

## THE EVOLVING ADAPTIVE NEURAL NETWORK FOR DATA PROCESSING WITH MISSING OBSERVATIONS

The problem of computational intelligence systems synthesis in on-line mode, capable for processing stochastic signals with missing observations in the data is considered. An adaptive approach based on using of orthogonal polynomials is developed.

**Keywords:** neural network, orthogonal polynomials, Chebyshev polynomials, incomplete data with missing observations.

### INTRODUCTION

Artificial neural networks, neuro-fuzzy systems, hybrid systems of computational intelligence currently are widespread for solving data processing problems, Data Mining, prediction, identification and control of nonlinear stochastic and chaotic objects and systems [1–5].

The most attractive properties of these systems are their universal approximation properties and learning ability, usually understood as a possibility of tuning the parameters by optimizing some quality criterion (objective function, learning criterion). In a wider sense, it is possible to configure the system architecture too. Currently, there exist number of approaches, but the most widely used is so-called constructive approach in which the system of computational intelligence, starting with the simplest architecture, gradually increasing its complexity and at the same time tuning their parameters to achieve the desired quality of the solution. This approach has formed a new direction in computational intelligence, known as evolving systems [6, 7]. At the same time, most of these systems process information in a batch mode, which makes them difficult to use in cases where the data for processing are fed in on-line mode in the form of time series.

In many practical applications involving the processing of sequences of real data, there exists situation when some observations of the controlled time series, for whatever reasons, are missing (lost). It is understood that for the normal operation of the neural network or a hybrid system, these missing data must somehow be restored. The problem of restoring the missing values has received sufficient attention [8–10], in this case as the most effective are neural networks [11–14]. However, the known approaches for restoration of missing values in time series are effective only in cases when the whole data set is given a priori, the amount of missing values is known, and the time series has a fixed number of observations. It is natural that in problems where data are fed for processing in real time, the number of missing values is unknown beforehand and the sequence has nonstationary character, the known approaches are ineffective.

So, the problem of synthesis of computational intelligence systems in on-line mode, capable for processing stochastic signals with missing observation in the data is interesting and useful.

### SEQUENTIAL RESTORATION OF MISSING OBSERVATIONS IN THE TIME SERIES

The proposed approach is based on the use of classical orthogonal polynomials [15] and first of all Chebyshev polynomials (T-systems) [16, 17], that have a number of good properties in terms of the approximation problem using quadratic criterion [16].

Generally Chebyshev polynomials can be compactly represented using trigonometric functions

$$T_l(x) = \cos(l \arccos x), \quad x \in [-1, 1], \quad l = 0, 1, 2, \dots \quad (1)$$

or using recurrent relation

$$T_l(x) = 2xT_{l-1}(x) - T_{l-2}(x). \quad (2)$$

In some situations it is easier to solve the problem in the interval  $x \in [0, 1]$ , which can be used with biased Chebyshev polynomials:

$$T_l^B(x) = T_l(2x - 1), \quad (3)$$

$$T_l^B(x) = (4x - 2)T_{l-1}^B(x) - T_{l-2}^B(x). \quad (4)$$

It is also easy to write biased polynomials for any interval  $x \in [x_{\min}, x_{\max}]$ .

To approximate arbitrary function of a scalar argument  $y = f(x)$  given on a set of nodes  $x_1, x_2, \dots, x_N; y_1, y_2, \dots, y_N$ , let us transform the original data using simple relations

$$\tilde{x}_k = 2 \frac{x_k - x_{\min}}{x_{\max} - x_{\min}} - 1;$$

$$\tilde{y}_k = 2 \frac{y_k - y_{\min}}{y_{\max} - y_{\min}} - 1, \quad k = 1, 2, \dots, N$$

for the polynomials (1), (2) and

$$\tilde{x}_k = \frac{x_k - x_{\min}}{x_{\max} - x_{\min}}; \quad \tilde{y}_k = \frac{y_k - y_{\min}}{y_{\max} - y_{\min}}, \quad k = 1, 2, \dots, N$$

for (3), (4) and write approximating T-system as

$$T_h^\Sigma(\tilde{x}) = \sum_{l=0}^h w_l T_l(\tilde{x}),$$

whose parameters  $w_l$  are determined by the simple relation

$$w_l = \frac{\sum_{k=1}^N \tilde{y}_k T_l(\tilde{x}_k)}{\sum_{k=1}^N T_l^2(\tilde{x}_k)},$$

with conditions of orthogonality.

