

143/2004

Raport Badawczy
Research Report

RB/56/2004

**Przykłady zastosowań
wybranych metod uczenia
maszynowego.**

**K.Mańczak, G.Szkatuła,
M.Krawczak**

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences



POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 8373578

fax: (+48) (22) 8372772

Kierownik Pracowni zgłaszający pracę:
Prof. dr hab. inż. Janusz Kacprzyk

Warszawa 2004

Przykłady zastosowań wybranych metod uczenia maszynowego

Contents

On a new way to gradient learning algorithms for neural networks	3
Analysis of rules describing electorate preferences generated by machine learning methods	19

ON A NEW WAY TO GRADIENT LEARNING ALGORITHMS FOR NEURAL NETWORKS

1. Introduction

In this paper we will recall the concept of adjoint neural networks that was described by Krawczak (2002, 2003) as a method for evolution the formulae of the backpropagation learning algorithm for multilayer neural networks. The methodology is based on the assumption that multilayer neural networks can be treated as *flow graphs*. The analysis leads to the proper equations of the backpropagation algorithm but in a much simpler manner. The graph methodology incorporates the *reciprocal graphs* in which signals flow in opposite directions. These kind of neural networks are called the *adjoint neural networks*. Construction of the adjoint neural networks yields directly the formulae of the considered learning algorithm, as well as to any gradient descent based algorithms.

Using the methodology for modelling complex systems, called the Generalized Net theory, developed by Atanassov (1991), we will construct such a model of the adjoint neural networks. In a paper by Krawczak and Aladjov (2002) we developed the Generalized Net model of the backpropagation algorithm as well as the first approach to development of the Generalized Net model of the adjoint neural networks.

2. Transformation of a Neural Network into a Graph

Since the pioneering work of McCulloch and Pitts (1943) a model of an artificial neuron is a very simple processing unit, Figure 1, which has a number of inputs x_i , say N , each input being weighted with an appropriate weight $E_p = \frac{1}{2} \sum_{j^{(L)}=1}^{N^{(L)}} (d_{j^{(L)}p} - x_{j^{(L)}p})^2$, $i = 1, 2, \dots, N$.

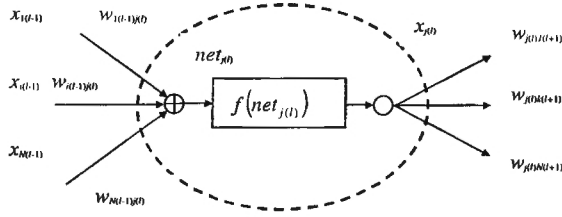


Figure 1: An elementary model of a neuron

The sum of the weighted inputs and the bias (included in the inputs) forms at the summation point \oplus the proper input

$$net_j = \sum_{i=1}^N w_{ij} x_i = \sum_{i=1}^N y_{ij} \quad (1)$$

to the activation function $f_j(net_j)$. In the model considered an additional element corresponding to a junction point, which is depicted by \circ , is included. Figure 1 shows an extended notation of indices, namely we indicate the position of each neuron in the whole network. For example the weight $w_{i(l-1)j(l)}$ indicates the connection between the neuron i belonging to the $(l-1)$ -st layer and the neuron j from the l -th layer.

Let us rearrange the neuron's elements in the following way:

- remove the activation function to the outside of the neuron,
- the removed activation functions are shifted to each of the connections between the considered neuron and all neurons of the next layer, becoming thereby the transmittances between neurons,
- the connection between neurons are still weighted,
- the summation point and the junction point make up a *node*,
- the neural network with the rearranged neurons becomes a *flow graph*.

The above rearrangement is pictured in Figure 2.

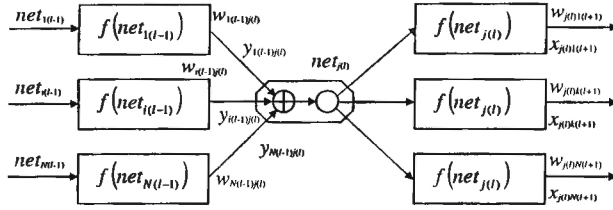


Figure 2: The rearranged neuron as a segment of a flow graph

Now, description of any separate edge takes the following form:

$$y_{i(l-1)j(l)} = w_{i(l-1)j(l)} x_{i(l-1)} = w_{i(l-1)j(l)} f_i(l-1)(net_{i(l-1)}). \quad (2)$$

The neural network with the new models of neurons becomes a *flow graph*. Comparing the neural network as a flow graph to the respective adjoint neural network we can notice that the architectures exactly the same (Krawczak, 2002).

