THE CASCADE GROWING NEURAL NETWORK USING QUADRATIC NEURONS AND ITS LEARNING ALGORITHMS FOR ON-LINE INFORMATION PROCESSING

Yevgeniy Bodyanskiy, Yevgen Viktorov, Iryna Pliss

Abstract: New non-conventional system of the computational intelligence is proposed. It has growing structure similar to the Cascade-Correlation Learning Architecture designed by S. E. Fahlman and C. Lebiere but differs from it in type of artificial neurons. Quadratic neurons are used as nodes in introduced architecture. These simple elements can be quickly adjusted using high-speed learning procedures. Proposed approach allows to reduce time required for weight coefficients adjustment and to decrease training dataset size in comparison with conventional neural networks. Also on-board realization of quadratic neuron is quite simple and therefore implementation of entire cascade architecture in hardware is very easy.

Keywords: artificial neural networks, constructive approach, quadratic neuron, real-time processing, online learning.

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

Introduction

Nowadays artificial neural networks (ANNs) are widely applied for solving a large class of problems related with the processing of information given as time-series or numerical “object-properties” tables generated by the non-stationary, chaotic or stochastic systems. The most attractive ANNs properties are their approximating possibilities and learning capabilities.

Conventionally “learning” is considered as process of the neural network’s synaptic weights adjustment accordingly to selected optimization procedure of accepted learning criterion [Cichocki, 1993; Haykin, 1999]. But during learning procedure not only weight coefficients but also network’s architecture (quantity of nodes) can be adjusted for the purpose of increasing quality of received results. There are two basic approaches for the neural network architecture adjustment: 1) “constructive approach” [Platt, 1991; Nag, 1998; Yingwei, 1998] — starts with simple architecture and gradually adds new nodes during learning; 2) “destructive approach” [Cun, 1990; Hassibi, 1993; Prechelt, 1997] — starts with initially redundant network and simplifies it throughout learning process.

Obviously, constructive approach needs less computational resources and within the bounds of this technique the cascade neural networks (CNNs) [Fahlman, 1990; Schalkoff, 1997; Avedjan, 1999] can be marked out. The most efficient representative of the CNNs is the Cascade-Correlation Learning Architecture (CasCorLA) [Fahlman, 1990]. This network begins with the simplest architecture which consists of a single neuron. Throughout a learning procedure new neurons are added to the network, producing a multilayer 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. The CasCorLA authors, S. E. Fahlman and C. Lebiere, point out high speed of the 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 an output signal of each neuron is non-linearly depended from its weight coefficients. Therefore it is necessary to use gradient learning methods such as delta-rule or its modifications, and optimization an operation speed becomes impossible. In connection with the above it seems to be reasonable to synthesize the cascade architecture based on the elementary nodes with linear or quadratic dependence of an output signal from the synaptic weights. It allows to increase a speed of synaptic weights adjustment and to reduce minimally required size of training set.

In [Bodyanskiy, 2007] ortho-neurons were proposed as such nodes. Also it was shown how simply and effectively an approximation of sufficiently non-linear function can be performed using this technique. But it should be noticed that on-board realization of the ortho-neuron is quite complex due to its functional specificities. At this paper we propose to use quadratic neurons as basic elements for the cascade architecture. They have simple structure and therefore their realization in hardware is simple too.

The Quadratic Neuron and Its Gradient Learning Procedure

The quadratic neuron is a nonlinear in inputs but linear in synaptic weights multi-input single output system shown on Fig. 1. It realizes the following mapping:

$$\hat{y}_j(k) = \theta_j(k) + \sum_{i=1}^{n} w_{ji}(k)x_i(k) + \sum_{p=1}^{n}\sum_{l=p}^{n} w_{jpl}(k)x_p(k)x_l(k) \qquad (1)$$

where x_i is the i-th input (i=1,2,…,n); $\hat{y}$ is an output; θ_j is a bias in the j-th quadratic neuron; w_{ji} is a weight coefficient connected to i-th input in the j-th quadratic neuron; w_{jpl} is a weight coefficient connected to composition of p-th and l-th inputs in the j-th quadratic neuron.

