descriptions of classes (patterns), providing mass parallelism at data processing and significant acceleration of processes for solution making in the process of pattern recognition has been described.

Results have been obtained at support of RFBR grant № 06-08-01612-a and Program “Innovations Support” of Presidium of RAS.

Bibliography

[1]. Kosovskaya T.M. Evaluations of Complexity for Solution of Pattern Recognition Problems at Hierarchical Class Description. – Proceedings of XI International conference on problems of theoretical cybernetics. Ulyanovsk, 1996. (In Russian).

[2]. Kosovskaya T.M. Multi-level Descriptions of Classes for Solution Making in – Proceedings of III International conference "Discrete models in theory of control systems". M., Dialogue — MSU, 1998. (In Russian).

[3]. Timofeev A.V. Robots and Artifiial Intelligence. – M., "Nauka", 1978. 191 pp. (In Russian).

[4]. Kosovskaya T.M., Timofeev A.M. About One New Approach to Forming of Logic Resolution Rules – LSU Informer, 1985, №8.P. 22–29. No 1, P. 160–176. (In Russian).

[5]. Yablonskiy S.V. Test – Mathematical Encyclopedy, v.5. M., 1985. P. 342–346. Publication. John Wiley \& Sons, Inc. 2000. 491 pp. (In Russian).

[6]. Timofeev A.M., Kalyaev A.V. Methods for Synthesis and Minimization of Complexity of Cognitive Neuromodules of Super-Macro-Neurocomputers with Programmed Architecture. – Papers of Russian Academy of Sciences, 1994, v. 273, № 2, pp. 180–183 (In Russian and English).

[7]. Timofeev A.V. M Methods for Synthesis of Diophantine Neural Networks of Minimal Complexity. – Papers of Russian Academy of Sciences, 1995, т. 301, № 3, pp. 1106–1109 (In Russian and English).

[8]. Timofeev A.V., Semyonov A.V. Genetic Algorithms of Database Control and Knowledge Base Synthesis and Their Applications. – International Journal of Information Theories & Applications, Sofia, 1996, v.4, N1, pp. 17–22.

[9]. Timofeev A.V., Sheozhev A.M., Shibzukhov Z.M. Multi-Agent Diophantine Neural Networks in Problems of Recognition and Diagnostics. – Neurocomputers: Development and Application, 2005, № 10–11, pp. 69–74. (In Russian).

Authors' Information

Timofeev Adil V. *– Dr.Sc., Professor, Head of the Laboratory of Information Technologies in Control and Robotics of Saint-Petersburg Institute for Informatics and Automation of Russian Academy of Sciences, tav@iias.spb.su*

Kosovskaya Tatiana M. *– Doctor's Degree Student, Laboratory of Information Technologies in Control and Robotics of Saint-Petersburg Institute for Informatics and Automation of Russian Academy of Sciences, Russia;*

Ph.D., Assistant Professor, Department of Mathematics and Mechanics, Saint-Petersburg State University, Russia, e-mail: kosov@nk1022.spb.edu

ADAPTIVE WAVELET-NEURO-FUZZY NETWORK IN THE FORECASTING AND EMULATION TASKS

Yevgeniy Bodyanskiy, Iryna Pliss, Olena Vynokurova

Abstract: *The architecture of adaptive wavelet-neuro-fuzzy-network and its learning algorithm for the solving of nonstationary processes forecasting and emulation tasks are proposed. The learning algorithm is optimal on rate of convergence and allows tuning both the synaptic weights and dilations and translations parameters of wavelet activation functions. The simulation of developed wavelet-neuro-fuzzy network architecture and its learning algorithm justifies the effectiveness of proposed approach.*

Keywords: *wavelet, adaptive wavelet-neuro-fuzzy network, recurrent learning algorithm, forecasting, emulation.*

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

Introduction

At present time the neuro-fuzzy systems have been an increasingly popular technique of soft computing [1-4] successfully applied for the processing of information containing complex nonlinear regularities and distortions of all kinds. These systems combine the linguistic interpretability and the approximation properties of the fuzzy inference systems [5, 6] with the learning and universal approximation capabilities of artificial neural networks [7, 8]. This means, that they can be used in forecasting and emulation of the stochastic and chaotic signals and sequences with complex nonlinear trends and nonstationary parameters, described by the difference nonlinear autoregression equations (NAR) in the form

$$x(k) = F(X(k)) + \xi(k),$$

where $X(k) = (x(k-1), x(k-2), \ldots, x(k-n))^T$ is $(n \times 1)$ the prehistory vector, which determines present state $x(k)$, $x(k)$ is a signal value in k-th instant of discrete time $k = 0,1,2,\ldots,$ $F(\bullet)$ is an arbitrary nonlinear function, unknown in the general case, $\xi(k)$ is a stochastic disturbance with unknown characteristics but with bounded second moment.

