \hat{f}(x) = w_0\varphi_0(x) + w_1\varphi_1(x) + ... + w_h\varphi_h(x) = \sum_{j=0}^{h} w_j\varphi_j(x), \quad (1)$$

where x and $y(x)$ are input and output variables of the estimated process correspondingly, $\varphi_j(x)$ – orthogonal polynomial of the *j*-th order (*j* = 0, 1, 2,..., *h*), which possesses the orthogonality property, *j*, *q*, – nonnegative integer numbers, *k* = 1, 2,...,*N* – current discrete time or the ordinal number of an element in the sampling.

Equation (1) can be realized by the elementary scheme shown at the Fig. 1 and called us the ortho-synapse [22].

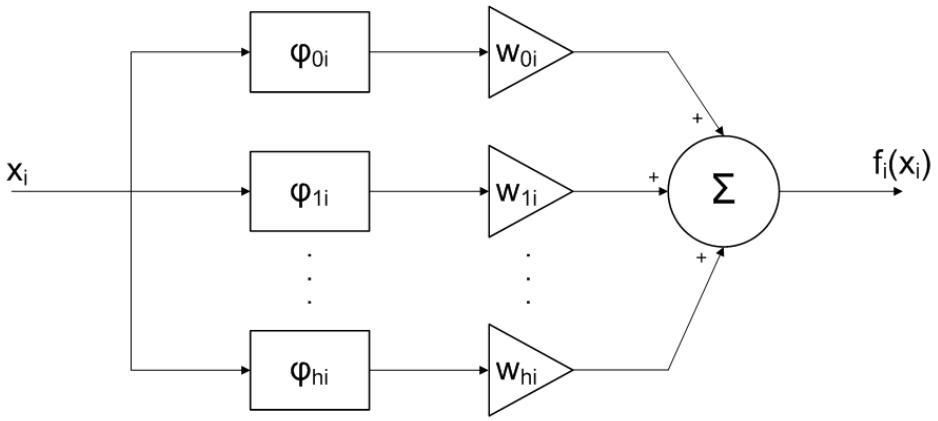

Figure 1. The ortho-synapse – OS_i

At the Fig. 1 x_i is the *i*-th (*i* = 1, 2,..., *n*) component of the multidimensional input signal $x = (x_1, x_2, ..., x_n)^T$, w_{ji} (*j* = 1, 2,...,*h*) – synaptic weights which should be determined, $f_i(x_i)$ – output signal of the ortho-synapse, which can be expressed in the form

$$f_i(x_i) = \sum_{j=0}^{h} w_{ji}\varphi_{ji}(x_i). \quad (2)$$

Different systems of orthogonal polynomials (Chebyshev, Hermite, Laguerre, Legendre, etc) can be used as the activation functions of ortho-synapse. Particular system of functions can be chosen accordingly to the specificity of the solved problem. If the input data is normalized on the hypercube $[-1, 1]^n$, the system of Legendre polynomials orthogonal on the interval [-1, 1] with weight $\gamma(x) = 1$ can be used:

$$\varphi_{ji}^{L}(x_i) = 2^{-j}\sum_{p=0}^{[j/2]}(-1)^p \frac{(2j-2p)!}{p!(j-p)!(j-2p)!}x_i^{j-2p}, \quad (3)$$

where $[\bullet]$ – is the integer part of a number.

Also to simplify calculations we can exploit recurrent formula

$$\varphi_{j+1,i}^{L}(x_i) = \frac{2j+1}{j+1}x_i P_j(x_i) - \frac{j}{j+1}P_{j-1}(x_i). \quad (4)$$

System of Legendre polynomials is the best suited for the case when we exactly know interval of data changes before network construction. This is quite common situation as well as an opposite one. For the other case the following system of Hermite orthogonal polynomials can be used:

$$H_l(u) = l! \sum_{p=1}^{[l/2]} (-1^p) \frac{(2u)^{l-2p}}{p!(l-2p)!}. \quad (5)$$

This system is orthogonal on $(-\infty,+\infty)$ with weight function $h(u) = e^{-u^2}$ and gives us a possibility to decrease influence of the data lying far from origin.