It is easy to see that these expressions are standard estimates of least squares. The values  $T_l(\tilde{x}_k)$  are calculated using the recurrent relations

$$T_0(\tilde{x}_k) = \tilde{x}_k - b_l,$$

$$T_l(\tilde{x}_k) = (\tilde{x}_k - b_l)T_{l-1}(\tilde{x}_k) - a_l T_{l-2}(\tilde{x}_k),$$

$$a_l = \frac{\sum_{k=1}^N \tilde{x}_k^{l-1} T_{l-1}(\tilde{x}_k)}{\sum_{k=1}^N \tilde{x}_k^{l-2} T_{l-2}(\tilde{x}_k)},$$

$$b_l = \frac{\sum_{k=1}^N \tilde{x}_k^l T_{l-1}(\tilde{x}_k)}{\sum_{k=1}^N \tilde{x}_k^{l-1} T_{l-1}(\tilde{x}_k)} - \frac{\sum_{k=1}^N \tilde{x}_k^{l-1} T_{l-2}(\tilde{x}_k)}{\sum_{k=1}^N \tilde{x}_k^{l-2} T_{l-2}(\tilde{x}_k)},$$

and achievable accuracy of approximation – based on the variance

$$\sigma_h^2 = \sigma_{h-1}^2 - w_h^2 \sum_{k=1}^N \tilde{x}_k^h T_h(\tilde{x}_k),$$

$$\sigma_0^2 = \sum_{k=1}^N \tilde{y}_k^2 - \frac{1}{N} \left( \sum_{k=1}^N \tilde{y}_k \right)^2.$$

Next, consider the time series  $\tilde{x}_k$ ,  $k = 1, 2, \dots, N$  and introduce approximating T-system

$$T_h^\Sigma(k) = \sum_{l=0}^h w_l T_l(k), \quad (5)$$

but in this case, we assume that at some moments in the discrete time  $k$  measurements or have not been done or lost.

Let's introduce two subsets:  $X_P = \{\tilde{x}_k, k\}$ , that contains  $N_P \leq N$  available observations and  $X_G = \{k\}$ , contains moments with missing observations.

Then the coefficients of the approximating polynomial can be calculated using the following relations:

$$w_l = \frac{\sum_{\substack{k=1 \\ k \in X_P}}^N \tilde{x}_k T_{lN}(k)}{\sum_{\substack{k=1 \\ k \in X_P}}^N T_{lN}^2(k)},$$

$$w_0 = \frac{1}{N_P} \sum_{\substack{k=1 \\ k \in X_P}}^N \tilde{x}_k,$$

where index  $N$  in  $T_{lN}$  means that the corresponding polynomials are orthogonal on the interval  $k \in [1, N]$ .

After this it is easy to restore the missing observations in the form

$$\hat{x}_k = \sum_{l=0}^h w_l T_{lN}(k) \quad \forall k \in X_G.$$

Combining the available ( $\tilde{x}_k \in X_P$ ) and restored ( $\hat{x}_k \in X_G$ ) values of the time series, it's possible to form a sequence containing all  $N$  values, referred  $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N$ .

Next, let's introduce the parameter vector  $w = (w_0, w_1, \dots, w_h)^T$  and T-system of orthogonal polynomials  $T_N(k) = (T_{0N}(k), T_{1N}(k), \dots, T_{hN}(k))^T$  and rewrite (5) in vector form

$$T_h^\Sigma(k) = w^T T_N(k),$$

where vector  $w$  is calculated by the standard method of least squares

$$\begin{aligned} w(N) &= \left( \sum_{\substack{k=1 \\ k \in X_P}}^N T_N(k) T_N^T(k) \right)^{-1} \sum_{\substack{k=1 \\ k \in X_P}}^N T_N(k) \tilde{x}_k = \\ &= P(N) \sum_{\substack{k=1 \\ k \in X_P}}^N T_N(k) \tilde{x}_k. \end{aligned} \quad (6)$$

After that we can finally write

$$\begin{cases} \hat{x}_k = w^T(N) T_N(k) \forall k \in X_G, \\ \hat{x}_k = \tilde{x}_k \forall k \in X_P. \end{cases} \quad (7)$$

Using relations (6) and (7) processing is realized in batch mode for a fixed number of points  $N$ . If the data are fed to the processing sequentially we have to organize on-line data processing. In [17] for this purpose it was proposed to use recurrent least-squares method in the form

$$\begin{cases} w(N) = w(N-1) + \frac{P(N-1)(\tilde{x}_N - w^T(N-1)T_N(N))}{1 + T_N^T(N)P(N-1)T_N(N)} T_N(N), \\ P(N) = P(N-1) - \frac{P(N-1)T_N(N)T_N^T(N)P(N-1)}{1 + T_N^T(N)P(N-1)T_N(N)}, \end{cases} \quad (8)$$

but it should be noted that with the arrival of the new  $(N+1)$ -th observation  $\tilde{x}_{N+1}$ , the structure of the approximating polynomials essentially changes, so that  $T_N(k) \neq T_{N+1}(k)$ . It is clear that this fact significantly complicates the realization of on-line processing. To keep the structure of the approximating polynomials, we can organize data processing on a sliding window  $s$ , with at each moment  $k$ , we will use the computer time  $\tilde{k} = 1, 2, \dots, s$ , connected with real time so that the calculations are provided only in moments  $k-s+1, k-s+2, \dots, k$ ,