Along with the neuro-fuzzy systems for processing the signals of all kinds, the wavelet transform has been an increasingly popular technique [9-11] which provides a compact local signal representation in both time and frequency domain. At the turn of the artificial neural network and wavelets theories the wavelet neural networks have evolved for the analysis of nonstationary processes with considerably nonlinear trends [12-18].

The natural step is to combine the transparency and the interpretability of fuzzy inference systems, powerful approximation and learning capabilities of artificial neural networks and compact description and the flexibility of wavelet transform in the context of hybrid systems of computational intelligence, which further we shall call as the adaptive wavelet-neuro-fuzzy networks (AWNFN).

The key point, defining effectiveness of such systems, is the choice of learning algorithm, which is usually based on the gradient procedures of the accepted criterion minimization. Combination of the gradient optimization with the error backpropagation essentially reduces the rate of learning hybrid systems [19] and leads to necessity of using rather large training samples. In the case when the data processing has to be carried out in real time, forecasted or emulated sequence is nonstationary and distorted, conventional gradient descent learning algorithms (let alone genetic algorithms) appeared to be ineffective.

The paper is devoted to the tasks of synthesis of adaptive wavelet-neuro-fuzzy network for the forecasting and emulation tasks. This network has higher rate of learning in comparison with systems using conventional backpropagation gradient algorithm.

Architecture of the adaptive wavelet-neuro-fuzzy network

Let us introduce into consideration the five-layers architecture, shown on fig. 1, someway similar to the well-known ANFIS [2] which is in turn the learning system of Takagi-Sugeno-Kang fuzzy inference [20,21].

The input layer of the architecture is formed of the time-delay elements z^{-1} ($z^{-1}x(k) = x(k-1)$) and under the input of current signal value $x(k)$ the prehistory vector $X(k) = (x(k-1), x(k-2), \ldots, x(k-n))^T$ is formed as an output of this layer.

The first hidden layer unlike the neuro-fuzzy systems is formed not of conventional non-negative membership functions, but of hn wavelets (h wavelets for each input) $\varphi_{ji}(x(k-i)) = \varphi_{ji}(x(k-i), c_{ji}, \sigma_{ji}) = \varphi_{ji}(k)$ with $2hn$ tuning parameters of dilation (center) c_{ji} and translation (width) σ_{ji}.

Various kinds of analytical wavelets can be used as the activation functions in adaptive wavelet-neuro-fuzzy network, for example: Morlet wavelets, “Mexican hat” wavelets, Polywog wavelets, Rasp wavelets [12], the generator of analytic wavelets [22], the triangular wavelets [23].

Here it can be noticed, that the oscillation character of wavelet function doesn’t contradict the unipolarity of membership functions as negative values φ_{ji} can be interpreted in terms of the small membership or nonmembership levels [24, 25].