Also it can be readily seen that ortho-synapse has the same architecture like a nonlinear synapse of the neo-fuzzy-neuron [23 - 25], but provides smooth polynomial approximation, based on orthogonal polynomials, instead of piecewise-linear approximation.

We use ortho-synapse as a structural block for the architecture called us ortho-neuron and shown at the Fig. 2.

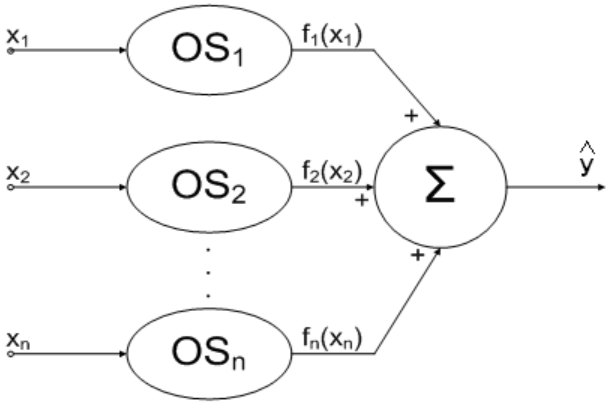

Figure 2. Ortho-neuron – ON

Ortho-neuron which has the same architecture like a neo-fuzzy-neuron realizes the mapping

$$\hat{y} = \sum_{i-1}^{n} f_i(x_i) = \sum_{i=1}^{n} \sum_{j=0}^{h} w_{ji} \varphi_{ji}(x_i), \quad (6)$$

and provides high precision of approximation and extrapolation of significantly nonlinear nonstationary signals and processes [16, 17, 19 - 22]. But in what follows ortho-neuron will be used as the elementary node of the architecture called us the Cascade Orthogonal Neural Network (CONN).

The Cascade Orthogonal Neural Network Architecture

The CONN architecture is shown at the Fig. 3 and mapping, which it realizes, have the following form:

- first cascade neuron :

$$\hat{y}_1 == \sum_{i=1}^{n} \sum_{j=0}^{h} w_{ji}^{[1]} \varphi_{ji}(x_i); \quad (7)$$

- second cascade neuron

$$\hat{y}_2 == \sum_{i=1}^{n} \sum_{j=0}^{h} w_{ji}^{[2]} \varphi_{ji}(x_i) + \sum_{j=0}^{h} w_{j,n+1}^{[2]} \varphi_{j,n+1}(\hat{y}_1); \quad (8)$$

- third cascade neuron

$$\hat{y}_2 == \sum_{i=1}^{n}\sum_{j=0}^{h} w_{ji}^{[3]}\varphi_{ji}(x_i) + \sum_{j=0}^{h} w_{j,n+1}^{[3]}\varphi_{j,n+1}(\hat{y}_1) + \sum_{j=0}^{h} w_{j,n+2}^{[3]}\varphi_{j,n+2}(\hat{y}_2); \quad (9)$$

- *m*-th cascade neuron

$$\hat{y}_m == \sum_{i=1}^{n}\sum_{j=0}^{h} w_{ji}^{[m]}\varphi_{ji}(x_i) + \sum_{l=n+1}^{n+m-1}\sum_{j=0}^{h} w_{jl}^{[m]}\varphi_{jl}^{[m]}(\hat{y}_{l-n}). \quad (10)$$