$$\begin{cases} T_{0s}(\tilde{k}) = \frac{1}{\sqrt{s}}, \\ T_{1s}(\tilde{k}) = \sqrt{\frac{3}{s(s^2-1)}}(2\tilde{k}-s-1), \\ T_{ls}(\tilde{k}) = (a_l\tilde{k} + b_l)T_{l-1,s}(\tilde{k}) - c_lT_{l-2,s}(\tilde{k}), l = 2, 3, \dots, h, \\ a_l = \frac{2}{l} \sqrt{\frac{d_l(d_l-2)}{g_l}}, \\ b_l = -\frac{s+1}{l} \sqrt{\frac{d_l(d_l-2)}{g_l}} = -\frac{s+1}{2} a_l, \\ c_l = \frac{l-1}{l} \sqrt{\frac{d_l(d_l+g_l-2)}{(d_l-4)g_l}}, \\ d_l = 2l+1, \\ g_l = s^2 - l^2. \end{cases} \quad (9)$$

Then the estimate of type (6) with the fixed structure of the polynomials  $T_s(N)$  can be written as

$$\begin{aligned} w_s(N) &= \left( \sum_{\substack{\tilde{k}=1, \\ k=N-s+1 \\ k \in X_p}}^{s,N} T_s(\tilde{k})T_s^T(\tilde{k}) \right)^{-1} \sum_{\substack{\tilde{k}=1, \\ k=N-s+1 \\ k \in X_p}}^{s,N} T_s(\tilde{k})\tilde{x}_k = \\ &= P_s(N) \sum_{\substack{\tilde{k}=1, \\ k=N-s+1 \\ k \in X_p}}^{s,N} T_s(\tilde{k})T_s^T(\tilde{k}) \end{aligned} \quad (10)$$

and estimates of missing observations

$$\hat{x}_k = \hat{x}_{\tilde{k}} = w_s^T(N)T_s(\tilde{k}) \quad \forall k \in X_G.$$

Implementation of the described approach is very convenient with the apparatus of artificial neural networks using ortho-synapse [18], shown in Fig. 1 and trained by the algorithm (10). Ortho-synapse on the structure coincides with the nonlinear synapse of neo-fuzzy neuron [19, 20], but instead of the membership functions it contains the orthogonal activation functions  $T_{ls}$ , making the learning process easier and quicker. It is also important that due to using of a sliding window, these activation functions are not changed in the learning process. The value of the sliding window is selected from empirical considerations  $s \geq h+1$ . Thus if approximating sequence is nonstationary, the value should not significantly exceed the number of parameter estimates.

### ARCHITECTURE OF ORTHOGONAL NEURAL NETWORK

Using orthogonal polynomials as activation functions leads to the creation of a whole group of orthogonal neural networks [21–29] possessing good approximating properties and high speed learning of synaptic weights. In [30–33] growing orthogonal networks based on the ortho-synapses and ortho-neurons have been proposed [18]. These networks are characterized by simplicity of learning of synaptic weights and architecture.

Fig. 2 shows the architecture of orthogonal neural network for data processing with the lost observations that implements a nonlinear mapping

$$\hat{y}_k = \hat{f}(\hat{x}_k) = \sum_{i=1}^n \sum_{l=0}^h w_{li} T_{li}(\hat{x}_{ki}), \quad (11)$$

where  $k = 1, 2, \dots$  – current discrete time,  $w_{li}$  – tuned synaptic weights,  $T_{li}(\bullet)$  –  $l$ -th orthogonal activation function of the type (1) or (3) for the input signal  $\hat{x}_{ki}$ ,  $i = 1, 2, \dots, n$ .

At the zero (receptive) layer of the network series of observations  $\tilde{x}_{ki}$  containing a priori unknown number of missing values are fed. Note also that the external training signal  $\tilde{y}_k$  also can contain gaps.

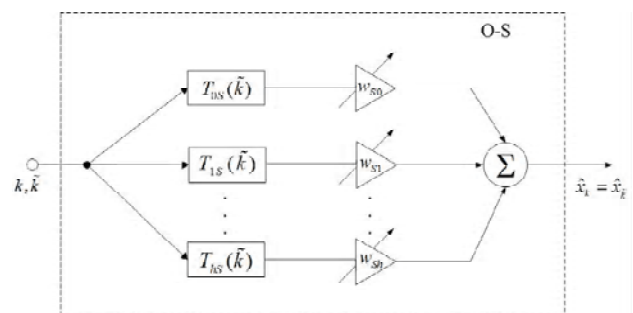
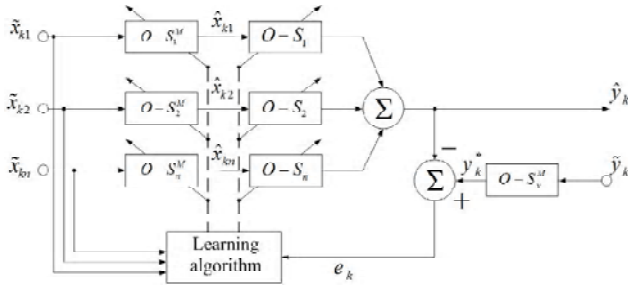


Fig. 1. Ortho-synapse



**Fig. 2.** The orthogonal neural network for data processing with the lost observations

The first hidden layer of the network is formed by  $n$  ortho-synapses  $O-S_i^M$  and serves to restore missing observations.