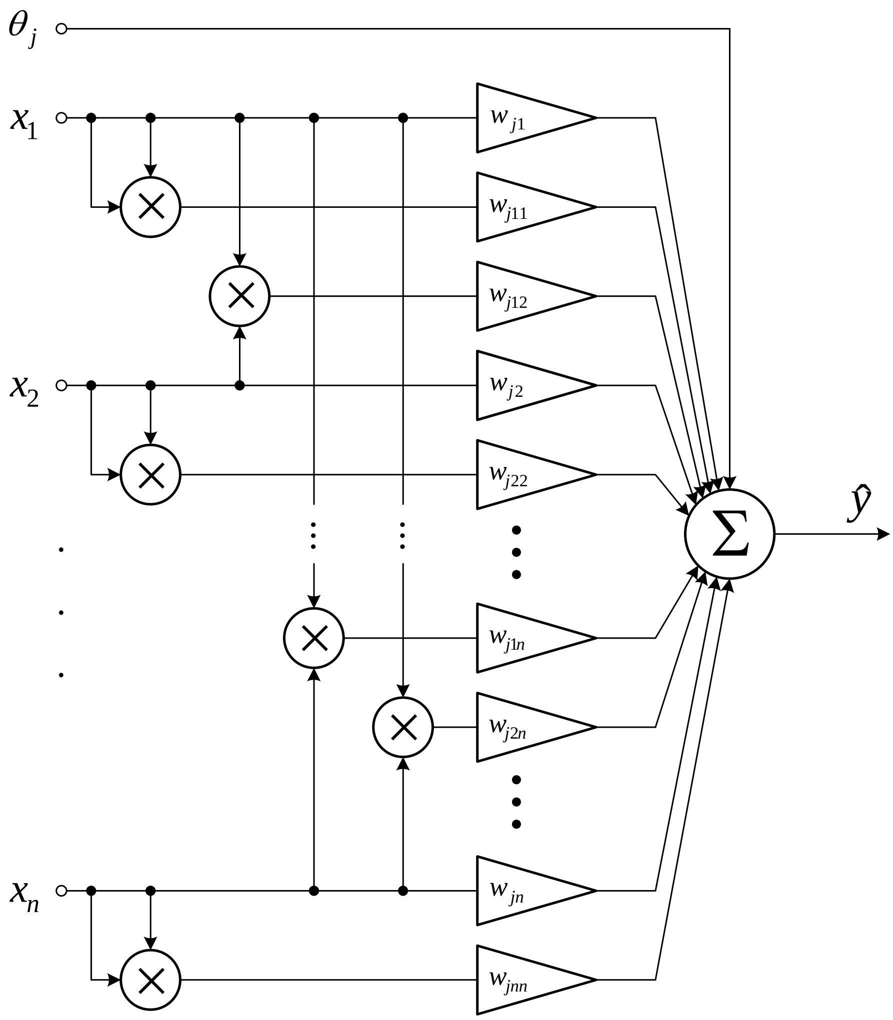

Figure 1. The Quadratic Neuron – QN

Let us define *additional* designations $w_{j0}(k)=\theta_j(k)$, $b_j(k)=(w_{j1}(k),w_{j2}(k),...,w_{jn}(k))^T$ – $(n\times 1)$-vector, $C_j(k)=\{w_{jpl}(k)\}$ – $(n\times n)$-matrix, $x\_(k)=(x_1(k),x_2(k),...,x_n(k))^T$ – $(n\times 1)$-vector, $x(k)=(1,x_-^T(k))^T$ – $((n+1)\times 1)$-vector. Then we can rewrite expression (1) in the form:

$$\hat{y}_j(k)=w_{j0}(k)+b_j^T(k)x_-(k)+x_-^T(k)C_j(k)x_-(k)$$

or

$$\hat{y}_j(k)=x^T(k)W_j(k)x(k)$$

where

$$W_j(k)=\left(\begin{array}{c|c} w_{j0}(k) & 0.5b_j^T(k) \\ \hline 0.5b_j(k) & C_j(k)\end{array}\right) \quad (2)$$

is block $((n+1)\times(n+1))$-matrix.