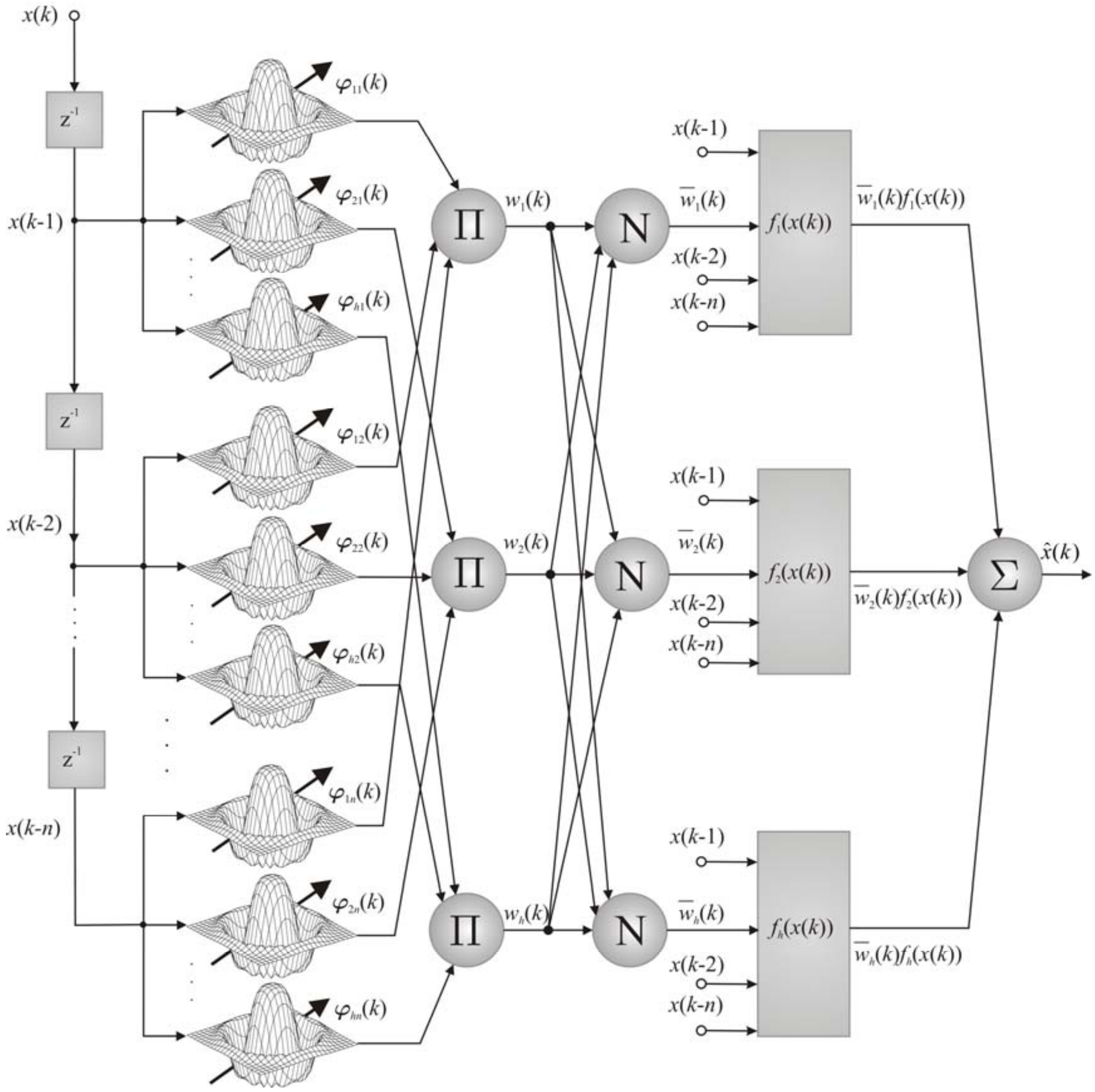

Fig. 1 – Adaptive wavelet-neuro-fuzzy network for the forecasting and emulation tasks

The second hidden layer performs the operation similar to computing of fuzzy T -norm

$$w_j(k) = \prod_{i=1}^{n} \varphi_{ji}(x(k-i)), \quad j = 1, 2, \ldots, h,$$

after that the normalization is performed in the third hidden layer

$$\bar{w}_j(k) = \frac{w_j(k)}{\sum_{j=1}^{h} w_j(k)} = \frac{\prod_{i=1}^{n} \varphi_{ji}(x(k-i))}{\sum_{j=1}^{h} \prod_{i=1}^{n} \varphi_{ji}(x_i(k-i))},$$

providing fulfillment of the condition

$$\sum_{j=1}^{h} \bar{w}_j(k) = 1.$$

The fourth hidden layer performs an operation similar to computing of the consequent in the fuzzy inference systems. The most often used function $f_j(x(k))$ in fuzzy inference systems is linear form (in our case local autoregression model):

$$f_j(X(k)) = p_{j0} + \sum_{i=1}^{n} p_{ji} x(k-i) .$$

In this case in the fourth layer signal values are computed

$$\overline{w}_j(k)(p_{j0} + \sum_{i=1}^{n} p_{ji} x(k-i)) = \overline{w}_j(k) p_j^T \overline{X}(k),$$

where $\overline{X}(k) = (1, X^T(k))^T$, $p_j = (p_{j0}, p_{j1}, \dots, p_{jn})^T$, and $h(n+1)$ parameters p_{ji} , $j = 1,2,\dots,h$, $i = 0,1,2,\dots,n$ are to be determined.