The same function for the reference training signal  $\tilde{y}_k$  performs ortho synapse  $O-S_y^M$ . As a result, in on-line mode the time series formed us  $\hat{x}_{ki}, y_k^*$ . The output layer is formed by ortho-neuron that coincides with the architecture of neo-fuzzy neuron, but with orthogonal activation functions instead of the usual membership functions. Error

$$e_k = y_k^* - \hat{y}_k$$

is used by learning algorithm to tune both the weights and architecture.

The present architecture contains  $2n+1$  ortho-synapses and  $(h+1)(2n+1)$  synaptic weights to be estimated at the same time it is very important that the output signal  $\hat{y}_k$  depends linearly on the weights.

### SYNAPTIC ADAPTATION

Let's introduce the standard criterion of learning

$$E_k = \sum_{k=1}^N e_k^2 = \sum_{k=1}^N \left( y_k^* - \sum_{i=1}^n \sum_{l=0}^h w_{li} T_{li}(\hat{x}_{ki}) \right)^2,$$

$(h+1)n \times 1$  – vectors of activation functions

$$T(\hat{x}_k) = (T_{01}(\hat{x}_{k1}), T_{11}(\hat{x}_{k1}), \dots, T_{h1}(\hat{x}_{k1}),$$

$$T_{02}(\hat{x}_{k2}), \dots, T_{h2}(\hat{x}_{k2}), \dots, T_{hn}(\hat{x}_{kn}))^T =$$

$$= (T_1^T(\hat{x}_{k1}), T_2^T(\hat{x}_{k2}), \dots, T_n^T(\hat{x}_{kn}))^T$$

and synaptic weights

$$w = (w_{01}, w_{11}, \dots, w_{h1}, w_{02}, w_{12}, \dots, w_{hn})^T = (w_1^T, w_2^T, \dots, w_n^T)^T$$

and rewrite the output network (11) in a compact form

$$\hat{y}_k = w^T T(\hat{x}_k) = \sum_{i=1}^N w_i^T T_i(\hat{x}_k). \quad (12)$$

To tune the vector of synaptic weights  $w$  in a sequential mode it's possible to use a recurrent least squares method (8) processing non-stationary signals is using adaptive procedures that have tracking properties, such as the same method of least squares on the sliding window, which in this situation, can be written as

$$\left\{ \begin{aligned} \tilde{w}(N) &= w(N-1) + \frac{P(N-1)(y_N^* - w^T(N-1)T(\hat{x}_N))}{1 + T^T(\hat{x}_N)P(N-1)T(\hat{x}_N)} T(\hat{x}_N), \\ \tilde{P}(N) &= P(N-1) - \frac{P(N-1)T(\hat{x}_N)T^T(\hat{x}_N)P(N-1)}{1 + T^T(\hat{x}_N)P(N-1)T(\hat{x}_N)}, \\ P(N) &= \tilde{P}(N) + \frac{\tilde{P}(N)T(\hat{x}_{N-s})T^T(\hat{x}_{N-s})\tilde{P}(N)}{1 - T^T(\hat{x}_{N-s})\tilde{P}(N)T(\hat{x}_{N-s})}, \\ w(N) &= \tilde{w}(N) - \frac{\tilde{P}(N)(y_{N-s}^* - \tilde{w}^T(N)T(\hat{x}_{N-s}))}{1 - T^T(\hat{x}_{N-s})\tilde{P}(N)T(\hat{x}_{N-s})} T(\hat{x}_{N-s}) \end{aligned} \right. \quad (13)$$

or

$$\begin{aligned} w(N) &= \left( \sum_{k=N-s+1}^N T(\hat{x}_k)T^T(\hat{x}_k) \right)^{-1} \sum_{k=N-s+1}^N T(\hat{x}_k)y_k^* = \\ &= P_s(N) \sum_{k=N-s+1}^N T(\hat{x}_k)y_k^*, \end{aligned} \quad (14)$$

thus, due to the diagonality of matrix  $P_s(N)$  calculation of estimation (14) even at large  $n$  and  $h$  does not cause difficulty.

### ARCHITECTURAL ADAPTATION

The number of activation functions  $h+1$  in each ortho-synapse  $O-S_i$  of output layer is chosen rather arbitrarily, so if it is found that the synthesized neural network does not provide the required quality of the information processing, the number of these functions can be increased (or decreased, if necessary) in on-line mode directly in the learning process. Thus network evolutionary properties by adapting the structure, due to orthogonality with activation functions for the calculated correction synaptic weights made simply.