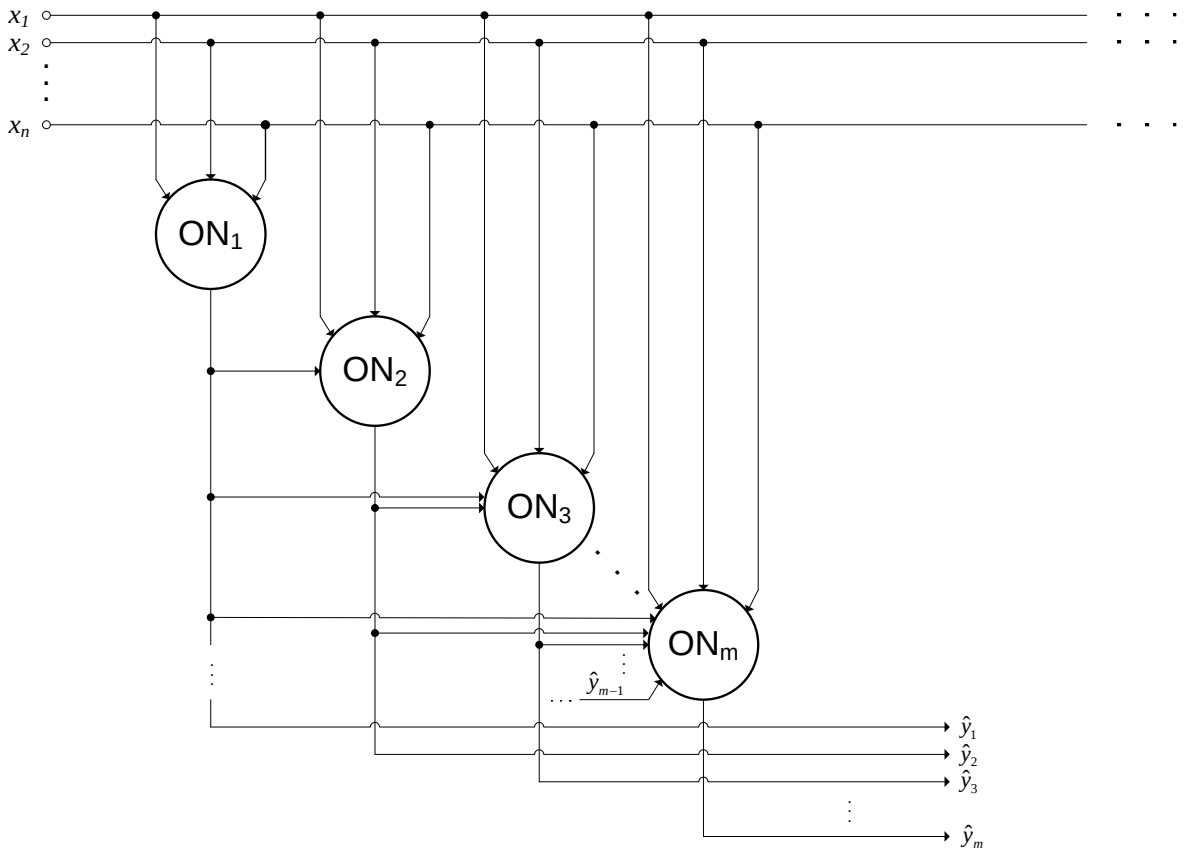

Figure 3. The Cascade Orthogonal Neural Network

Thus the CONN contains $(h+1)(n+\sum_{l=n+1}^{n+m-1} l)$ adjustable parameters and it is important that all of them are linearly included in the definition (10).

Let us define vector $(h+1)(n+m-1)\times 1$ of orthogonal polynomials of the *m*-th ortho-neuron $\varphi^{[m]} = (\varphi_{01}(x_1), \varphi_{11}(x_1),..., \varphi_{h1}(x_1), \varphi_{02}(x_2),..., \varphi_{h2}(x_2),..., \varphi_{ji}(x_i),..., \varphi_{hn}(x_n), \varphi_{0,n+1}(\hat{y}_1),..., \varphi_{h,n+1}(\hat{y}_1),..., \varphi_{h,n+m-1}(\hat{y}_{m-1}))^T$ and corresponding vector of synaptic weights $w^{[m]} = (w_{01}^{[m]}, w_{11}^{[m]},..., w_{h1}^{[m]}, w_{02}^{[m]},..., w_{h2}^{[m]},..., w_{ji}^{[m]},..., w_{hn}^{[m]}, w_{0,n+1}^{[m]},..., w_{h,n+1}^{[m]},..., w_{h,n+m-1}^{[m]})^T$ which has the same dimensionality. Then we can represent expression (10) in the vector notation:

$$\hat{y}_m = w^{[m]T}\varphi^{[m]}. \quad (11)$$

The Cascade Orthogonal Neural Network Learning

The Cascade Orthogonal Neural Network learning is performed in the batch mode using entire training set $x(1), y(1); x(2), y(2); ...; x(k), y(k); ...; x(N), y(N)$. At the beginning a set of orthogonal functions values $\varphi^{[1]}(1), \varphi^{[1]}(2), ..., \varphi^{[1]}(N)$ is calculated for each training sample, so we obtain a sequence of vectors $(h+1)n \times 1$. Then using direct minimization of the learning criterion