And at last output signal (forecast $\hat{x}(k)$) of network is computed in the fifth output layer

$$\hat{x}(k) = \sum_{j=1}^{h} \overline{w}_j(k) f_j(X(k)) = \sum_{j=1}^{h} \frac{w_j(k)}{\sum_{j=1}^{h} w_j(k)} f_j(X(k)) = \sum_{j=1}^{h} \frac{\prod_{i=1}^{n} \varphi_{ji}(x_i(k-i), c_{ji}, \sigma_{ji})}{\sum_{j=1}^{h} \prod_{i=1}^{n} \varphi_{ji}(x_i(k-i), c_{ji}, \sigma_{ji})} f_j(X(k)),$$

which, introducing the variables vectors $f(X(k)) = (\overline{w}_1(k), \overline{w}_1(k)x(k-1), \dots, \overline{w}_1(k)x(k-n), \overline{w}_2(k), \overline{w}_2(k)x(k-1), \dots, \overline{w}_2(k)x(k-n), \dots, \overline{w}_h(k), \overline{w}_h(k)x(k-1), \dots, \overline{w}_h(k)x(k-n))^T$, $p = (p_{10}, p_{11}, \dots, p_{1n}, p_{20}, p_{21}, \dots, p_{2n}, p_{h0}, p_{h1}, \dots, p_{hn})^T$ of dimensionality $h(n+1)$, can be rewritten in the compact form

$$\hat{x}(k) = p^T f(X(k)) .$$

The tunable parameters of this network are located only in the first and fourth hidden layers. These are $2hn$ wavelets parameters c_{ji} and σ_{ji} , and $h(n+1)$ parameters of the linear local autoregression models p_{ji} . Namely they must be determined during the learning process.

The learning of adaptive wavelet-neuro-fuzzy network

As far as tunable vector of parameters p is contained in the network description linearly, for its refinement any of the algorithms used in adaptive identification [26] will operate, primarily the exponentially weighted recurrent least squares method (this method is the second order optimization procedure and has both filtering and following properties) in the form

$$\begin{cases} p(k+1) = p(k) + \dfrac{P(k)(x(k) - p^T(k) f(X(k)))}{\alpha + f^T(X(k))P(k) f(X(k))} f(X(k)), \\ P(k+1) = \dfrac{1}{\alpha}\left(P(k) - \dfrac{P(k) f(X(k+1)) f^T(X(k+1))P(k)}{\alpha + f^T(X(k+1))P(k) f(X(k+1))} \right) \end{cases} \tag{1}$$

where $x(k) - p^T(k) f(x(k)) = x(k) - \hat{x}(k) = e(k)$ is the forecasting (emulation) error, $0 < \alpha \le 1$ is the out-dated information forgetting factor; optimal on operation rate one-step gradient Kaczmarz algorithm [27, 28], having the following properties

$$p(k+1) = p(k) + \frac{x(k) - p^T(k) f(X(k))}{f^T(X(k)) f(X(k))} f(X(k)) , \tag{2}$$

or Goodwin-Ramadge-Caines algorithm [29]

$$\begin{cases} p(k+1) = p(k) + r^{-1}(k)(x(k) - p^T f(X(k))) f(X(k)), \\ r(k+1) = r(k) + \left\| f(X(k+1)) \right\|^2, \end{cases} \tag{3}$$

which is the stochastic approximation procedure.

Here it should be mentioned, that exponentially weighted recurrent least squares method (1), having filtering and following properties, can be unstable under small values of parameter α; convergence of the algorithm (2) under the intensive disturbance ξ is disrupted, and stochastic approximation procedures, including (3), operate only in the stationary conditions.

For tuning of the first hidden layer parameters in AWNFN backpropagation learning algorithm based on the chain rule of differentiation and gradient descend optimization of local criterion

$$E(k) = \frac{1}{2}e^2(k) = \frac{1}{2}(x(k) - \hat{x}(k))^2$$

is used.

In the general case learning procedure in this layer has the form

$$\begin{cases} c_{ji}(k+1) = c_{ji}(k) - \eta_c(k)\dfrac{\partial E(k)}{\partial c_{ji}(k)}, \\ \sigma_{ji}(k+1) = \sigma_{ji}(k) - \eta_\sigma(k)\dfrac{\partial E(k)}{\partial \sigma_{ji}(k)}, \end{cases}$$

and its properties are completely determined by the learning rate parameter $\eta_c(k)$, $\eta_\sigma(k)$, selected according to the empirical reasons. It should be noticed that if the parameters of the fourth layer can be tuning most rapidly, the operation rate is lost in the first layer.