Suppose that at the instant  $N = N^*$  it was decided that each ortho-synapse  $O-S_i$  need to add an activation function  $T_{h+1,i}$  so that to the network should be introduced  $n$  additional synaptic weights  $w_{h+1,i}$ .

The estimate  $w(N)$  (14) should be corrected in this situation as follows:

$$w^*(N) = \begin{pmatrix} \sum_{k=N-s+1}^N T(\hat{x}_k)T^T(\hat{x}_k) & | & R_{l,h+1}(\hat{x}_k) \\ \hline R_{l,h+1}^T(\hat{x}_k) & | & R_{h+1,h+1}(\hat{x}_k) \end{pmatrix}^{-1} \times \begin{pmatrix} \sum_{k=N-s+1}^N T(\hat{x}_k)y_k^* \\ \hline r_{h+1}(\hat{x}_k) \end{pmatrix}, \quad (15)$$

where  $((h+1)n \times n)$ -matrix  $R_{l,h+1}(\hat{x}_k)$  formed by elements  $\sum_{k=N-s+1}^N T_{li}(\hat{x}_{ki})T_{h+1,j}(\hat{x}_{kj})$ ,  $l = 0, 1, \dots, h$ ;  $i = 1, 2, \dots, n$ ;  $j = 1, 2, \dots, n$ ;  $(n \times n)$ -matrix  $R_{h+1,h+1}(\hat{x}_k)$  – formed by elements  $\sum_{k=N-s+1}^N T_{h+1,i}(\hat{x}_{ki})T_{h+1,j}(\hat{x}_{kj})$ ;  $(n \times 1)$  – the column  $r_{h+1,i}(\hat{x}_{ki})$  – formed by elements  $\sum_{k=N-s+1}^N T_{h+1,i}(\hat{x}_{ki})y_k^*$ .  
Applying the Frobenius formula, estimate (15) can be rewritten as

$$w^*(N) = \begin{pmatrix} P_s(N) + P_s(N)R_{l,h+1}(\hat{x}_k)H^{-1}(\hat{x}_k)R_{l,h+1}^T(\hat{x}_k)P_s(N) & | & -P_s(N)R_{l,h+1}(\hat{x}_k)H^{-1}(\hat{x}_k) \\ \hline -H^{-1}(\hat{x}_k)R_{l,h+1}(\hat{x}_k)P_s(N) & | & H^{-1}(\hat{x}_k) \end{pmatrix} \times \begin{pmatrix} \sum_{k=N-s+1}^N T(\hat{x}_k)y_k^* \\ \hline r_{h+1}(\hat{x}_k) \end{pmatrix} = \begin{pmatrix} w(N) + P_s(N)R_{l,h+1}(\hat{x}_k)H^{-1}(\hat{x}_k)R_{l,h+1}^T(\hat{x}_k)w(N) - P_s(N)R_{l,h+1}(\hat{x}_k)H^{-1}(\hat{x}_k)r_{h+1}(\hat{x}_k) \\ \hline -H^{-1}(\hat{x}_k)R_{l,h+1}^T(\hat{x}_k)w(N) + H^{-1}(\hat{x}_k)r_{h+1}(\hat{x}_k) \end{pmatrix},$$

where

$$H^{-1}(\hat{x}_k) = R_{h+1,h+1}^{-1}(\hat{x}_k) - R_{l,h+1}^T(\hat{x}_k)P_s(N)R_{l,h+1}(\hat{x}_k).$$

It's obvious that architectural adaptation is more difficult than synaptic. The possibility of its implementation in on-line mode, by using of orthogonal activation functions, can achieve the required approximating properties in the learning process.

### CONCLUSION

The evolving neural network that due to using orthogonal activation functions tunes the synaptic weights and structure in the learning process is proposed. Another important feature of the proposed network is connected with the possibility to process in on-line mode information, spoiled by missing values in the data. The neural network under consideration is characterized by high speed and can to process distorted nonlinear nonstationary stochastic and chaotic signals in real time.

### SPISOK LITERATURY

1. Jang, J.-S. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence / J.-S. Jang, C.-T. Sun, E. Mizutani. – Upper Saddle River : Prentice Hall, 1997. – 640 p.
2. Haykin, S. Neural Networks. A Comprehensive Foundation / S. Haykin. – Upper Saddle River : Prentice Hall, 1999. – 842 p.

