ITHEA

International Journal INFORMATION THEORIES & APPLICATIONS

2009 Volume 16 Number 3

International Journal
INFORMATION THEORIES & APPLICATIONS
Volume 16 / 2009, Number 3

IJ ITA is official publisher of the scientific papers of the members of
the ITHEA® International Scientific Society

IJ ITA welcomes scientific papers connected with any information theory or its application.

IJ ITA rules for preparing the manuscripts are compulsory.
The **rules for the papers** for IJ ITA as well as the **subscription fees** are given on *www.ithea.org* .
The camera-ready copy of the paper should be received by http://ij.ithea.org.
Responsibility for papers published in IJ ITA belongs to authors.

General Sponsor of IJ ITA is the **Consortium FOI Bulgaria** (www.foibg.com).

International Journal "INFORMATION THEORIES & APPLICATIONS" Vol.16, Number 3, 2009

Printed in Bulgaria

Edited by the **Institute of Information Theories and Applications FOI ITHEA®**, Bulgaria,
in collaboration with the V.M.Glushkov Institute of Cybernetics of NAS, Ukraine,
and the Institute of Mathematics and Informatics, BAS, Bulgaria.

Publisher: ITHEA®
Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org, e-mail: info@foibg.com

ISSN 1310-0513 (printed) ISSN 1313-0463 (online) ISSN 1313-0498 (CD/DVD)

THE CASCADE NEO-FUZZY ARCHITECTURE USING CUBIC–SPKINE ACTIVATION FUNCTIONS

Yevgeniy Bodyanskiy, Yevgen Viktorov

Abstract: in the paper new hybrid system of computational intelligence called the Cascade Neo-Fuzzy Neural Network (CNFNN) is introduced. This architecture has the similar structure with the Cascade-Correlation Learning Architecture proposed by S.E. Fahlman and C. Lebiere, but differs from it in type of artificial neurons. CNFNN contains neo-fuzzy neurons, which can be adjusted using high-speed linear learning procedures. Proposed CNFNN is characterized by high learning rate, low size of learning sample and its operations can be described by fuzzy linguistic "if-then" rules providing "transparency" of received results, as compared with conventional neural networks. Using of cubic-spline membership functions instead of conventional triangular functions allows increasing accuracy of smooth functions approximation.

Keywords: artificial neural networks, constructive approach, fuzzy inference, hybrid systems, neo-fuzzy neuron, cubic-spline functions.

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

Introduction

At the present time artificial neural networks are widely applied for solving identification, prediction, and modeling problems of significantly nonlinear processes when information given as time-series or numerical "object-property" tables generated by stochastic or chaotic systems. However in real conditions data processing often must be performed simultaneously with the plant functioning and therefore timing budget becomes quite valuable. So called "optimization-based networks" such as Multilayer Perceptron, Radial Basis Functions Network (RBFN), Normalized Radial Basis Functions Network (NRBFN) in most cases can be ineffective to solve mentioned above problems because of their low convergence rate during learning procedure, curse of dimensionality, and impossibility to learn in on-line mode.

Traditionally by the learning we understand the process of the neural network's synaptic weights adjustment accordingly to selected optimization procedure of the accepted learning criterion [Cichocki, 1993; Haykin, 1999]. Quality of the received results can be improved not only by adjusting weight coefficients but also by adjusting architecture of the neural network (number of nodes). There are two basic approaches of 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 operation speed optimization becomes impossible. In connection with the above it seems to be reasonable to synthesize the cascade architecture based on the elementary nodes with linear 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, 2007a] the ortho-neurons were proposed as such nodes. Also it was shown how simply and effectively an approximation of sufficiently complex function can be performed using this technique. In [Bodyanskiy, 2004a; Bodyanskiy, 2004b; Bodyanskiy, 2006a; Bodyanskiy, 2006b; Bodyanskiy, 2007b; Bodyanskiy, 2008a; Viktorov, 2008] the orthogonal and the cascade orthogonal neural networks were introduced. These architectures have shown quite good results during simulation modeling, significantly exceeding the conventional cascade neural networks in training speed.

It is well known the main ANN's disadvantage is a non-interpretability of received results, i.e. trained neural network is a "black box", and often their usage is restrained because of this reason. An interpretability and transparency together with the learning capabilities are the main properties of the neuro-fuzzy systems [Jang, 1997], which can be trained using backpropagation and in consequence the time required for weights tuning and the size of a training set are significantly increase. The neural network which allows to avoid these disadvantages was introduced in [Bodyanskiy, 2008b]. It has the cascade architecture and uses neo-fuzzy neurons [Yamakawa, 1992; Uchino, 1997; Miki, 1999] as nodes. Traditionally triangular functions are used as membership functions in neo-fuzzy neuron. Therefore when we have deal with a process described by smooth function we should either increase quantity of membership functions, what leads to increasing of the time required for weight coefficients adjustment, or quality of obtained results would be reduced. At this paper an attempt to get over this difficulty is taken.

The Neo-Fuzzy Neuron

Neo-fuzzy neuron is a nonlinear multi-input single-output system shown in Fig.1.

It realizes the following mapping:

$$\hat{y} = \sum_{i=1}^{n} f_i(x_i) \tag{1}$$

where x_i is the *i*-th input (*i* = 1,2,…,*n*), $\hat{y}$ is a system output. Structural blocks of neo-fuzzy neuron are nonlinear synapses NS$_i$ which perform transformation of *i*-th input signal in the from

$$f_i(x_i) = \sum_{j=1}^{h} w_{ji}\mu_{ji}(x_i).$$

Each nonlinear synapse realizes the fuzzy inference

IF x_i IS x_{ji} THEN THE OUTPUT IS w_{ji}

where x_{ji} is a fuzzy set which membership function is μ_{ji}, w_{ji} is a singleton (synaptic weight) in consequent [Uchino, 1997]. As it can be readily seen nonlinear synapse in fact realizes Takagi-Sugeno fuzzy inference of zero order.