$$E_N^{[1]} = \frac{1}{2}\sum_{k=1}^{N} e_1(k)^2 = \frac{1}{2}\sum_{k=1}^{N} (y(k) - \hat{y}_1(k))^2, \tag{12}$$

vector of synaptic weights can be evaluated

$$w^{[1]}(N) = \left(\sum_{k=1}^{N} \varphi^{[1]}(k) \varphi^{[1]T}(k) \right)^{+} \sum_{k=1}^{N} \varphi^{[1]}(k) y(k) = P^{[1]}(N) \sum_{k=1}^{N} \varphi^{[1]}(k) y(k). \tag{13}$$

If dimension of this vector is sufficiently large it is suitable to use procedure (13) in the form of recursive least squares method with sequential training samples processing:

$$\begin{cases} w^{[1]}(k+1) = w^{[1]}(k) + \dfrac{P^{[1]}(k)(y(k+1) - w^{[1]T}(k)\varphi^{[1]}(k+1))}{1 + \varphi^{[1]T}(k+1) P^{[1]}(k) \varphi^{[1]}(k+1)} \varphi^{[1]}(k+1), \\ P^{[1]}(k+1) = P^{[1]}(k) - \dfrac{P^{[1]}(k) \varphi^{[1]}(k+1) \varphi^{[1]T}(k+1) P^{[1]}(k)}{1 + \varphi^{[1]T}(k+1) P^{[1]}(k) \varphi^{[1]}(k+1)} \end{cases}. \tag{14}$$

It is necessary to notice that using procedures (13), (14) for adjusting weight coefficients essentially reduces learning time in comparison with gradient algorithms underlying delta-rule. Also orthogonality of activation functions ensures numerical stability during matrixes inversion.

After first cascade learning completion, synaptic weights of the neuron ON_1 become 'frozen' and second cascade of network consisting from a single neuron ON_2 is generated. It has one additional input for the output signal of the first cascade. Then procedures (13), (14) again applied for adjusting vector of weight coefficients $w^{[2]}$, which dimensionality is $(h+1)(n+1) \times 1$.

The neural network growing process (increasing quantity of cascades) continues until we obtain required precision of the solved problem's solution, and for the adjusting weight coefficients of the last (*m*-th) cascade following expression are used:

$$w^{[m]}(N) = \left(\sum_{k=1}^{N} \varphi^{[m]}(k) \varphi^{[m]T}(k) \right)^{+} \sum_{k=1}^{N} \varphi^{[m]}(k) y(k) = P^{[m]}(N) \sum_{k=1}^{N} \varphi^{[m]}(k) y(k) \tag{15}$$

or

$$\begin{cases} w^{[m]}(k+1) = w^{[m]}(k) + \dfrac{P^{[m]}(k)(y(k+1) - w^{[m]T}(k)\varphi^{[m]}(k+1))}{1 + \varphi^{[m]T}(k+1) P^{[m]}(k) \varphi^{[m]}(k+1)} \varphi^{[m]}(k+1), \\ P^{[m]}(k+1) = P^{[m]}(k) - \dfrac{P^{[m]}(k) \varphi^{[m]}(k+1) \varphi^{[m]T}(k+1) P^{[m]}(k)}{1 + \varphi^{[m]T}(k+1) P^{[m]}(k) \varphi^{[m]}(k+1)} \end{cases}, \tag{16}$$

where vectors $w^{[m]}$ and $\varphi^{[m]}$ have dimensionality $(h+1)(n+m-1) \times 1$.

The main disadvantage of CasCorLA is the necessity of the batch mode learning usage, when all training set should be given priori. CONN can be trained in on-line mode, because of algorithm (16) possesses maximal possible squared rate of convergence. In this case at the first step architecture consisting of *m* cascades is generated. Each cascade trains using proper algorithm. Since outputs of the previous ortho-neurons become additional inputs for the *m*-th cascade, algorithm (16) realizes recurrent method of the prediction error [26], well known in the theory of adaptive identification. Changing cascades quantity during learning process also can be easily performed.