Increasing of the convergence rate can be achieved with more complex than gradient procedures, such as Hartley [30] or Marquardt [31] algorithms which can be written in general form [32]

$$\Phi(k+1) = \Phi(k) + \lambda(J(k)J^T(k) + \eta I)^{-1} J(k)e(k), \tag{4}$$

where $\Phi(k) = (c_{11}(k), \sigma_{11}^{-1}(k), c_{21}(k), \sigma_{21}^{-1}(k), \ldots, c_{ji}(k), \ \sigma_{ji}^{-1}(k), \ldots, c_{hn}(k), \sigma_{hn}^{-1}(k))^T$ is the $(2hn \times 1)$ tunable parameter vector (at that for the computation complexity reduction it includes not the width parameter σ_{ji}, but its inverse value σ_{ji}^{-1}), $J(k)$ is the $(2hn \times 1)$ gradient vector of output signal $\hat{x}(k)$ on the tunable parameters, I is the $(2hn \times 2hn)$ identity matrix, η is a scalar regularizing parameter, λ is the positive scalar gain.

To compute elements of gradient vector

$$J(k) = \left(\frac{\partial \hat{x}(k)}{\partial c_{11}}, \frac{\partial \hat{x}(k)}{\partial \sigma_{11}^{-1}}, \frac{\partial \hat{x}(k)}{\partial c_{21}}, \frac{\partial \hat{x}(k)}{\partial \sigma_{21}^{-1}}, \ldots, \frac{\partial \hat{x}(k)}{\partial c_{ji}}, \frac{\partial \hat{x}(k)}{\partial \sigma_{ji}^{-1}}, \ldots, \frac{\partial \hat{x}(k)}{\partial c_{hn}}, \frac{\partial \hat{x}(k)}{\partial \sigma_{hn}^{-1}} \right)^T$$

the chain rule can be used, at that

$$\begin{cases} \dfrac{\partial \hat{x}(k)}{\partial c_{ji}} = \dfrac{\partial \hat{x}(k)}{\partial \bar{w}_j} \cdot \dfrac{\partial \bar{w}_j}{\partial w_j} \cdot \dfrac{\partial w_j}{\partial \varphi_{ji}} \cdot \dfrac{\partial \varphi_{ji}}{\partial c_{ji}} = f_j(X(k))\bar{w}_j(k)(1 - \bar{w}_j(k)) \dfrac{1}{\varphi_{ji}(x_i(k), c_{ji}, \sigma_{ji}^{-1})} \cdot \dfrac{\partial \varphi_{ji}}{\partial c_{ji}}, \\ \dfrac{\partial \hat{x}(k)}{\partial \sigma_{ji}^{-1}} = \dfrac{\partial \hat{x}(k)}{\partial \bar{w}_j} \cdot \dfrac{\partial \bar{w}_j}{\partial w_j} \cdot \dfrac{\partial w_j}{\partial \varphi_{ji}} \cdot \dfrac{\partial \varphi_{ji}}{\partial \sigma_{ji}^{-1}} = f_j(X(k))\bar{w}_j(k)(1 - \bar{w}_j(k)) \dfrac{1}{\varphi_{ji}(x_i(k), c_{ji}, \sigma_{ji}^{-1})} \cdot \dfrac{\partial \varphi_{ji}}{\partial \sigma_{ji}^{-1}}. \end{cases}$$

where $\partial \varphi_{ji} / \partial c_{ji}$, $\partial \varphi_{ji} / \partial \sigma_{ji}^{-1}$ is partial derivatives of concrete wavelet activation function.

To reduce the computational complexity of the learning algorithm we can use the matrix inversion lemma in the form

$$(JJ^T + \eta I)^{-1} = \eta^{-1} I - \frac{\eta^{-1} IJJ^T \eta^{-1} I}{1 + J^T \eta^{-1} IJ},$$

using which it is easy to obtain the relation

$$\lambda (JJ^T + \eta I)^{-1} J = \lambda \frac{J}{\eta + \|J\|^2}.$$

Substituting this relation to the algorithm (4), we obtain first hidden layer parameters learning algorithm in the form

$$\Phi(k+1) = \Phi(k) + \lambda \frac{J(k)e(k)}{\eta + \|J(k)\|^2}. \tag{5}$$

It is easy to see, that algorithm (5) is the nonlinear additive-multiplicative modification of Kaczmarz algorithm, and under $\lambda = 1,\ \eta = 0$ coincides with it structurally.