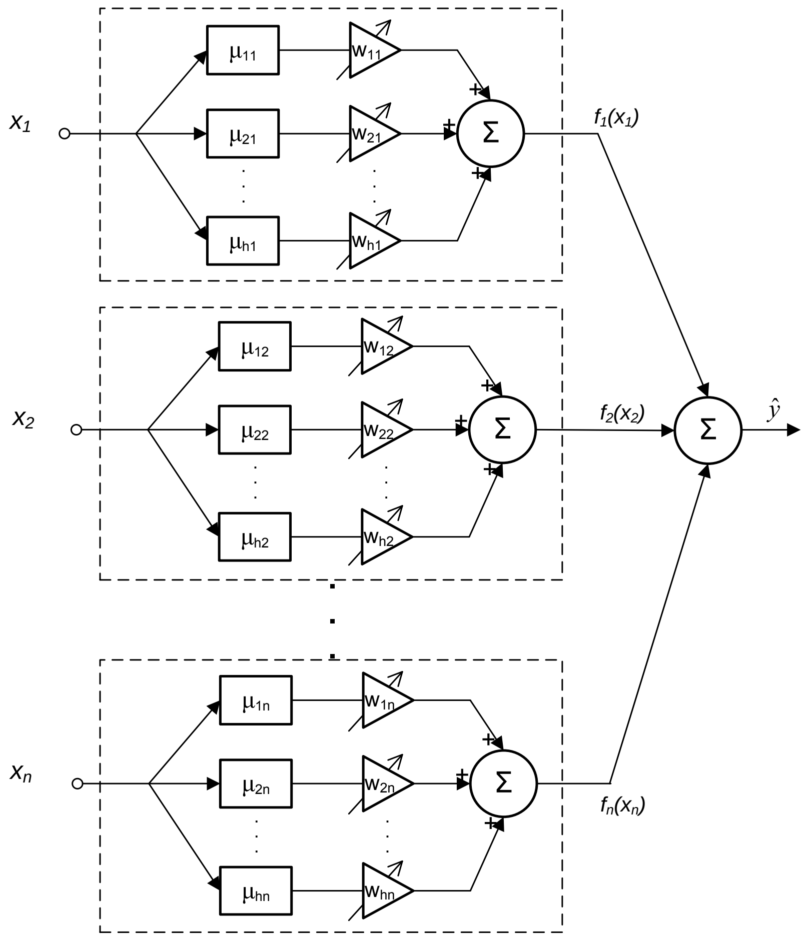


Figure 1. The Neo-Fuzzy Neuron

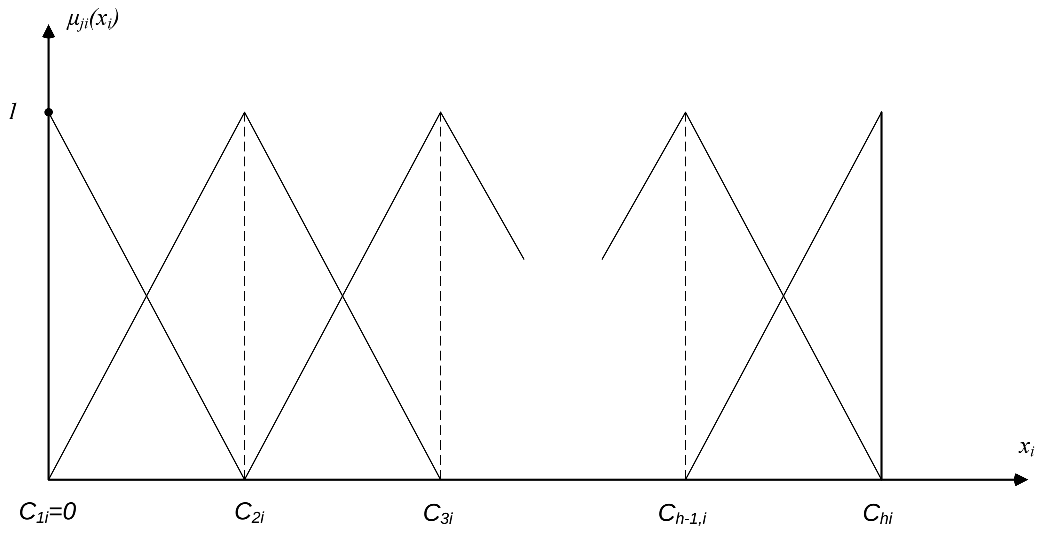


Figure 2. Triangular membership functions

Conventionally the membership functions $\mu_{ji}(x_i)$ in the antecedent are complementary triangular functions as shown in Fig. 2.

For preliminary normalized input variables x_i (usually $0 \le x_i \le 1$), membership functions can be expressed in the form:

$$\mu_{ji}(x_i) = \begin{cases} \dfrac{x_i - c_{j-1,i}}{c_{ji} - c_{j-1,i}}, x \in [c_{j-1,i}, c_{ji}], \\ \dfrac{c_{j+1,i} - x_i}{c_{j+1,i} - c_{ji}}, x \in [c_{ji}, c_{j+1,i}], \\ 0 - otherwise \end{cases}$$

where c_{ji} are arbitrarily selected centers of corresponding membership functions. Usually they are equally distributed on interval [0, 1]. This contributes to simplify the fuzzy inference process. That is, an input signal x_i activates only two neighboring membership functions simultaneously and the sum of the grades of these two membership functions equals to unity (Ruspini partitioning), i.e.

$$\mu_{ji}(x_i) + \mu_{j+1,i}(x_i) = 1. \tag{2}$$

Thus, the fuzzy inference result produced by the Center-of-Gravity defuzzification method can be given in the very simple form:

$$f_i(x_i) = w_{ji}\mu_{ji}(x_i) + w_{j+1,i}\mu_{j+1,i}(x_i). \tag{3}$$

By summing up $f_i(x_i)$, the output $\hat{y}$ of Eq. (1) is produced.