Simulation Results

We have applied proposed Cascade Orthogonal Neural Network to solve 'breast cancer in Wisconsin' benchmark classification problem.

Dataset containing 699 points have been used for this purpose (ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/cancer/cancer1/datacum). 16 points had parameters with missed values so they have been eliminated from the dataset and remaining 683 points have been separated on training set – 478 points (70%) and test set – 205 points (30%).

Each point has 9-dimensional feature vector and 1 class parameter which should be determined and identifies either benign or malignant tumor have current examined patient. Features values have been normalized on interval [-1; 1].

There are 3 optional parameters must be specified to start CONN constructive algorithm: 1) type of the orthogonal polynomials system in each ortho-synapse; 2) quantity of orthogonal polynomials in each ortho-synapse; 3) maximal number of cascades. Since input data have been normalized on interval [-1; 1], we choose systems of 1 type Chebyshev orthogonal polynomials as activation functions for each ortho-synapse to avoid unlimited weight values growth. Previous experiments have shown that optimal ortho-synapse dimensionality is 3-4 polynomials per input, so these values have been chosen for experiment. For avoiding generalization loss maximal number of cascades has been limited by 10.

For comparison the same classification problem has been solved using the Cascade-Correlation Learning Architecture and Multilayered Perceptron. The CasCorLA had 8 cascades and each of them utilized gradient minimization for adjusting weight coefficients. MLP had 9x15x1 architecture and tuned with Levenberg-Marquard minimization procedure during only 20 epochs. Increasing number of epochs (in this case) results in generalization loss. Obtained results of classifications can be found in table 1.

When output signal be found within the range [0.3; 0.7] it is lesser probability that classification were correct. We quantify and marked out such classified samples as points outside the 'belief zone'.

Table 1 – Classification results for different architectures

ANN Architecture	Accuracy on training set / Points outside the 'belief zone'	Accuracy on testing set / Points outside the 'belief zone'
CONN	99,8% / 1	98% / 4
CasCorLA	95% / 46	99% / 15
MLP	99,2% / 4	98,5% / 3

We can see that CONN shows quite good results of classification, comparable with MLP's, and much better than CasCorLA's ones. Therewith CONN's learning procedure takes considerably lesser time and computational recources than backpropogation or Levengerg-Marquardt minimization Also using CONN architecture makes possible to avoid two significant disadvantages of CasCorLA and MLP: unrepeatability of obtained results and necessity to use first- or second-order derivative procedures.

Conclusion

The Cascade Orthogonal Neural Network is proposed. It differs from its prototype, Cascade-Correlation Learning Architecture, in increased speed of operation, numerical stability and real-time processing possibility. Theoretical justification and experiment results confirm the efficiency of developed approach.

Bibliography

[1] Cichocki A., Unbehanen R. Neural Networks for Optimization and Signal Processing. Stuttgart, Teubner, 1993.

[2] Haykin S. Neural Networks. A Comprehensive Foundation. Upper Saddle River, N.J.: Prentice Hall, Inc., 1999.

[3] Platt J. A resource allocating network for function interpolation. Neural Computation, 3, 1991. P.213-225.

[4] Nag A., Ghosh J. Flexible resource allocating network for noisy data. In: Proc. SPIE Conf. on Applications and Science of Computational Intelligence, SPIE Proc. 1998. P.551-559.

[5] Yingwei L., Sundararajan N., Saratchandran P. Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. IEEE Trans. on Neural Networks, 9, 1998. P.308-318.

[6] Cun Y.L., Denker J.S., Solla S.A. Optimal Brain Damage. In: Advances in Neural Information Processing Systems, 2, 1990. P.598-605.

[7] Hassibi B. Stork D.G. Second-order derivatives for network pruning: Optimal brain surgeon. In: Advances in Neural Information Processing Systems. Ed. Hanson et al. 1993. P.164-171.

[8] Prechelt L. Connection pruning with static and adaptive pruning schedules. Neurocomputing, 16, 1997. P.49-61.