3. Nelles, O. Nonlinear System Identification / O. Nelles. – Berlin : Springer, 2001. – 585 p.
4. Rutkowska, D. Neuro-Fuzzy Architectures and Hybrid Learning / D. Rutkowska. – Berlin : Springer, 2002. – 288 p.
5. Rutkowski, L. Computational Intelligence. Methods and Techniques / L. Rutkowski. – Berlin-Heidelberg : Springer-Verlag, 2008. – 514 p.
6. Kasabov, N. Evolving Connectionist Systems / N. Kasabov. – London : Springer-Verlag, 2003. – 307 p.
7. Lughofer, E. Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications / E. Lughofer. – Berlin-Heidelberg : Springer-Verlag, 2011. – 454 p.
8. Загоруйко, Н. Г. Эмпирические предсказания / Н. Г. Загоруйко. – Новосибирск : Наука, 1979. – 120 с.
9. Han, J. Data Mining : Concepts and Techniques / J. Han, M. Kamber. – Amsterdam : Morgan Kaufman Publ., 2006. – 743 p.
10. Gorban, A. Principal Manifolds for Data Visualization and Dimension Reduction / A. Gorban, B. Kegl, B. Wunch, A. Zinovyev (Eds.). – LNCS. – V. 58. – Berlin-Heidelberg. – New York : Springer, 2007. – 330 p.
11. Bishop, C. M. Neural Networks for Pattern Recognition. – Oxford : Clarendon Press, 1995. – 482 p.
12. Gorban, A. N. Neural network modeling of data with gaps / A. N. Gorban, A. A. Rossiev, D. C. Wunch II // Радіоелектроніка, інформатика, управління. – 2000. – No. 1 (3). – С. 47–55.

13. Tkacz, M. Artificial neural networks in incomplete data sets processing / In: Eds. M. A. Kłopotek, S. T. Wierzbach, K. Trojanowski Intelligent Information Processing and Web Mining. – Berlin-Heidelberg: Springer-Verlag, 2005. – P. 577–583.
14. Marwala, T. Computational Intelligence for Missing Data Imputation, Estimation, and Management: Knowledge Optimization Techniques. Hershey-New York : Information Science Reference, 2009. – P. 303.
15. Суєтин, П. К. Классические ортогональные многочлены / П. К. Суєтин. – М. : Наука, 1976. – 328 с.
16. Karlin, S. Tchebycheff Systems with Applications in Analysis and Statistics / S. Karlin, W. J. Studden. – N. Y. – London – Sydney : Interscience Publishes, 1966. – 586 p.
17. Семесенко, М. П. Методи обробки та аналізу измерений в научних дослідженнях. – Київ-Донець : Вища школа, 1983. – 240 с.
18. Бодянский Е. В. Ортоинапс, ортонейроны и нейропредиктор на их основе / Е. В. Бодянский, Е. А. Викторов, А. Н. Слипенченко // Системи обробки інформації. – 2007. – Вип. 4 (62). – С. 139–143.
19. Uchino, E. Soft computing based signal prediction, restoration and filtering / E. Uchino, T. Yamakawa Ed. Da Ruan «Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks and Genetic Algorithms». – Boston : Kluwer Academic Publisher, 1997. – P. 331–349.
20. Miki, T. Analog implementation of neo-fuzzy neuron and its on-board learning / T. Miki, T. Yamakawa Ed. N. E. Mastorakis «Computational Intelligence and Applications». – Piraeus : WSES Press, 1999. – P. 144–149.
21. Yang, S.-S. An orthonormal neural network for function approximation / S.-S. Yang, C.-S. Tseng // IEEE Trans. on Syst., Man, and Cybern. – 1996. – No. 10. – P. 779–784.
22. Lee, T. T. The Chebychev polynomial-based unified model neural networks for functional approximation / T. T. Lee, J. T. Jeng // IEEE Trans. on Syst., Man, and Cybern. – 1998. – Vol. 28, No. 12. – P. 925–935.
23. Andras, P. Orthogonal RBF neural network approximation / P. Andras // Neural Processing Letters. – 1999. – Vol. 9, No. 2. – P. 141–151.
24. Chien F. Sh. Properties and performance of orthogonal neural network in function approximation / F. Sh. Chien, Ch.-Ch. Tseng, Ch.-S. Chen // Int. J. Intelligent Systems. – 2001. – Vol. 16. – P. 1377–1392.
25. Patra, J. C. Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks / J. C. Patra, A. C. Kot // IEEE Trans. on Syst., Man, and Cybern. – 2002. – Vol. 32, No. 4. – P. 505–511.
26. Bodyanskiy, Ye. Artificial neural network with orthogonal activation functions for dynamic system identification / Ye. Bodyanskiy, V. Kolodyazhnyi, O. Slipchenko Eds. by O. Sawodny, P. Scharff «Sinergies between Information Processing and Automation». – Aachen : Shaker-Verlag, 2004. – P. 122–127.
27. Statiak, B. Fast orthogonal neural network / B. Statiak, M. Yatsymirskyy // Artificial Intelligence and Soft Computing. – 2006. – No. 4029. – P. 142–149.
28. Rodriguez, N. Orthogonal neural network based prediction for OFDM systems / N. Rodriguez, C. Cubillos // Proc. Electronics, Robust, and Automotive Mechanics Conf., 2007. – P. 225–228.
29. Hongwei, W. Tracking control of robot manipulators based on orthogonal neural network / W. Hongwei, Yu. Shuanghe // Int. J. Modeling, Identification, and Control. – 2010. – No. 11 (1–2). – P. 130–136.
30. Bodyanskiy, Ye. Structural and synaptic adaptation in the artificial neural networks with orthogonal activation functions / Ye. Bodyanskiy, V. Kolodyazhnyi, O. Slipchenko // Sci. Proc. of Riga Technical University. Comp. Sci., Inf. Technology and Management Sci. – 2004. – No. 20. – P. 69–76.
31. Bodyanskiy, Ye. Ontogenic neural networks using orthogonal activation functions / Ye. Bodyanskiy, O. Slipchenko // Prace naukowe Akademii Ekonomicznej we Wroclawiu. – 2006. – No. 21. – P. 13–20.
32. Bodyanskiy, Ye. Growing neural networks using nonconventional activation functions / Ye. Bodyanskiy, I. Pliss, O. Slipchenko // Int. J. Information Theories & Applications. – 2007. – Vol. 14, No. 3. – P. 275.
33. Bodyanskiy, Ye. The cascade orthogonal neural network / Ye. Bodyanskiy, A. Dolotov, I. Pliss, Ye. Victorov // In «Advanced Research in Artificial Intelligence». – Vol. 2. – Sofia : FOI ITHEA, 2008. – P. 13–20.