When a vector signal $x(k) = (x_1(k), x_2(k), ..., x_n(k))^T$ ($k = 1,2,...$ is a discrete time) is fed to the input of the neo-fuzzy neuron, the output of this neuron is determined by both the membership functions $\mu_{ji}(x_i(k))$ and tunable synaptic weights $w_{ji}(k-1)$, which were obtained at the previous training epoch.

$$\hat{y}(k) = \sum_{i=1}^{n} f_i(x_i(k)) = \sum_{i=1}^{n}\sum_{j=1}^{h} w_{ji}(k-1)\mu_{ji}(x_i(k))$$

and thereby neo-fuzzy neuron contains $h \times n$ synaptic weights which should be determined.

The authors of the NFN note [Yamakawa, 1992; Uchino, 1997; Miki, 1999] among its most important advantages, the high rate of learning, computational simplicity, the possibility of finding the global minimum of the learning criterion in real time and also that it is characterized by fuzzy linguistic “if-then” rules.

The learning criterion (goal function) is the standard local quadratic error function:

$$E(k) = \frac{1}{2}\left(y(k) - \hat{y}(k)\right)^2 = \frac{1}{2}e(k)^2 = \frac{1}{2}\left(y(k) - \sum_{i=1}^{n}\sum_{j=1}^{h} w_{ji}\mu_{ji}(x_i(k)) \right)^2$$

minimized via the conventional gradient stepwise algorithm, resulting in the following weight update procedure:

$$w_{ji}(k+1) = w_{ji}(k) + \eta e(k+1)\mu_{ji}(x_i(k+1)) =$$
$$= w_{ji}(k) + \eta\left(y(k+1) - \sum_{i=1}^{n}\sum_{j=1}^{h} w_{ji}(k)\mu_{ji}(x_i(k+1))\right)\mu_{ji}(x_i(k+1))$$

where $y(k)$ is the target value of the output (learning signal), η is the scalar learning rate parameter which determines the speed of convergence and is chosen empirically.

For the purpose of increasing training speed [Bodyanskiy, 2003; Kolodyazhniy, 2005] Kaczmarz-Widrow-Hoff optimal one-step algorithm [Kaczmarz, 1937; Kaczmarz, 1993; Widrow, 1960] can be used in the following form:

$$w(k+1) = w(k) + \frac{y(k+1) - w^T(k)\mu(x(k+1))}{\left\|\mu(x(k+1))\right\|^2}\mu(x(k+1)) \tag{4}$$

where

$\mu(x(k+1)) = \left(\mu_{11}(x_1(k+1)),...,\mu_{h1}(x_1(k+1)),...,\mu_{h2}(x_2(k+1)),...,\mu_{ji}(x_i(k+1)),...,\mu_{hn}(x_n(k+1))\right)^T$,

$w(k) = \left(w_{11}(k),...,w_{h1}(k),...,w_{h2}(k),...,w_{ji}(k),...,w_{hn}(k)\right)^T$ are $(hn)\times 1$ vectors, generated by the corresponding variables. Also exponentially weighted modification of procedure (4) can be used:

$$\begin{cases} w(k+1) = w(k) + r^{-1}(k+1)\left(y(k+1) - w^T(k)\mu(x(k+1))\right)\mu(x(k+1)), \\ r(k+1) = \alpha r(k) + \left\|\mu(x(k+1))\right\|^2, 0 \le \alpha \le 1 \end{cases} \tag{5}$$

which possesses both smoothing and following properties.

In case we have priori defined data set training process can be performed in a batch mode for one epoch using conventional least squares method.

On basis of neo-fuzzy neurons in [Kolodyazhniy, 2004a; Kolodyazhniy, 2004b; Bodyanskiy, 2005a; Bodyanskiy, 2005b; Bodyanskiy, 2005c; Kolodyazhniy, 2006] two-layer feedforward neuro-fuzzy network was synthesized. It possesses improved approximation capabilities in comparison with conventional multilayer perceptron. Given ANN utilized backpropagation for weight adaptation and obviously it results in decreasing rate of convergence in the hidden layer. This circumstance also was a reason for developing cascade neo-fuzzy neural network and improvement of its approximation possibilities.

Cubic-Spline Activation Functions

As stated above conventionally triangular membership functions are used as activation functions in neo-fuzzy neuron. It entails some difficulties during modeling or forecasting processes which are described by smooth functions. At this case piecewise-linear approximation performed by conventional neo-fuzzy neuron can bring us to decreased accuracy of received results. To minimize its negative effect we can increase number of membership functions. But it results in increasing of synaptic weight coefficients quantity and therefore complexity of our architecture is rising as well as time required for its learning.

To avoid this disadvantage cubic-spline membership functions (6) can be used. Commonly they are given in the following form:

$$\mu(x)=\begin{cases}0,\ if\ x\le a,\\ 1-3\left(\frac{x-b}{a-b}\right)^2+2\left(\frac{x-b}{a-b}\right)^3,\ if\ a<x\le b,\\ 1,\ if\ b<x\le c,\\ 1-3\left(\frac{x-c}{d-c}\right)^2+2\left(\frac{x-c}{d-c}\right)^3,\ if\ c<x\le d,\\ 0,\ if\ x>d\end{cases}\qquad (6)$$

and shown in the Fig. 3.