Using the graph theory notation a feedforward network topology can be specified by considering the values of nodes, i.e. $net_{j(l)}$, for $l = 1, 2, \dots, L$, $j(l) = 1, 2, \dots, N(l)$

$$net_{j(l)} = \begin{cases} net_{i(0)} = x_{i(0)}, \text{ for } l = 0 \\ \sum_{i(0)=1}^{N(0)} w_{i(0)j(l)} x_{i(0)}, \text{ for } l = 1 \\ \sum_{i(l-1)=1}^{N(l-1)} w_{i(l-1)j(l)} f_i(l-1)(net_{i(l-1)}), \text{ for } 1 < l < L \\ net_{j(N(L))} = \sum_{i(L)=1}^{N(L)} w_{i(L-1)j(L)} f_i(L-1)(net_{i(L-1)}), \text{ for } l = L \end{cases} \quad (3)$$

Using the formula for $net_{j(l)}$ we can illustrate flows of signals in a neural network treated as a flow graph in Figure 3.

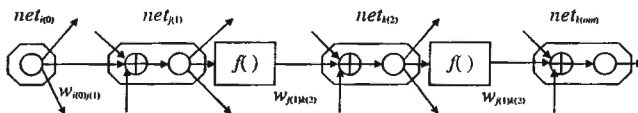


Figure 3: A schematic exemplary two-layer neural network as a flow graph

The generalized delta rule for updating the weights has the following form

$$\Delta w_{i(l-1)j(l)} = -\eta \frac{\partial E_p}{\partial w_{i(l-1)j(l)}} = -\eta \frac{\partial E_p}{\partial net_{j(l)}} \frac{\partial net_{j(l)}}{\partial w_{i(l-1)j(l)}} = \eta \delta_{j(l)} x_{i(l-1)} \quad (4)$$

for $j(l) = 1, 2, \dots, N(l)$, $i(l-1) = 1, 2, \dots, N(l-1)$, $l = 1, 2, \dots, L$, where the factor delta is expressed bellow

$$\delta_{i(l-1)} = \begin{cases} \sum_{j(l)=1}^{N(l)} f'(net_{j(l)}) \delta_{j(l)} & , \text{if } l=L \\ f'(net_{i(l-1)}) \sum_{j(l)=1}^{N(l)} w_{i(l-1)j(l)} \delta_{j(l)} & , \text{if } 2 \leq l \leq L-1 \\ \delta_{i(0)} = \delta_{i(n)} = \sum_{j(0)=1}^{N(0)} w_{i(0)j(0)} \delta_{j(0)} & , \text{if } l=1 \end{cases} \quad (5)$$

where

$$f'_{j(l)} = -\frac{\partial E}{\partial net_{j(l)}} \frac{\partial f_{j(l)} \left(\sum_{i=1}^n w_{i(l-1)j(l)} x_{i(l-1)} \right)}{\partial net_{j(l)}} \quad (6)$$

It can be easily noticed that equations describing the signals net and δ have the same structure, and directions of these signals are opposite. In Figure 4 we have presented a counterpart to the example shown in Figure 3.

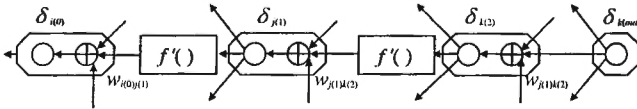


Figure 4: An exemplary two-layer neural network as an adjoint NN

The adjoint neural network was found by application of the same network architecture, reversal of the direction of signal flows, replacement of activation functions by their derivatives, and switching of the positions of summing points with junction points within each node. It was shown that the transformation of the original network into the adjoint network is governed by very simple rules described in Krawczak (2002).

3. Generalized Net Modelling

The basic difference between Generalized Nets and the ordinary Petri nets is the *place – transition relation* (Atanassov, 1991), in the theory of Generalized Nets the transitions are objects of a very complex nature. The places are marked by \bigcirc , and the transitions by \lrcorner . Generalized Nets contain *tokens*, which are transferred from place to place. Every token bears some information, which is described by token's *characteristic*, and any token enters the net with an *initial characteristic*. After passing a transition the tokens' characteristics are modified.

The transition has *input* and *output* places, as shown in Figure 5.