Weight coefficients matrix $W_j(k)$ adjustment can be performed by minimization of the quadratic learning criterion

$$E_j(k)=\frac{1}{2}e_j^2(k)=\frac{1}{2}(y(k)-x^T(k)W_jx(k))^2$$

(where $y(k)$ is an *external* learning signal) using gradient algorithm:

$$W_j(k+1)=W_j(k)+\eta(k)e_j(k)x(k)x^T(k) \quad (3)$$

where

$$e_j(k)=y(k)-x^T(k)W_j(k)x(k).$$

For the purpose of *evaluation* parameter $\eta(k)$ which provides optimal rate of convergence to algorithm (3) let us define values deviation matrix

$$\tilde{W}_j(k)=W_j-W_j(k)$$

where W_j is unknown matrix of optimal coefficients values, $W_j(k)$ (2) is its estimate on the *k*-th learning iteration.

Then solving the differential equation

$$\frac{\partial Tr(\tilde{W}_j(k)\tilde{W}_j^T(k))}{\partial\eta}=0$$

(where $Tr(\bullet)$ is trace of matrix) optimal value of the step parameter can be obtained in the form [Bodyanskiy, 1987; Bodyanskiy, 1997]:

$$\eta(k)=\|x(k)\|^{-4}.$$

Using evaluated step parameter, expression (3) can be rewritten as

$$W_j(k+1)=W_j(k)+\frac{d_j(k)-x^T(k)W_j(k)x(k)}{\|x(k)\|^4}x(k)x^T(k). \quad (4)$$

Learning procedure (4) is Kaczmarz-Widrow-Hoff [Kaczmarz, 1937; Kaczmarz, 1993; Widrow, 1960] optimal algorithm extension for quadratic neuron.

As it can be readily seen the quadratic neuron is a generalization of the well known N-Adaline widely used in GMDH Neural Networks [Pham, 1995].

Quadratic neuron provides quite high precision of approximation and extrapolation of significantly non-stationary non-linear signals and processes but further we use it as an elementary node in the cascade architecture.

The Cascade Neural Network Based On Quadratic Neurons

The architecture of cascade neural network based on quadratic neurons is shown on Fig. 2