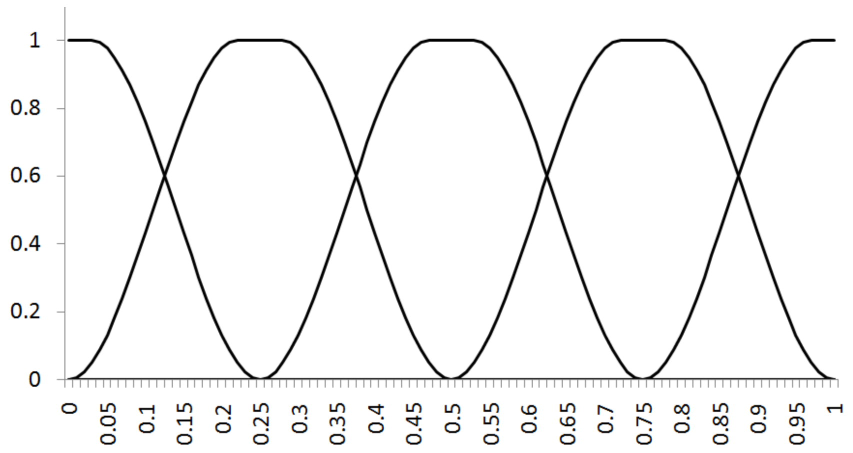


Figure 3. Cubic-spline membership functions (in case $b \ne c$)

It can be readily seen from the Fig. 3 that input signal activates only two neighboring functions simultaneously just like in case with triangular membership functions. But given system of functions doesn't meet the requirements to satisfy the Ruspini partitioning (2). It brings us to impossibility of using Center-Of-Gravity deffuzification method in a simple form (3) and so overall computational complexity is increases. To avoid this imperfection we can use cubic spline membership functions with $b = c$ and also in this case expression (6) can be replaced by another expression (7) which gives completely identical results but its usage is more suitable.

$$\mu(x)=\begin{cases}0.25\left(2+3\frac{2x-x_i-x_{i-1}}{x_i-x_{i-1}}-\left(\frac{2x-x_i-x_{i-1}}{x_i-x_{i-1}}\right)^3\right),\ x\in[x_{i-1},x_i],\\ 0.25\left(2-3\frac{2x-x_{i+1}-x_i}{x_{i+1}-x_i}+\left(\frac{2x-x_{i+1}-x_i}{x_{i+1}-x_i}\right)^3\right),\ x\in(x_i,x_{i+1}].\end{cases}\qquad (7)$$

Proposed cubic spline membership functions shown in Fig. 4.

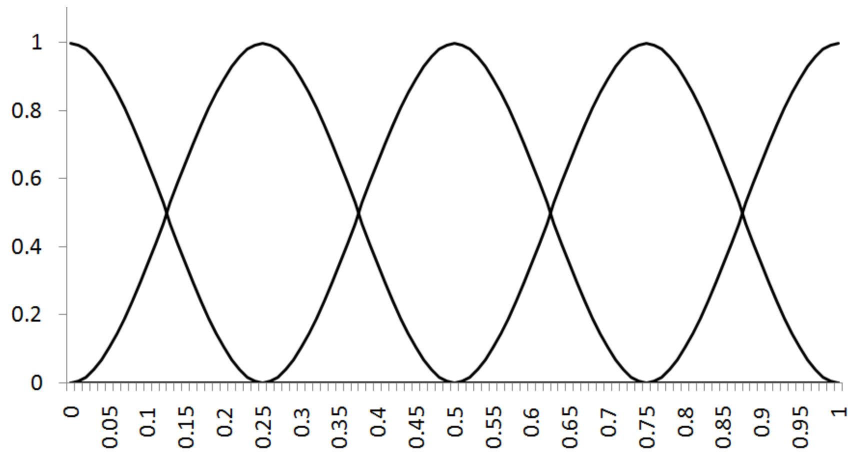


Figure 4. Cubic spline membership functions (in case $b = c$ or in case of notation (7))

Utilizing such recorded membership functions (7) we've got all requirements to meet the Ruspini partitioning (2) and it is considerably contributes to simplify the fuzzy inference process. On the other hand, using of cubic spline activation functions provides smooth polynomial approximation instead of piecewise-linear approximation and makes possible to perform a high quality modeling of significantly nonlinear nonstationary signals and processes.

The Cascade Neo-Fuzzy Architecture Using Cubic-Spline Activation Functions and Its Learning Algorithm

The Cascade Neo-Fuzzy Neural Network architecture shown in Fig.5 and mapping which it realizes has the following form:

- neo-fuzzy neuron of the first cascade

$$\hat{y}^{[1]} = \sum_{i=1}^{n} \sum_{j=1}^{h} w_{ji}^{[1]} \mu_{ji}(x_i),$$

- neo-fuzzy neuron of the second cascade

$$\hat{y}^{[2]} = \sum_{i=1}^{n} \sum_{j=1}^{h} w_{ji}^{[2]} \mu_{ji}(x_i) + \sum_{j=1}^{h} w_{j,n+1}^{[2]} \mu_{j,n+1}(\hat{y}^{[1]}),$$

- neo-fuzzy neuron of the third cascade

$$\hat{y}^{[3]} = \sum_{i=1}^{n} \sum_{j=1}^{h} w_{ji}^{[3]} \mu_{ji}(x_i) + \sum_{j=1}^{h} w_{j,n+1}^{[3]} \mu_{j,n+1}(\hat{y}^{[1]}) + \sum_{j=1}^{h} w_{j,n+2}^{[3]} \mu_{j,n+2}(\hat{y}^{[2]}),$$