To provide the filtering properties to the learning algorithm (5) let us introduce additional tuning procedure of the regularizing parameter η in the form [33, 34, 35]

$$\begin{cases} \Phi(k+1) = \Phi(k) + \lambda \dfrac{J(k)e(k)}{\eta(k)}, \\ \eta(k+1) = \alpha \eta(k) + \|J(k+1)\|^2. \end{cases} \tag{6}$$

If $\alpha = 0$, then this procedure coincides with (5) and has the highest rate of convergence, and if $\alpha = 1$, then this procedure obtains properties of stochastic approximation, and serves as generalization of procedure (3) in the nonlinear case.

Here it should be noticed, that the algorithm (6) is stable at any value of forgetting factor α , what favorably differs it from the exponentially weighted recurrent least squares method (1). As a result this procedure can be used too in the form

$$\begin{cases} p(k+1) = p(k) + \lambda_p \eta_p^{-1}(k)(x(k) - p^T(k) f(X(k))) f(X(k)), \\ \eta_p(k+1) = \alpha \eta_p(k) + \| f(x(k))\|^2 \end{cases} \tag{7}$$

for the fourth layer parameters tuning. One can notice close relation of the algorithms (1) and (7), as

$$\eta^{-1}(k) = TrP(k).$$

However algorithm (7) is much simpler in the computing implementation and easily reconstructs its properties from the most following to the most filtering ones.

Simulation results

To demonstrate the effectiveness of the proposed adaptive wavelet-neuro-fuzzy-network and its learning algorithm (6), (7), AWNFN was trained to emulate the nonlinear dynamical system which proposed in [36]. Emulation of the Narendra's dynamical system is a standard test, widely used to evaluate and compare the performance of neural and neuro-fuzzy systems for nonlinear system modeling and time series forecasting. The nonlinear dynamical system is generated by equation in form [36]

$$y(k+1) = 0.3y(k) + 0.6y(k-1) + f(u(k)), \tag{8}$$

where $f(u(k)) = 0.6\sin(u(k)) + 0.3\sin(3u(k)) + 0.1\sin(5u(k))$ and $u(k) = \sin(2k/250)$, k is discret time.

The values $x(t-4), x(t-3), x(t-2), x(t-1)$ were used to emulate $x(t+1)$. In the online mode of learning, AWNFN was trained with procedure (6), (7) using signal $u(k) = \sin 2k / 250$ for $k = 1...1000$. The parameters of the learning algorithm were $\alpha = 0.9$, $\lambda_p = 2$, $\lambda = 1$. Initial values were $\eta(0) = 1$ and $\eta_p(0) = 10000$. After 1000 iterations the training was stopped, and the next 800 points for the signals $u(k) = \sin 2k / 250$, $k = 1...300$ and $u(k) = 0.5\sin 2k / 250 + 0.5\sin 2k / 25$, $k = 501...1000$ were used as the testing data set to emulate dynamical system. As the activation function "Mexican hat" wavelet is used. Initial values of synaptic weights were generated in a random way from -0.1 to $+0.1$.

The root mean-square error (RMSE) was used as criterion for the quality of emulation

$$RMSE = \frac{1}{N} \sum_{k=1}^{N} (x(k) - \hat{x}(k))^2 .$$

Fig. 5 shows the results of nonlinear dynamical system emulation. The two curves, representing the actual (dot line) and emulation (solid line) values, are almost indistinguishable.

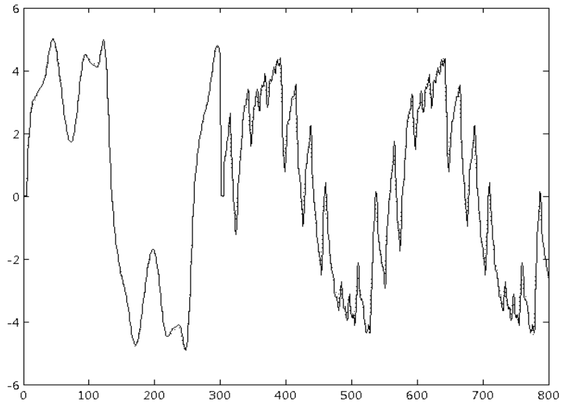

Fig. 5 – Emulation of the nonlinear dynamical system using adaptive wavelet-neuro-fuzzy network

Table 1 shows the results of the emulation process on the basis of the adaptive wavelet-neuro-fuzzy-network compared the results of emulation process on the basis of standard ANFIS with the backpropagation learning algorithm.

Table 1: The results of nonlinear dynamical system emulation

Neural network/ Learning algorithm	RMSE
Adaptive wavelet-neuro-fuzzy-network / Proposed learning algorithm (6), (7)	0.025
Backpropagation ANFIS	0.110

Thus as it can be seen from experimental results the proposed adaptive wavelet-neuro-fuzzy-network with the learning algorithm (6), (7) having the same number of adjustable parameters ensures the best quality of emulationt and high learning rate in comparison with conventional ANFIS architecture.