[9] Fahlman S.E., Lebiere C. The cascade-correlation learning architecture. Advances in Neural Information Processing Systems. Ed. D.S. Touretzky. San Mateo, CA: Morgan Kaufman, 1990. P.524-532.

[10] Schalkoff R.J. Artificial Neural Networks. N.Y.: The McGraw-Hill Comp., Inc., 1997..

[11] Avedjan E.D., Barkan G.V., Levin I.K. Cascade Neural Networks. Avtomatika i Telemekhanika, 3, 1999. P.38-55.

[12] Bateman H., Erdelyi A. Higher Transcendental Functions. Vol.2. N.Y.: McGraw-Hill Comp., Inc., 1953.

[13] Graupe D. Identification of Systems. Huntington, N.Y.: Robert E. Kreiger Publishing Comp., Inc., 1976.

[14] Scott I., Mulgrew B. Orthonormal function neural network for nonlinear system modeling. In: Proceedings of the International Conference on Neural Networks (ICNN-96), June 1996.

[15] Patra J.C., Kot A.C. Nonlinear dynamic system identification using Chebyshev functional link artificial neural network. IEEE Trans. on System, Man and Cybernetics. Part B, 32, 2002. P.505-511.

[16] Bodyanskiy Ye., Kolodyazhniy V., Slipchenko O. Artificial neural network with orthogonal activation functions for dynamic system identification. Synergies between Information Processing and Automation. Ed. O. Sawodny and P. Scharff – Aachen: Shaker Verlag, 2004. P.24-30.

[17] Bodyanskiy Ye., Kolodyazhniy V., Slipchenko O. Structural and synaptic adaptation in the artificial neural networks with orthogonal activation functions. Sci. Proc. of Riga Technical University. Comp. Sci., Inf. Technology and Management Sci, 20, 2004. P.69-76.

[18] Liying M., Khorasani K. Constructive feedforward neural network using Hermite polynomial activation functions. IEEE Trans. on Neural Networks, 4, 2005. P.821-833.

[19] Bodyanskiy Ye., Pliss I., Slipchenko O. Growing neural networks based on orthogonal activation functions. Proc. XII-th Int. Conf. "Knowledge – Dialog – Solution". Varna, 2006. P84-89.

[20] Bodyanskiy Ye., Slipchenko O. Ontogenic neural networks using orthogonal activation functions. Prace naukowe Akademii Ekonomicznej we Wroclawiu, 21, 2006. P.13-20.

[21] Bodyanskiy Ye., Pliss I., Slipchenko O. Growing neural network using nonconventional activation functions. Int. J. Information Theories & Applications, 14, 2007. P.275-281.

[22] Bodyanskiy Ye., Viktorov Ye., Slipchenko O. Orthosynapse, ortho-neurons, and neuropredictor based on them. Systemi obrobki informacii. Issue 4(62), 2007. P.139-143.

[23] 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. 2-nd Int.Conf. on Fuzzy Logic and Neural Networks "LIZUKA-92". Lizuka, Japan, 1992. P.477-483.

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

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

[26] Ljung L. System Identification: Theory for the User. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1987.

Authors' Information

Yevgeniy Bodyanskiy – Doctor of Technical Sciences, Professor of Artificial Intelligence Department and Scientific Head of the Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Lenina av. 14, Kharkiv, 61166, Ukraine, Tel +380577021890, e-mail: bodya@kture.kharkov.ua

Artem Dolotov – Ph.D. Student, Kharkiv National University of Radio Electronics, Lenin Av., 14, Kharkiv, 61166, Ukraine, Tel +380508361789, e-mail: artem.dolotov@gmail.com

Iryna Pliss – Candidate of Technical Sciences (equivalent Ph.D.), Senior Researcher, Leading Researcher of the Control Systems Research Laboratory, Kharkiv National University of Radio Electronic, Lenina av. 14, Kharkiv, 61166, Ukraine, Tel +380577021890, e-mail: pliss@kture.kharkov.ua

Yevgen Viktorov - Ph.D. Student, Kharkiv National University of Radio Electronics, Lenin Av., 14, Kharkiv, 61166, Ukraine, Tel +380681613429, e-mail: yevgen.viktorov@gmail.com