- neo-fuzzy neuron of the *m*-th cascade

$$\hat{y}^{[m]} = \sum_{i=1}^{n} \sum_{j=1}^{h} w_{ji}^{[m]} \mu_{ji}(x_i) + \sum_{l=n+1}^{n+m-1} \sum_{j=1}^{h} w_{jl}^{[m]} \mu_{jl}(\hat{y}^{[l-n]}). \quad (8)$$

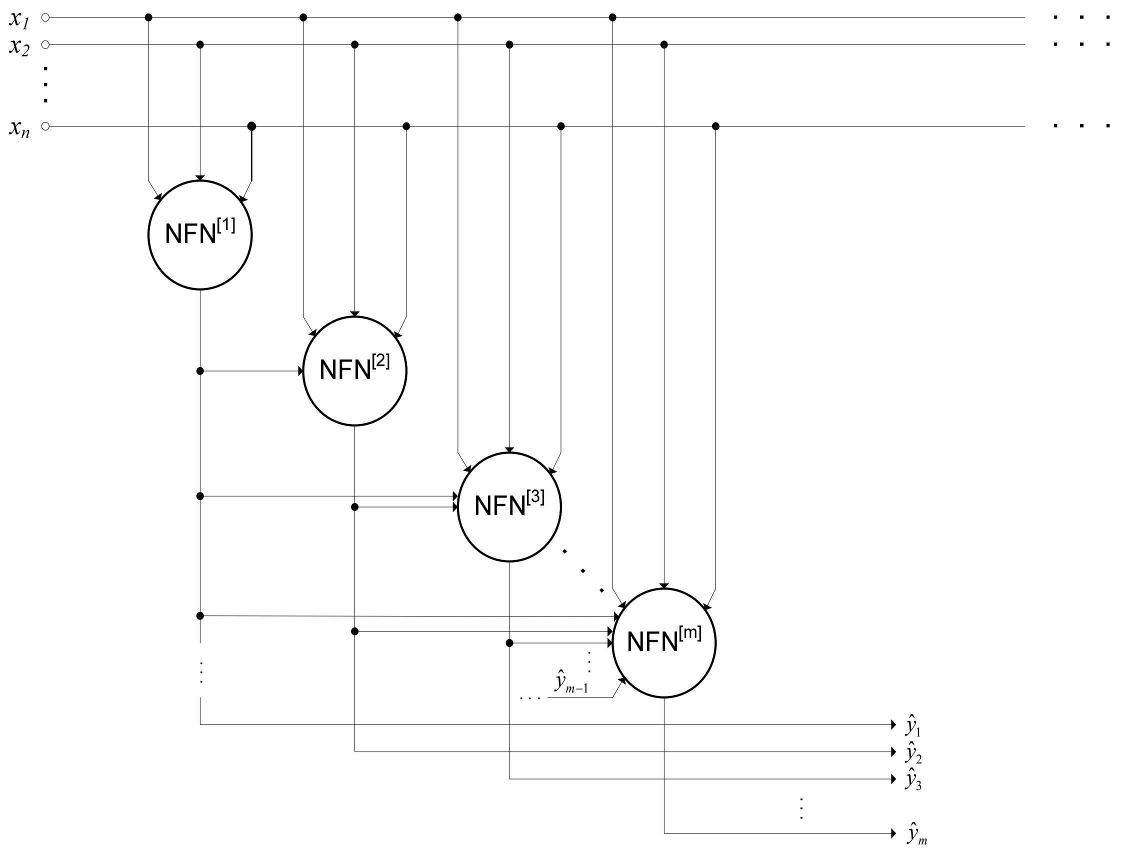


Figure 5. The Cascade Neo-Fuzzy Neural Network Architecture.

Thus cascade neo-fuzzy neural network contains $h\left(n+\sum_{l=1}^{m-1} l\right)$ adjustable parameters and it is important that all of them are linearly included in the definition (8).

Let us define $h(n+m-1)\times 1$ membership functions vector of *m*-th neo-fuzzy neuron $\mu^{[m]} = (\mu_{11}(x_1),...,\mu_{h1}(x_1),\mu_{12}(x_2),...,\mu_{h2}(x_2),...,\mu_{ji}(x_i),...,\mu_{hn}(x_n),\mu_{1,n+1}(\hat{y}^{[1]}),...,\mu_{h,n+1}(\hat{y}^{[1]}),$ $,...,\mu_{h,n+m-1}(\hat{y}^{m-1}))^T$ and corresponding vector of synaptic weights $w^{[m]} =$ $= (w_{11}^{[m]}, w_{21}^{[m]},..., w_{h1}^{[m]}, w_{12}^{[m]},..., w_{h2}^{[m]},..., w_{ji}^{[m]},..., w_{hn}^{[m]}, w_{1,n+1}^{[m]},..., w_{h,n+1}^{[m]},..., w_{h,n+m-1}^{[m]})^T$ which has the same dimensionality. Then we can represent expression (8) in vector notation:

$$\hat{y}^{[m]} = w^{[m]T}\mu^{[m]}.$$

The cascade neo-fuzzy neural network learning can be performed in both batch mode and mode of sequential information processing (adaptive weights tuning).

Firstly, let us examine situation when all training dataset priori defined, i.e. we have a set of points $x(1), y(1); x(2), y(2);...; x(k), y(k);...; x(N), y(N)$. For the neo-fuzzy neuron of the first cascade NFN[1] a set of membership level values $\mu^{[1]}(1), \mu^{[1]}(2),..., \mu^{[1]}(k),..., \mu^{[1]}(N)$ is evaluated, where

$\mu^{[1]}(k) = \left(\mu_{11}(x_1(k)),...,\mu_{h1}(x_1(k)),\mu_{12}(x_2(k)),...,\mu_{h2}(x_2(k)),...,\mu_{ji}(x_i(k)),...,\mu_{hn}(x_n(k))\right)^T$. Then using direct minimization of the learning criterion

$$E_N^{[1]} = \frac{1}{2}\sum_{k=1}^{N}(e^{[1]}(k))^2 = \frac{1}{2}\sum_{k=1}^{N}(y(k) - \hat{y}^{[1]}(k))^2$$

vector of synaptic weights can be evaluated

$$w^{[1]}(N) = \left(\sum_{k=1}^{N}\mu^{[1]}(k)\mu^{[1]T}(k)\right)^{+}\sum_{k=1}^{N}\mu^{[1]}(k)y(k) = P^{[1]}(N)\sum_{k=1}^{N}\mu^{[1]}(k)y(k) \tag{9}$$

where $(\bullet)^+$ - symbol of Moore-Penrose pseudoinversion.