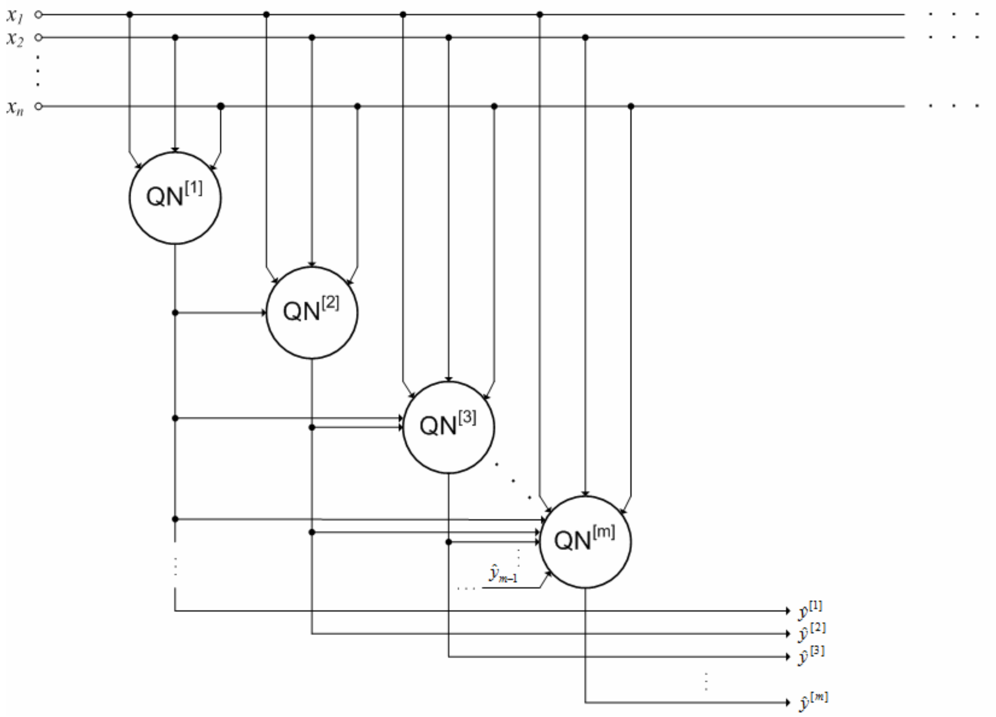

Figure 2. The Cascade Neural Network based on Quadratic Neurons

and mapping that it realizes has the following form:

– first cascade quadratic neuron

$$\hat{y}^{[1]}(k)=\theta_1(k)+\sum_{i=1}^{n}w_{1i}(k)x_i(k)+\sum_{p=1}^{n}\sum_{l=p}^{n}w_{1pl}(k)x_p(k)x_l(k)=x^{[1]T}w^{[1]}(k)x^{[1]}(k),\ \ x^{[1]}=(1,x^T(k))^T,$$

– second cascade quadratic neuron

$$\hat{y}^{[2]}(k)=\theta_2(k)+\sum_{i=1}^{n}w_{2i}(k)x_i(k)+\sum_{p=1}^{n}\sum_{l=p}^{n}w_{2pl}(k)x_p(k)x_l(k)+w_{2(n+1)}\hat{y}_1+\sum_{p_x=1}^{n}w_{2p_x,n+1}x_{p_x}\hat{y}_1+w_{2,n+1,n+1}\hat{y}_1\hat{y}_1=$$

$$=x^{[2]T}w^{[2]}(k)x^{[2]}(k),\ \ x^{[2]}=(x^{[1]T},\hat{y}^{[1]})^T,$$

– third cascade quadratic neuron

$$\hat{y}^{[3]}(k)=\theta_3(k)+\sum_{i=1}^{n}w_{3i}(k)x_i(k)+\sum_{p=1}^{n}\sum_{l=p}^{n}w_{3pl}(k)x_p(k)x_l(k)+\sum_{i_y=1}^{2}w_{3,n+i_y}\hat{y}_{i_y}+\sum_{l_x=1}^{2}\sum_{p_x=1}^{n}w_{3p_x,n+l_x}x_{p_x}\hat{y}_{l_x}+$$

$$+\sum_{l_y=1}^{2}\sum_{p_y=1}^{l_y}w_{3,n+p_y,n+l_y}\hat{y}_{p_y}\hat{y}_{l_y}=x^{[3]T}w^{[3]}(k)x^{[3]}(k),\ \ x^{[3]}=(x^{[2]T},\hat{y}^{[2]})^T,$$

– *m*-th cascade neuron

$$\hat{y}^{[m]}(k)=\theta_m(k)+\sum_{i=1}^{n}w_{mi}(k)x_i(k)+\sum_{p=1}^{n}\sum_{l=p}^{n}w_{mpl}(k)x_p(k)x_l(k)+\sum_{i_y=1}^{n+m-1}w_{m,n+i_y}\hat{y}_{i_y}+$$

$$\sum_{l_x=1}^{n+m-1}\sum_{p_x=1}^{n}w_{mp_x,n+l_x}x_{p_x}\hat{y}_{l_x}+\sum_{l_y=1}^{n+m-1}\sum_{p_y=1}^{l_y}w_{m,n+p_y,n+l_y}\hat{y}_{p_y}\hat{y}_{l_y}=x^{[m]T}w^{[m]}(k)x^{[m]}(k),\ \ x^{[m]}=(x^{[m-1]T},\hat{y}^{[m-1]})^T \quad (5)$$

where *m* is quantity of cascades.