Стаття надійшла до редакції 16.10.2013.

Шафроненко А.<sup>1</sup>, Плісс І.<sup>2</sup>, Бодянский Є.<sup>3</sup>

<sup>1</sup>Аспірантка, Харківський національний університет радіоелектроніки, Україна

<sup>2</sup>Канд. техн. наук, старший науковий співробітник, провідний науковий співробітник, Харківський національний університет радіоелектроніки, Україна

<sup>3</sup>Д-р техн. наук, професор, науковий керівник, Харківський національний університет радіоелектроніки, Україна

### ЕВОЛЮЦІОНУЮЧА АДАПТИВНА НЕЙРОННА МЕРЕЖА ДЛЯ ОБРОБКИ ДАНИХ ІЗ ВТРАЧЕНИМИ СПОСТЕРЕЖЕННЯМИ

Розглянуто задачу синтезу еволюціонуючої в on-line режимі системи обчислювального інтелекту, здатної обробляти стохастичні сигнали, що містять пропуски в даних. Запропоновано адаптивний підхід, в основі якого полягає використання ортогональних поліномів.

**Ключові слова:** нейронна мережа, ортогональні поліноми, поліноми Чебишева, неповні дані з втраченими спостереженнями.

Шафроненко А.<sup>1</sup>, Плісс І.<sup>2</sup>, Бодянский Є.<sup>3</sup>

<sup>1</sup>Аспірантка, Харківський національний університет радіоелектроніки, Україна

<sup>2</sup>Канд. техн. наук, старший науковий співробітник, провідний науковий співробітник, Харківський національний університет радіоелектроніки, Україна

<sup>3</sup>Д-р техн. наук, професор, науковий керівник, Харківський національний університет радіоелектроніки, Україна

### ЭВОЛЮЦИОНИРУЮЩАЯ АДАПТИВНАЯ НЕЙРОННАЯ СЕТЬ ДЛЯ ОБРАБОТКИ ДАННЫХ С ПОТЕРЯННЫМИ НАБЛЮДЕНИЯМИ

Рассмотрена задача синтеза эволюционирующей в on-line режиме системы вычислительного интеллекта, способной обрабатывать стохастические сигналы, содержащие пропуски в данных. Предложен адаптивный подход, в основе которого лежит использование ортогональных полиномов.

**Ключевые слова:** нейронная сеть, ортогональные полиномы, полиномы Чебышева, неполные данные с потерянными наблюдениями.