Formally, every transition is described by a seven-tuple

$$Z = \langle L', L'', t_1, t_2, r, M, \square \rangle \quad (7)$$

where:

- $L' = \{l'_1, l'_2, \dots, l'_n\}$ is a finite, non empty set of the transition's input places,
- $L'' = \{l''_1, l''_2, \dots, l''_m\}$ is a finite, non empty set of the transition's output places,
- t_1 is the current time of the transition's firing,
- t_2 is the current duration of the transition active state,
- r is the transition's *condition* determining which tokens will pass (or *transfer*) from the transition's inputs to its outputs; it has the form of an *index matrix* described in (Atanassov, 1987),

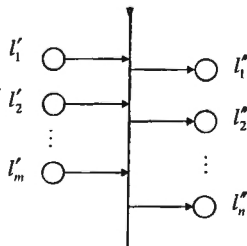


Figure 5: A Generalized Net transition

- M is an index matrix of the capacities of transition's arcs,

\square is an object of a form similar to a Boolean expression, it may contain as variables the symbols that serve as labels for transition's input places, and \square is an expression built up from variables and the Boolean connectives \wedge and \vee , The following ordered four-tuple

$$E = \langle \langle A, \pi_A, \pi_L, c, f, \Theta_1, \Theta_2 \rangle, \langle K, \pi_k, \Theta_K \rangle, \langle T, t^0, t^* \rangle, \langle X, \Phi, b \rangle \rangle \quad (8)$$

is called *Generalized Net* if the elements are described as follows:

- A is a set of transitions,
- π_A is a function yielding the priorities of the transitions, i.e. $\pi_A: A \rightarrow N$, where $N = \{0, 1, 2, \dots\} \cup \{\infty\}$,
- π_L is a function specifying the priorities of the places, i.e. $\pi_L: L \rightarrow N$, where $L = pr_1A \cup pr_2A$, and pr_iX is the i -th projection of the n -dimensional set, where $n \in N$, $n \geq 1$ and $1 \leq i \leq n$ (obviously, L is the set of all Generalized Net places),
- c is a function providing the capacities of the places, i.e. $c: L \rightarrow N$,
- f is a function that calculates the truth values of the predicates of the transition's conditions (for the Generalized Net described here let the function f have the value *false* or *true*, i.e. a value from the set $\{0, 1\}$),
- Θ_1 is a function specifying the next time-moment when a given transition Z can be activated, i.e. $\Theta_1(t) = t'$, where $pr_3Z = t$, $t' \in [T, T + t^*]$ and $t \leq t'$; the value of this function is calculated at the moment when the transition terminates its functioning,
- Θ_2 is a function yielding the duration of the active state of a given transition Z , i.e. $\Theta_2(t) = t'$, where $pr_4Z = t \in [T, T + t^*]$ and $t' \geq 0$; the value of this function is calculated at the moment when the transition starts its functioning,
- K is the set of the Generalized Net's tokens,

- π_K is a function specifying the priorities of the tokens, i.e. $\pi_K : K \rightarrow N$,
- Θ_K is a function producing the time-moment when a given token can enter the net, i.e. $\Theta_K(\alpha) = t$, where $\alpha \in K$ and $t \in [T, T + t^*]$,
- T is the time-moment when the Generalized Net starts functioning; this moment is determined with respect to a fixed (global) time-scale,
- t^0 is an elementary time-step, related to the fixed (global) time-scale,
- t^* is the duration of the Generalized Net functioning,
- X is the set of all initial characteristics the tokens can receive on entering the net,
- Φ is a characteristic function that assigns new characteristics to every token when it makes the transfer from an input to an output place of a given transition,
- b is a function specifying the maximum number of characteristics a given token can receive, i.e. $b : K \rightarrow N$; for example, if $b(\alpha) = 1$ for some token α , then this token will enter the net with some initial characteristic and subsequently it will keep only its current characteristic; when $b(\alpha) = \infty$ the token α will keep all its characteristics; when $b(\alpha) = k < \infty$ the token α will keep its last k characteristics (the characteristics older than the last k will be forgotten); in the general case every token α has $b(\alpha) + 1$ characteristics on leaving the net.

A given Generalized Net may lack some of the above components. In these cases, any missing component will be omitted. The Generalized Nets of this kind form a special class of Generalized Nets called *reduced Generalized Nets*.

4. Modelling of the Adjoint Neural Networks

In this part we will determine the Generalized Net model of the adjoint neural network for the backpropagation algorithm, partially following Krawczak, Aladjov (2002).

The Generalized Net model of the adjoint neural networks contains four transitions, see Figure 6. Each transition of the model represents a separate stage of the adjoint neural network functioning. These stages are as follows:

- construction of the adjoint neural network
- initialisation of connection weights
- propagation of signals
- back propagation of the error.

Here we will constrain our consideration to some elements of the reduced Generalized Net form, in order to show how this methodology can be used to construct the model.