In case data are proceed sequentially more suitable to use a recurrent form of least squares method instead of (9)

$$\begin{cases} w^{[1]}(k+1) = w^{[1]}(k) + \dfrac{P^{[1]}(k)(y(k+1) - w^{[1]T}(k)\mu^{[1]}(k+1))}{1+\mu^{[1]T}(k+1)P^{[1]}(k)\mu^{[1]}(k+1)}\mu^{[1]}(k+1), \\ P^{[1]}(k+1) = P^{[1]}(k) - \dfrac{P^{[1]}(k)\mu^{[1]}(k+1)\mu^{[1]T}(k+1)P^{[1]}(k)}{1+\mu^{[1]T}(k+1)P^{[1]}(k)\mu^{[1]}(k+1)}, P^{[1]}(0) = \beta I \end{cases} \tag{10}$$

where β is a large positive number, I is a unity matrix with corresponding dimensionality.

Usage of adaptive algorithms (4) or (5) is also possible and leads to reducing of computational complexity of learning process. In any case utilization of procedures (4), (5), (9), (10) essentially reduces learning time in comparison with gradient algorithms underlying delta-rule and backpropagation.

After the first cascade learning completion, synaptic weights of the neo-fuzzy neuron NFN$_1$ become ‘frozen’, all values $\hat{y}^{[1]}(1),\hat{y}^{[1]}(2),...,\hat{y}^{[1]}(k),...,\hat{y}^{[1]}(N)$ are evaluated and second cascade of network which consists of a single neo-fuzzy neuron NFN$_2$ is generated. It has one additional input for the output signal of the first cascade. Then procedure (5) again applied for adjusting vector of weight coefficients $w^{[2]}$, which dimensionality is $h(n+1)$.

In serial mode neurons are trained sequentially, i.e. on basis of input signal $x(k)$ synaptic weights $w^{[1]}(k)$ are evaluated and vector of outputs $\hat{y}^{[1]}(k)$ is obtained, then using vector of second cascade inputs $\left(x^T(k),\hat{y}^{[1]}(k)\right)$ weights $w^{[2]}(k)$ and outputs $\hat{y}^{[2]}(k)$ are calculated. For this purpose algorithms (2), (3) and (6) can be used equally well.

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 expressions are used:

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

in a batch mode or

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

or

$$\begin{cases} w^{[m]}(k+1) = w^{[m]}(k) + (r^{[m]}(k+1))^{-1}\left(y(k+1) - w^{[m]T}(k)\mu^{[m]}(k+1)\right)\mu^{[m]}(k+1), \\ r^{[m]}(k+1) = \alpha r^{[m]}(k) + \left\|\mu^{[m]}(k+1)\right\|^2, 0 \le \alpha \le 1 \end{cases}$$

in a serial mode.

Proposed CNFNN significantly excels its prototype – cascade-correlation architecture – in learning speed and can be trained in both batch mode and serial (adaptive) mode. Linguistic interpretation of received results considerably extends functional facilities of cascade neo-fuzzy neural network. Using of cubic spline functions as activation functions in neo-fuzzy neurons significantly increases approximation qualities of network when we have deal with processes which described by highly non-linear signals.

Simulation Results

In order to confirm the efficiency of proposed approach for smooth functions approximation we solved two test problems. First was the 'breast cancer in Wisconsin' benchmark classification problem. It is evident that when we have deal with such problem piecewise-linear separating hyperspace which can be obtained using conventional neo-fuzzy neuron gives almost the same quality of classification like smooth shaped separating hyperspace. So at this case advantage of proposed method wouldn't be significant, what shown below. On the other hand if we model some process which described by the smooth function proposed at this paper technique allows considerably reduce quantity of weight coefficients and therefore overall computational complexity of neuro-fuzzy architecture and also increase quality of received results. So the second test problem was the dynamic object identification problem.

Let us have a look at the first test problem – 'breast cancer in Wisconsin' benchmark classification problem. We used dataset which contained 699 points and can be found at the address ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/cancer/cancer1/datacum. 16 points had parameters with missed values so they were eliminated from the dataset and remaining 683 points were 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 has examined patient. Feature values were normalized on interval [0; 1]. Normalization procedure preceding synaptic weights adjustment process is very important when we train the Cascade Neo-Fuzzy Neural Network because used set of cubic spline activation functions gives zero output if input signal lies outside specified interval. Also normalization procedure is necessary between cascades.

For comparison two architectures were trained and simulated. First of them used conventional neo-fuzzy neurons with triangular activation functions and the second one utilized neo-fuzzy neurons with cubic spline activation functions. Both architectures consist of 3 cascades and each cascade consists of a single neo-fuzzy neuron with 5 activation functions. For synaptic weight coefficients adaptation in each cascade least squares method was utilized. 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 Cascade Neo-Fuzzy Neural Networks with different types of neo-fuzzy neurons.