## REFERENCES

- Jang J.-S., Sun C.-T., Mizutani E. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Upper Saddle River: Prentice Hall, 1997, 640 p.
- Haykin S. *Neural Networks. A Comprehensive Foundation*. Upper Saddle River, Prentice Hall, 1999, 842 p.
- Nelles O. *Nonlinear System Identification*, Berlin, Springer, 2001, 585 p.
- Rutkowska D. *Neuro-Fuzzy Architectures and Hybrid Learning*. Berlin, Springer, 2002, 288 p.
- Rutkowski L. *Computational Intelligence. Methods and Techniques*. Berlin-Heidelberg, Springer-Verlag, 2008, 514 p.
- Kasabov N. *Evolving Connectionist Systems*. London, Springer-Verlag, 2003, 307 p.
- Lughofer E. *Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications*. Berlin-Heidelberg, Springer-Verlag, 2011, 454 p.
- Zagoruyko N. G. *Empiricheskiye predskazaniya*. Novosibirsk, Nauka, 1979, 120 p.
- Han J., Kamber M. *Data Mining, Concepts and Techniques*. Amsterdam, Morgan Kaufman Publ., 2006, 743 p.
- Gorban A., Kegl B., Wunch B., Zinovyev A. (Eds.) *Principal Manifolds for Data Visualization and Dimension Reduction*, LNCS, V.58, Berlin-Heidelberg, New York, Springer, 2007, 330 p.
- Bishop C. M. *Neural Networks for Pattern Recognition*. Oxford, Clarendon Press, 1995, 482 p.
- Gorban A. N., Rossiev A. A., Wunch II D. C. Neural network modeling of data with gaps, *Radio Electronics, Computer Science, Control*, 2000, No. 1 (3), pp. 47–55.
- Tkacz M. Artificial neural networks in incomplete data sets processing, In: Eds. Klopotek M. A., Wierzchon S. T., Trojanowski K. *Intelligent Information Processing and Web Mining*. Berlin- Heidelberg, Springer-Verlag, 2005, pp. 577–583.
- Marwala T. *Computational Intelligence for Missing Data Imputation, Estimation, and Management, Knowledge Optimization Techniques*. Hershey-New York, Information Science Reference, 2009, pp. 303.
- Suetin P. K. *Klassicheskiye ortogonal'nye mnogochleny*. Moscow, Nauka, 1976, 328 p.
- Karlin S., Studden W. J. *Tchebycheff Systems with Applications in Analysis and Statistics*. N.Y. London, Sydney, Interscience Publishes, 1966, 586 p.
- Semesenko M. P. *Metody obrabotki i analiza izmereniy v nauchnykh issledovaniyakh*, Kiyev-Donetsk, Vishcha shkola, 1983, 240 p.
- Bodyanskiy Ye. V., Viktorov Ye. A., Slipchenko A. N. *Ortosinaps, ortoneyrony i neyroprediktor na ikh osnove, Sistemi obrobki informatsii*, 2007. Vip.4 (62), pp.139–143.
- Uchino E., Yamakawa T. Soft computing based signal prediction, restoration and filtering. Ed. Da Ruan *«Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks and Genetic Algorithms»*, Boston, Kluwer Academic Publisher, 1997, pp. 331–349.
- Miki T., Yamakowa T. Analog implementation of neo-fuzzy neuron and its on-board learning, Ed. N. E. Mastorakis *«Computational Intelligence and Applications»*. Piraeus, WSES Press, 1999, pp. 144–149.
- Yang S.-S., Tseng C.-S. An orthonormal neural network for function approximation, *IEEE Trans. on Syst., Man, and Cybern*, 1996, No. 10, pp. 779–784.
- Lee T. T., Jeng J. T. The Chebychev polynomial-based unified model neural networks for functional approximation, *IEEE Trans. on Syst., Man, and Cybern*, 1998, vol. 28, No. 12, pp. 925–935.
- Andras P. Orthogonal RBF neural network approximation, *Neural Processing Letters*, 1999, 9(2), pp. 141–151.
- Chien F. Sh., Tseng Ch.-Ch., Chen Ch.-S. Properties and performance of orthogonal neural network in function approximation, *Int. J. Intelligent Systems*, 2001, 16, pp. 1377–1392.
- Patra J. C., Kot A. C. Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks, *IEEE Trans. on Syst., Man, and Cybern*, 2002, vol. 32, No. 4, pp. 505–511.
- Bodyanskiy Ye., Kolodyazhniy V., Slipchenko O. Artificial neural network with orthogonal activation functions for dynamic system identification, Eds. by O. Sawodny, P. Scharff *«Sinergies between Information Processing and Automation»*, Aachen, Shaker-Verlag, 2004, pp. 122–127.
- Statiak B., Yatsymirskyy M. Fast orthogonal neural network, *Artificial Intelligence and Soft Computing*, 2006, 4029, pp. 142–149.
- Rodriguez N., Cubillos C. Orthogonal neural network based prediction for OFDM systems, *Proc. Electronics, Robust, and Automotive Mechanics Conf.*, 2007, pp. 225–228.
- Hongwei W., Shuanghe Yu. Tracking control of robot manipulators based on orthogonal neural network, *Int. J. Modeling, Identification, and Control*, 2010, No. 11 (1–2), pp. 130–136.
- 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*, 2004, No. 20, pp. 69–76.
- Bodyanskiy Ye., Slipchenko O. Ontogenic neural networks using orthogonal activation functions, *Prace naukowe Akademii Ekonomicznej we Wroclawiu*, 2006, No. 21, pp. 13–20.
- Bodyanskiy Ye., Pliss I., Slipchenko O. Growing neural networks using nonconventional activation functions, *Int. J. Information Theories & Applications*, 2007, vol. 14, No. 3, pp. 275.
- Bodyanskiy Ye., Dolotov A., Pliss I., Viktorov Ye. The cascade orthogonal neural network, In *«Advanced Research in Artificial Intelligence»*, Vol. 2, Sofia, FOI ITHEA, 2008, pp. 13–20.