Thus the cascade neural network based on quadratic neurons contains $\sum_{q_1=1}^{n+1} q_1 + \sum_{q_2=1}^{n+2} q_2 + ... + \sum_{q_m=1}^{n+m} q_m$ adjustable parameters and it is important that all of them are linearly included in the description (5).

The Cascade Neural Network Based On Quadratic Neuron Learning Procedure

The cascade neural network learning can be performed in both the batch mode and the mode of sequential information processing using global learning criterion (6)

$$E_N^{[j]} = \frac{1}{2}\sum_{k=1}^{N} e_j(k)^2 = \frac{1}{2}\sum_{k=1}^{N} (y(k) - \hat{y}_j(k))^2. \quad (6)$$

Firstly, let us consider situation when the training data set is defined *a priory,* i.e. we have a set of points $x\_(1), y(1); x\_(2), y(2);...; x\_(k), y(k);...; x\_(N), y(N).$ Then for quadratic neuron of the first layer (QN[1]) weight coefficients vector is defined in form $w^{[1]} = = (w_{10}, w_{11},..., w_{1n}, w_{111}, w_{122},..., w_{1nn}, w_{112}, w_{113},..., w_{11n}, w_{123}, w_{124},..., w_{12n}, w_{134},..., w_{1,n-1,n})^T$ and also corresponding vector of internal quadratic neuron signals is defined as well $s^{[1]}(k) = (\theta_1, x_1(k), x_2(k),..., x_n(k), x_1(k)x_1(k), x_2(k)x_2(k),..., x_n(k)x_n(k), x_1(k)x_2(k), x_1(k)x_3(k),..., x_1(k)x_n(k), x_2(k)x_3(k), x_2(k)x_4(k),..., x_2(k)x_n(k), x_3(k)x_4(k),...x_{n-1}(k)x_n(k))^T$.

Then using direct minimization of the learning criterion (6) vector of synaptic weights can be evaluated in the form

$$w^{[1]}(N) = \left(\sum_{k=1}^{N} s^{[1]}(k)s^{[1]T}(k)\right)^{+} \sum_{k=1}^{N} s^{[1]}(k)y(k) = P^{[1]}(N)\sum_{k=1}^{N} s^{[1]}(k)y(k) \quad (7)$$

where $(\bullet)^{+}$ denotes the Moore-Penrose pseudoinversion.

In the case of sequential data processing recurrent form of the least squares method can be used instead of procedure (7):

$$\begin{cases} w^{[1]}(k+1) = w^{[1]}(k) + \dfrac{P^{[1]}(k)(y(k+1) - w^{[1]T}(k)s^{[1]}(k+1))}{1 + s^{[1]T}(k+1)P^{[1]}(k)s^{[1]}(k+1)} s^{[1]}(k+1), \\ P^{[1]}(k+1) = P^{[1]}(k) - \dfrac{P^{[1]}(k)s^{[1]}(k+1)s^{[1]T}(k+1)P^{[1]}(k)}{1 + s^{[1]T}(k+1)P^{[1]}(k)s^{[1]}(k+1)}, \quad P^{[1]}(0) = \beta I \end{cases} \quad (8)$$

where β is a large positive number and I is a unity matrix of corresponding dimensionality.

Using of adaptive algorithms (3) or (4) is also possible and leads to reducing of computational complexity of learning process. But utilization of learning procedure (7) or (8) essentially reduces a learning time in comparison with gradient algorithms underlying delta-rule and backpropagation.

After the first cascade learning completion, the synaptic weights of the quadratic neuron QN[1] become “frozen”, all values $(\hat{y}^{[1]}(1), \hat{y}^{[1]}(2),..., \hat{y}^{[1]}(k),..., \hat{y}^{[1]}(N))$ are evaluated and the second cascade of the network which consists of a single quadratic neuron QN[2] is generated. It has one additional input for the output signal of the first cascade. Then the procedure (7) or (8) is again applied for adjusting a vector of weight coefficients $w^{[2]}$, which has dimensionality is $\sum_{q_2=1}^{n+2} q_2$.