Type of neo-fuzzy neuron	Correctly classified examples	Incorrectly classified examples	Outside the ‘belief zone’
Conventional NFN	197	6	2
NFN with cubic spline activation functions	197	3	5

From the table we can see that Cascade Neo-Fuzzy Neural Network which utilizes neo-fuzzy neurons with cubic spline activation functions gives slightly better results than architecture with triangular activation functions. As it was already stated above difference between two examined architectures at this case isn’t considerable. But let us further pay attention on the class of problems which can be solved better and faster using proposed at this paper cascade architecture.

The second test problem was the dynamic plant identification problem. Proposed dynamic plant [Patra, 2002; Narendra, 1990] can be defined by following equation:

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

where

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

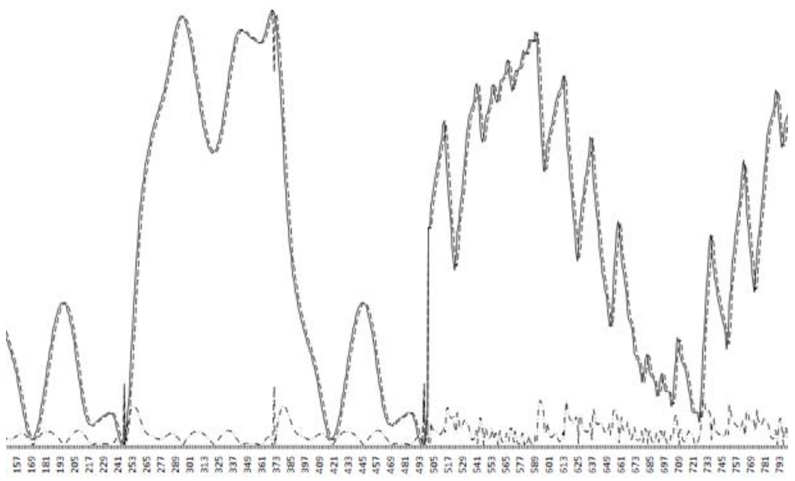
There was generated a sequence which contained 1500 values of signal for *k*=1,2,…,1500. On training set signal $u(k) = \sin 2k/250$ (*k*=1,…,500) have been used and on the testing set $u(k) = \sin 2k/250$ (*k*=501,…,1000), $u(k) = 0.5\sin 2k/250 + 0.5\sin 2k/25$ (*k*=1001,…1500). It means that on testing set sinusoidal component of the dynamic plant is changing and therefore output signal changes its form too. Obtained set was normalized on interval [0, 1] because of reasons mentioned above.

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

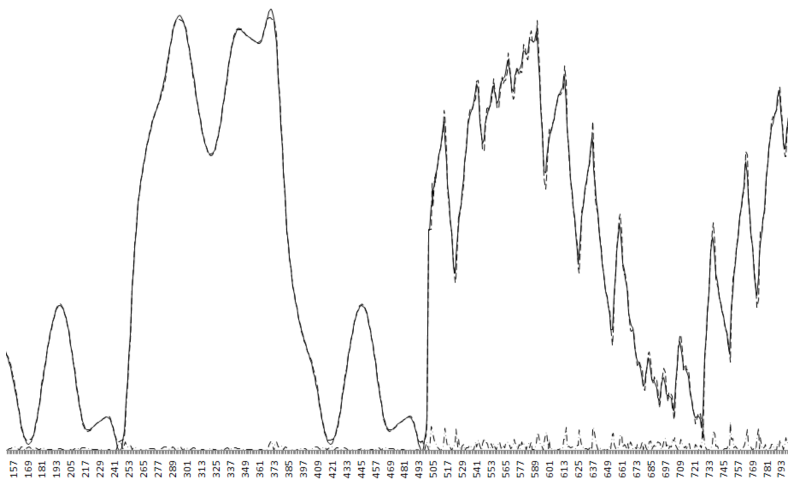
$$NRMSE(k,N) = \frac{\sum_{q=1}^{N} e^2(k+q)}{N\sigma}$$

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

Three architectures were trained and simulated. First architecture used conventional neo-fuzzy neuron with 6 activation functions in each cascade and had eight cascades. Second and third architectures were the same except the type of activation functions. One of them used triangular functions and other cubic spline functions. Each nonlinear synapse contained 4 activation functions and quantity of cascades for both architectures was 4. For adjusting weight coefficients in all cases LSM was used. Obtained results are given in table 2 and shown in Fig. 6 (dynamic object identification results using second and third architecture are shown).



a)



b)

Figure 2. Dynamic plant identification using
the Cascade Neo-Fuzzy Neural Network with a) conventional neo-fuzzy neurons;
b) neo-fuzzy neurons with cubic spline activation functions in non-linear synapses;
normalized plant output – solid line; network output – dashed line; identification error – chain line.

Table 2. Results of the dynamic plant identification.

Type of neo-fuzzy neuron in the cascade architecture	Quantity of activation functions	Quantity of cascades	Total number of adjustable parameters in architecture	NRMSE
Conventional NFN	6	8	102	0.0008
Conventional NFN	2	4	16	0.0063
NFN with cubic spline activation functions	2	4	16	0.0005

As it can be seen from the table 2 usage of neo-fuzzy neurons with cubic spline activation functions provides decreasing of architecture complexity as well as increasing of obtained results quality. In concerned test problem, when we applied two identical architectures, obtained error for that which utilized conventional neo-fuzzy neurons was appreciably greater and therefore quality of output signal was noticeably worse, what can be seen in Fig. 6. To obtain quality of output signal comparable with provided by Cascade Neo-Fuzzy Neural Network with cubic spline activation functions, complexity of architecture which utilized conventional neo-fuzzy neurons was considerably increased. And approximately identical result was obtained when total number of adjustable parameters was about six times greater than in architecture with cubic spline activation functions.