Conclusions

Computationally simple learning algorithms for the adaptive wavelet-neuro-fuzzy network in the forecasting and emulation of the nonlinear nonstationary signals tasks are proposed. The simulation of developed approach justifies the effectiveness of AWNFN using for solving wide category of emulation, forecasting and diagnostics problems.

Bibliography

[1]. Jang J.-S. R. Neuro-Fuzzy Modeling: Architectures, Analyses and Applications. Berkeley, CA: University of California. 1992, 155 p.

[2]. Jang J.-S. R. ANFIS: Adaptive network-based fuzzy inference systems. IEEE Trans. on Systems, Man, and Cybernetics. 23, 1993, P. 665-685.

[3]. Jang J.-S. R., Sun C.-T. Neuro-fuzzy modeling and control. Proc. IEEE. 83 (3), 1995, P. 378-406.

[4]. Jang J.-S. R., Sun C.-T., Muzutani E. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Upper Saddle River, N.J.: Prentice Hall, Inc., 1997, 614 p.

[5]. Kosko B. Fuzzy systems as universal approximators. Proc. 1-st IEEE Int. Conf. on Fuzzy Systems. San Diego, CA. 1992, P. 1153-1162.

[6]. Kreinovich V., Mouzouris Y. C., Nguen H. T. Fuzzy rule based modeling as a universal approximation tool. In “Fuzzy Systems: Modeling and Control”. Eds.: H. T. Nguen, M. Sugeno. Boston: Kluwer Academic Publishers. 1998, P. 135-195.

[7]. Hornik K., Stinchcombe M., White H. Multilayer feedforward networks are universal approximators. Neural Networks. 2, 1982, P. 359-366.

[8]. Scarcelli F., Tsoi A. S. Universal approximation using feedforward neural networks: a survey of some existing methods and some new results. Neural Networks. 11, 1998, P. 15-37.

[9]. Chui C. K. An Introduction to Wavelets. New York: Academic. 1992, 264 p.

[10]. Daubechies I. Ten Lectures on Wavelets. Philadelphia, PA: SIAM. 1992, 228 p.

[11]. Meyer Y. Wavelets: Algorithms and Applications. Philadelphia, PA: SIAM. 1993, 133 p.

[12]. Lekutai G., VanLandingham H. F. Self-tuning control of nonlinear systems using neural network adaptive frame wavelets. Proc. IEEE Int. Conf. on Systems, Man and Cybernetics. Piscataway, N.J. 2, 1997, P. 1017-1022.

[13]. Billings S. A., Wei H.-L. A new class of wavelet networks for nonlinear system identification. IEEE Trans. on Neural Networks. 16 (4), 2005, P. 862-874.

[14]. Zhang Q. H., Benveniste A. Wavelet networks. IEEE Trans. on Neural Networks. 3 (6), 1992, P. 889–898.

[15]. Oussar Y., Dreyfus G. Initialization by selection for wavelet network training. Neurocomputing. 34, 2000, P. 131–143.

[16]. Zhang J., Walter G. G., Miao Y., Lee W. N. W. Wavelet neural networks for function learning. IEEE Trans. on Signal Process. 43(6), 1995, P. 1485–1497.

[17]. Zhang Q. H. Using wavelet network in nonparametric estimation. IEEE Trans. on Neural Networks. 8 (2), 1997, P. 227–236.

[18]. Bodyanskiy Ye., Lamonova N., Pliss I., Vynokurova O. An adaptive learning algorithm for a wavelet neural network. Blackwell Synergy: Expert Systems. 22 (5), 2005, P. 235-240.

[19]. Avci E., Turkoglu I., Poyraz M. Intelligent target recognition based on wavelet adaptive network based fuzzy inference system. Lecture Notes on Computer Science. Berlin-Heidelberg: Springer-Verlag, 3522, 2005, P. 594-603.

[20]. Takagi T., Sugeno M. Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. on Systems, Man, and Cybernetics. 15, 1985, P. 115-132.

[21]. Sugeno M., Kang G. T. Structure identification of fuzzy model. Fuzzy Sets and Systems, 28, 1998, P. 15-33.