In the considered multilayer neural network each neuron is represented by a single neuron of α -type, and the token has the following initial characteristic

$$y(\alpha_l) = \langle NN1, l, I, f_l, x_l, imW, d \rangle, \text{ for } l = 0, 1, \dots, L \quad (9)$$

where $NN1$ - the neural network identifier,

l - the layer number,

I - the number of the token (neuron),

f_l - an activation function of the l -th neuron,

x_l - the current value of neuron output,

imW - the index matrix of the weights, which contains the connection, having the following form

	<i>in</i>	<i>out</i>
1	$W_{1,1}$	$W_{1,1}$
2	$W_{2,1}$	$W_{1,2}$
⋮	⋮	⋮
N	$W_{N,1}$	$W_{1,N}$

where

N - the number of all neurons in the considered neural network,

$W_{m,n}$ - the weight connecting the m -th neuron with the n -th neuron,

d - description, which can be defined as follows

$$d = \begin{cases} \text{"in"} & \text{if } (\forall i \in (1, 2, \dots, N)) (imW_{i,j} = 0) \ \& \ (\exists i \in (1, 2, \dots, N)) (imW_{i,j} \neq 0) \\ \text{"out"} & \text{if } (\forall i \in (1, 2, \dots, N)) (imW_{i,j} = 0) \ \& \ (\exists i \in (1, 2, \dots, N)) (imW_{i,j} \neq 0) \\ \text{"int"} & \text{if } (\exists i \in (1, 2, \dots, N)) (imW_{i,j} \neq 0) \ \& \ (\exists i \in (1, 2, \dots, N)) (imW_{i,j} \neq 0) \\ \text{"iso"} & \text{if } (\forall i \in (1, 2, \dots, N)) (imW_{i,j} = 0) \ \& \ (\forall i \in (1, 2, \dots, N)) (imW_{i,j} = 0) \end{cases} \quad (10)$$

where "in", "out", "int", "iso" denote the input, output, internal and isolated neurons, respectively. In Figure 6 the considered places are denoted by $\ddot{X}_i, i = 1, \dots, 8$, $m_j, j = 1, \dots, 7$, $n_k, k = 1, \dots, 7$.

It is worth noticing that the characteristic (9) includes all information required to estimate the whole neural network, namely the connectivity and characteristics.

The process of the adjoint neural network construction is based on changes of the neurons features, that is - the neurons must be able to propagate the signals in the forward direction as well as to propagate the error in the back direction, and to possess all the information required for the connection weights evaluation. These changes of the new neuron features are represented by generation of the new characteristics of the tokens in the place \ddot{X}_2 , which are as follows

$$y(\alpha_i) = \langle NN1, l, l, f_i, f'_i, x_i, \delta_i, imW, imW_1, d, d_1 \rangle \quad (11)$$

where the new components of the characteristics have the following meaning

$$f'_i(net_i) = \frac{\partial f_i(net_i)}{\partial net_i} \quad (12)$$

where net_i is related to Equation 3, and

$$\delta_i = - \frac{\partial E}{\partial net_i} \quad (13)$$

which is related to Equation 5,

imW_1 has the same components as imW but the weights of inputs are replaced by the weights of outputs and vice versa, that is

		<i>in</i>	<i>out</i>
$imW_1 =$	1	$W_{1,1}$	$W_{1,l}$
	2	$W_{1,2}$	$W_{2,l}$
	\vdots	\vdots	\vdots
	N	$W_{1,N}$	$W_{N,l}$

d_1 describes the connectivity of the neurons within the adjoint neural network, and can be obtained from d in the following way

$$d_1 = \begin{cases} \text{"in"} & \text{if } d = \text{"out"} \\ \text{"out"} & \text{if } d = \text{"in"} \\ \text{"int"} & \text{if } d = \text{"int"} \\ \text{"iso"} & \text{if } d = \text{"iso"} \end{cases} \quad (14)$$

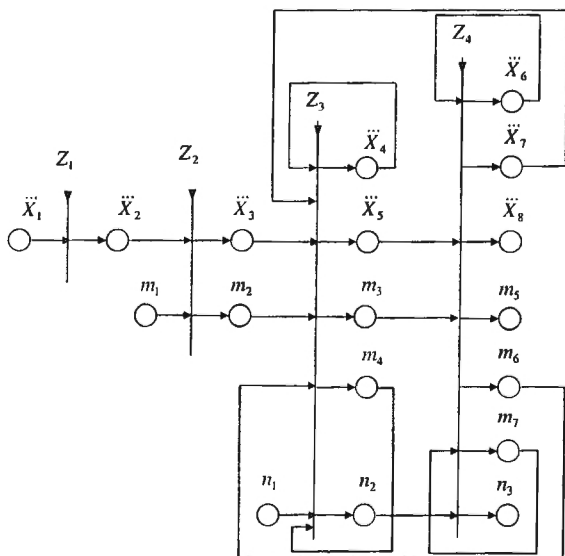


Figure 6: The Generalized Net model of the adjoint neural network

From the Generalized Net point of view, the process of the adjoint neural network model can be represented by the following transition Z_1

$$Z_1 = \langle \{\ddot{X}_1\}, \{\ddot{X}_2\}, \begin{array}{c|c} & \ddot{X}_2 \\ \hline \ddot{X}_1 & true \end{array} \rangle. \quad (15)$$