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

$$w^{m1]}(N)=\left(\sum_{k=1}^{N}s^{[m]}(k)s^{[m]T}(k)\right)^{+}\sum_{k=1}^{N}s^{[m]}(k)y(k)=P^{[m]}(N)\sum_{k=1}^{N}s^{[m]}(k)y(k)$$

or

$$\begin{cases} w^{[m]}(k+1)=w^{[m]}(k)+\dfrac{P^{[m]}(k)(y(k+1)-w^{[m]T}(k)s^{[m]}(k+1))}{1+s^{[m]T}(k+1)P^{[m]}(k)s^{[m]}(k+1)}s^{[m]}(k+1), \\ P^{[m]}(k+1)=P^{[m]}(k)-\dfrac{P^{[m]}(k)s^{[m]}(k+1)s^{[m]T}(k+1)P^{[m]}(k)}{1+s^{[m]T}(k+1)P^{[m]}(k)s^{[m]}(k+1)},\ \ P^{[m]}(0)=\beta I \end{cases}$$

where vectors $w^{[m]}$ and $s^{[m]}$ have dimensionalities $\sum_{q_m=1}^{n+m}q_m$.

Simulation Results

In order to confirm efficiency of introduced architecture we have solved a dynamic plant identification problem. Proposed dynamic plant [Patra, 2002; Narendra, 1990] can be defined by the equation:

$$y(k+1)=0.3y(k)+0.6y(k-1)+f(u(k))$$

where

$$f(u)=0.6\sin u+0.3\sin 3u+0.1\sin 5u.$$

There was generated a sequence which contained 1500 values of signal for *k*=1,2,…,1500. On training set signal $u(k)=\sin 2k/250$ (*k*=1,…,500) have been used and on the testing set $u(k)=\sin 2k/250$ (*k*=501,…,1000), $u(k)=0.5\sin 2k/250+0.5\sin 2k/25$ (*k*=1001,…1500). It means that on testing set sinusoidal component of the dynamic object changes and therefore output signal changes its form too. Obtained set was normalized on interval [-1 1].

For estimation of received results we have used normalized mean square error:

$$NRMSE(k,N)=\frac{\sum_{q=1}^{N}e^{2}(k+q)}{N\sigma}$$

where σ is a mean square deviation of the predicted process on the training set.

During simulation modeling we have used least squares method as well as adaptive algorithm (4) for the purpose of adjusting synaptic weight coefficients inside quadratic neurons. Also, the same problem had been solved using conventional multilayer perceptron. Obtained results are given in table 1 and on figure 3.

Table 1. Results of the dynamic object identification.

Artificial Neural Network	NRMSE
Multilayered perceptron (50 epochs using Levenberg-Marquardt procedure)	0.0011
Cascade architecture – 3 cascades (batch mode using LSM)	0.0009
Cascade architecture (mode of sequential real-time data processing using adaptive algorithm (4) – 1 epoch)	0.0015