Conclusion

The Cascade Neo-Fuzzy Neural Network which utilized neo-fuzzy neurons with cubic spline activation functions is proposed. It differs from the known cascade networks in increasing of operation speed, real-time processing possibility and transparency due to linguistic interpretability of received results. Used type of activation functions allows to increase accuracy of smooth functions approximation, reduce time required for adjusting weight coefficients and decrease overall architecture complexity. Theoretical justification and experiment results confirm the efficiency of developed approach to cascade neo-fuzzy 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, 2007a] 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, 2004a] Bodyanskiy Ye., Kolodyazhniy V., Slipchenko O. Artificial neural network with orthogonal activation functions for dynamic system identification. Synergies between Information Processing and Automation. Eds. O. Sawodny and P. Scharff. Aachen: Shaker Verlag, 2004. P. 24-30.

[Bodyanskiy, 2004b] 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.

[Bodyanskiy, 2006a] Bodyanskiy Ye., Pliss I., Slipchenko O. Growing neural networks based on orthogonal activation functions. Proc. XII-th Int. Conf. “Knowledge–Dialog–Solution”. Varna, 2006. P. 84-89.

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

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

[Bodyanskiy, 2008a] Bodyanskiy Ye., Dolotov A., Pliss I., Viktorov Ye. The cascade orthogonal neural network. Advanced Research in Artificial Intelligence. Bulgaria, Sofia: Institute of Informational Theories and Applications FOI ITHEA, 2. 2008. P. 13-20.

[Viktorov, 2008] Viktorov Ye., Bodyanskiy Ye., Dolotov A. Solving approximation and forecasting problems using double ortho-neuron. Komp’juterni nauky ta informacijni tekhnologhiji, Issue 598. 2008. P. 70-77.

[Jang, 1997] Jang Jr. S. R., Sun C. T., Mizutani E. Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence. N.J.: Prentice Hall, 1997.

[Bodyanskiy, 2008b] Bodyanskiy Ye., Viktorov Ye., Pliss I. The cascade neo-fuzzy neural network and its learning algorithm. Visnyk Uzhghorods’kogho Nacional’nogho universytetu. Serija “Matematyka i informatyka”, Issue 17. 2008. P. 48-58.

[Yamakawa, 1992] 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. of 2-nd Int.Conf. on Fuzzy Logic and Neural Networks “IIZUKA-92”. Iizuka, Japan, 1992. P. 477-483.

[Uchino, 1997] 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.

[Miki, 1999] 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.

[Bodyanskiy, 2003] Bodyanskiy Ye., Kokshenev I., Kolodyazhniy V. An adaptive learning algorithm for a neo-fuzzy neuron. Proc. 3rd Int. Conf. of European Union Soc. for Fuzzy Logic and Technology (EUSFLAT 2003). Zittau, Germany, 2003. P. 375-379.

[Kolodyazhniy, 2005] Kolodyazhniy V., Bodyanskiy Ye., Otto P. Universal approximator employing neo-fuzzy neurons. In: Computational Intelligence: Theory and Applications. Ed. by B. Reusch. Berlin-Heidelberg: Springer, 2005. P. 631-640.

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

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

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

[Kolodyazhniy, 2004a] Kolodyazhniy V., Bodyanskiy Ye. Fuzzy neural network with Kolmogorov’s structure. Proc. East West Fuzzy Coll. Zittau/Gorlitz: HS, 2004. P. 139-146.

[Kolodyazhniy, 2004b] Kolodyazhniy V., Bodyanskiy Ye. Fuzzy Kolmogorov’s network. In: Lecture Notes in Computer Science. Heidelberg: Springer Verlag, V.3214, 2004. P. 764-771.

[Bodyanskiy, 2005a] Bodyanskiy Ye., Gorshkov Ye., Kolodyazhniy V. Neuro-fuzzy Kolmogorov’s network with a hybid learning algorithm. Proc. XI-th Int. Conf. “Knowledge-Dialog-Solution”, V.2. Varna, Bulgaria, 2005. P. 622-627.

[Bodyanskiy, 2005b] Bodyanskiy Ye., Kolodyazhniy V., Otto P. Neuro-fuzzy Kolmogorov’s network for time-series prediction and pattern classification. In: Lecture Notes in Artificial Intelligence, V.3698. Heidelberg: Springer Verlag, 2005. P. 191-202.

[Bodyanskiy, 2005c] Bodyanskiy Ye., Gorshkov Ye., Kolodyazhniy V., Poyedintseva V. Neuro-fuzzy Kolmogorov's network. In: Lecture Notes in Computer Science, V.3697. Berlin-Heidelberg: Springer Verlag, 2005. P. 1-6.

[Kolodyazhniy, 2006] Kolodyazhniy V., Bodyanskiy Ye., Poedintseva V., Stephan A. Neuro-fuzzy Kolmogorov's network with a modified perceptron learning rule for classification problems. In: Computational Intelligence: Theory and Applications. Ed. by B. Reusch. Berlin-Heidelberg: Springer-Verlag, 2006. P. 41-49.

[Patra, 2002] Patra J.C., Kot A.C. Nonlinear dynamic system identification using Chebyshev functional link artificial neural network. IEEE Trans. on Systems, Man, and Cybernetics, 4, Part B. 2002. P. 505-511.

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

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.*

Major Fields of Scientific Research: Hybrid Systems of Computational Intelligence

Yevgen Viktorov *– PhD Student of Artificial Intelligence Department; Kharkiv National University of Radio Electronics, Lenina av., 14, Kharkiv, 6116, Ukraine. Tel: +380686163429, e-mail: yevgen.viktorov@gmail.com.*

Major Fields of Scientific Research: Artificial Neural Networks: Nonconventional Cascade and Multilayer Architectures; Fuzzy Logic; Neuro-Fuzzy Hybrid Systems