[22]. Vynokurova O., Lamonova N., Pliss I. Analytic wavelets generator. Problemy Bioniki, 60, 2004, P. 104-109. (in Russian)

[23]. Bodyanskiy Ye., Vynokurova O. Variable form of triangular wavelet in the wavelet neural networks. Proc. of 13th International Conference on Automatic Control “Automatics-2006”, Vinnitsa, 2006, P. 393. (in Russian)

[24]. Mitaim S., Kosko B. What is the best shape for a fuzzy set in function approximation? Proc. 5th IEEE Int. Conf on Fuzzy Systems “Fuzz-96”, 2, 1996, P. 1237-1213.

[25]. Mitaim S., Kosko B. Adaptive joint fuzzy sets for function approximation. Proc. Int. Conf. on Neural Networks “ICNN-97”, 1997, P. 537-542.

[26]. Ljung L. System Identification: Theory for the User. PTR Prentice Hall, Upper Saddle River, N.J., 1999

[27]. Kaczmarz S. Angenaeherte Ausloesung von Systemen linearer Gleichungen. Bull. Int. Acad. Polon. Sci., Let. A, 1973, S. 355-357.

[28]. Kaczmarz S. Approximate solution of systems of linear equations. Int. J. Control, 53, 1993, P. 1269-1271.

[29]. Goodwin Y. C., Ramadge P. J., Caines P. E. A globally convergent adaptive predictor. Automatica, 17, 1981, P. 135-140.

[30]. Hartley H. The modified Gauss-Newton method for the fitting of nonlinear regression functions of least squares. Technometrics, 3, 1961, P. 269-280.

[31]. Marquardt D. An algorithm for least squares estimation of nonlinear parameters. SIAM J. Appl. Math., 11, 1963, P. 431-441.

[32]. Bodyanskiy Ye. Adaptive identification algorithm of nonlinear control object. Automatic Control System and Devices of Automatic, Kharkiv: Vyshcha Shk., 81, 1987, P. 43-46. (in Russian)

[33]. Bodyanskiy Ye., Pliss I., Solovyova T. Multistage optimal predictors of multidimensional nonstationary stochastic processes. Docl. AN USSR. 12, 1986, P. 41-49.

[34]. Bodyanskiy Ye., Kolodyazhniy V., Stephan A. An adaptive learning algorithm for a neuro-fuzzy network. Ed. by B. Reusch "Computational Intelligence. Theory and Applications." Berlin-Heidelberg-New York: Springer-Verlag, 2001, P. 68-75.

[35]. Bodyanskiy Ye., Kolodyazhniy V., Otto P. A new learning algorithm for a forecasting neuro-fuzzy network. Integrated Computer-Aided Engineering. Amsterdam: IOS Press, 10 (4), 2003, P. 399-409.

[36]. Narendra K.S., Parthasarathy K. Identification and control of dynamic systems using neural networks. IEEE Trans. on Neural Networks, 1990, 1(1), P. 4-26.

Authors' Information

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

Pliss Iryna - 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, Ukraine, 61166, Tel +380577021890, pliss@kture.kharkov.ua

Vynokurova Olena - Candidate of Technical Sciences (equivalent Ph.D.), Senior Researcher of the Control Systems Research Laboratory, Kharkiv National University of Radio Electronic, Lenina av. 14, Kharkiv, Ukraine, 61166, Tel +380577021890, vinokurova@kture.kharkov.ua

MULTIALGEBRAIC SYSTEMS IN INFORMATION GRANULATION

Alexander Kagramanyan, Vladimir Mashtalir, Vladislav Shlyakhov

Abstract: In different fields a conception of granules is applied both as a group of elements defined by internal properties and as something inseparable whole reflecting external properties. Granular computing may be interpreted in terms of abstraction, generalization, clustering, levels of abstraction, levels of detail, and so on. We have proposed to use multialgebraic systems as a mathematical tool for synthesis and analysis of granules and granule structures. The theorem of necessary and sufficient conditions for multialgebraic systems existence has been proved.

Keywords: granular computing, multirelations, multioperations.

ACM Classification Keywords: I.2.4 Knowledge representation formalisms and methods: relation systems.

Introduction

Granular computing explores knowledge from different standpoints to reveal various types of structures and information embedded in the data [Zadeh, 1997, Bargiela, Pedrycz, 2002]. A paradigm of granular computing consists in grouping elements together (in a granule) by indistinguishability, similarity, proximity or functionality in arbitrary feature or signal spaces. Taking into account a semantic interpretation of why two objects are put into the same granule and how two objects are related with each other it provides one of a general methodology for intelligent data analysis on different levels of roughening or detailing [Pal et al., 2005, Yao, Yao, 2002].