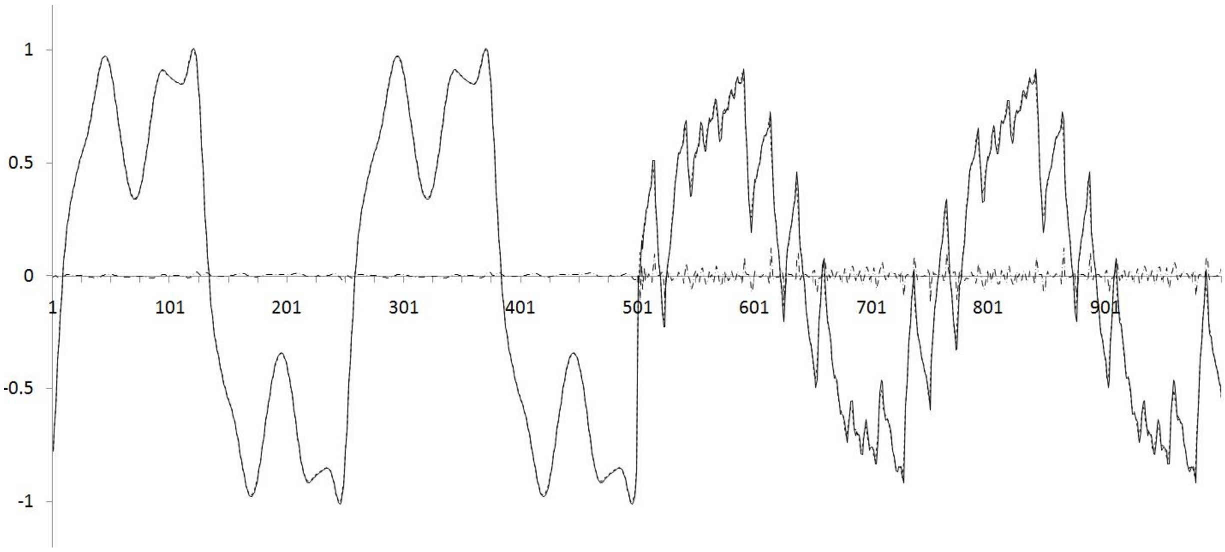

Figure 3. Dynamic object identification using cascade architecture trained with LSM: object output – solid line; network output – dashed line; identification error – chain line.

Conclusion

The Cascade Neural Network based on Quadratic Neurons is proposed. It differs from the known cascade networks in increased speed of operation, real-time processing possibility and simplicity of its on-board realization. Theoretical justification and experiment results confirm the efficiency of developed approach to cascade systems synthesis.

Bibliography

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

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

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

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

[Yingwei, 1998] 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.

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

[Hassibi, 1993] Hassibi B. Stork D. G. Second-order derivatives for network pruning: Optimal brain surgeon. In: Advances in Neural Information Processing Systems. Eds. Hanson et al. San Mateo, CA: Morgan Kaufman, 1993. P. 164-171.

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

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

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

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

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

[Bodyanskiy, 1987] Ye. V. Bodyanskiy, Yu. A. Bobrovskiy. Adaptivnaya identifikaciya, prognozirovanie i upravlenie na osnove kvadratichnyx modelei. Kharkiv, 1987, 33 p. Ruk. dep. v UkrNIINTI 25.05.2987, №1515 – Uk 87.

[Bodyanskiy, 1997] Ye. V. Bodyanskiy, S. G. Udovenko, A. Ye. Achkasov, G. K. Voronovskii. Suboptimal'noe upravlenie stohasticheskimi processami. Kharkiv, Osnova, 1997. 140 p.

[Pham, 1995] D. T. Pham, X. Liu. Neural Networks for Identification Prediction and Control. London, Springer-Verlag, 1995.

[Kaczmarz, 1937] S. Kaczmarz. Angenaeherte Ausloesung von Systemen Linearer Gleichungen. Bull. Int. Acad. Polon. Sci, Let. A. 1937. S. 355-357.

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

[Widrow, 1960] B. Widrow, M. E. Hoff Jr. Adaptive switching circuits. 1960 IRE WESCON Convention Record, Part 4. N.Y.: IRE, 1960. P. 96-104.

[Patra, 2002] J.C. Patra, A.C. Kot. Nonlinear dynamic system identification using Chebyshev functional link artificial neural network // IEEE Trans. on System, Man and Cybernetics, Part B. 2002, 32, №4. P. 505-511.

[Narendra, 1990] K.S. Narendra, K. Parthasarathy. Identification and control of dynamic systems using neural networks // IEEE Trans. on Neural Networks. 1990, 1, № 1. P. 4-26.

Authors' Information

Yevgeniy Bodyanskiy *– Professor, Dr.-Ing. habil., 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*

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*

Iryna Pliss *– Dr.-Ing., 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*