The next transition Z_2 describes the first stage of the training process, namely the initialisation of weights

$$Z_2 = \langle \{\ddot{X}_2, m_1\}, \{\ddot{X}_3, m_2\}, \begin{array}{c|cc} & \ddot{X}_3 & m_2 \\ \hline \ddot{X}_2 & true & false \\ m_1 & false & true \end{array} \rangle. \quad (16)$$

In the place m_1 a token associated with the performance index enters the Generalized Net with the following initial characteristic

$$y(\beta) = \langle NN1, E, E_{max} \rangle \quad (17)$$

where $NN1$ - the neural network identifier,

E - the performance index of the neural network learning,

E_{max} - the threshold value of the performance index, which must be reached.

In the place m_2 this token does not change this characteristic.

The transition Z_3 associated with the pattern recognition process has the form

$$Z_3 = \langle \{\ddot{X}_3, \ddot{X}_4, \ddot{X}_7, m_2, m_6, n_1\}, \{\ddot{X}_4, \ddot{X}_5, m_3, m_4, n_2\}, \quad (18)$$

	\ddot{X}_5	\ddot{X}_6	m_3	m_4	n_2
\ddot{X}_3	V_1	false	false	false	false
\ddot{X}_4	false	V_2	false	false	false
\ddot{X}_7	V_1	false	false	false	false
m_2	false	false	V_3	$\neg V_3$	false
m_4	false	false	V_3	$\neg V_3$	false
m_6	false	false	V_3	$\neg V_3$	false
n_1	false	false	false	false	V_3

where

V_1 = "the neuron has assigned the input signal",

V_2 = "the neuron has assigned the output signal",

V_3 = "all neurons of the network have assigned the output signals".

The tokens γ_p , $p = 1, 2, \dots, P$, where p is the number of the training pattern enter the place n_1 with the initial characteristic

$$y(\gamma_p) = \langle X_p(0), D_p, p \rangle \quad (19)$$

where $X_p(0)$ - is the inputs vector of the neural network, and D_p - is the vector of desired network outputs.

The tokens of α -type enter the place \ddot{X}_4 for the purpose of calculation of the neuron outputs, and next the tokens are transferred to the place \ddot{X}_5 , where the tokens α form the output layer (for d = "out"), and obtain the new characteristics in the following form

$$y(\alpha_i) = \langle N, N1, l, I, p, f_i, f'_i, x_i, \bar{\delta}_i, im\bar{W}, im\bar{W}_1, d, d_1 \rangle \quad (20)$$

related to the nominal values of connection weights and the pattern p . The factor $\bar{\delta}_i$ can be viewed as representing the inputs for the adjoint neural network.

In the place m_3 the token β obtains the new characteristic

$$y(\beta) = \langle NNL, E', E_{\max} \rangle. \quad (21)$$

The next transition Z_4 is responsible for the error propagation via the adjoint neural network, and the weights correction process, and has the form

$$Z_4 = \langle \{\ddot{X}_5, \ddot{X}_6, m_3, n_2, m_7\}, \{\ddot{X}_6, \ddot{X}_7, \ddot{X}_8, m_5, m_6, m_7, n_3\} \rangle \quad (22)$$

	\ddot{X}_6	\ddot{X}_7	\ddot{X}_8	m_5	m_6	m_7	n_3
\ddot{X}_5	V_4	<i>fals</i>	<i>fals</i>	<i>fals</i>	<i>false</i>	<i>fals</i>	<i>fals</i>
		<i>e</i>	<i>e</i>	<i>e</i>		<i>e</i>	<i>e</i>
$r_4 = \ddot{X}_6$	<i>fals</i>	V_5	$\neg V_5$	<i>fals</i>	<i>false</i>	<i>fals</i>	<i>fals</i>
	<i>e</i>			<i>e</i>		<i>e</i>	<i>e</i>
m_3	<i>fals</i>	<i>fals</i>	<i>fals</i>	$\neg V_5$	$V_5 \& \neg V_6$	V_6	<i>fals</i>
	<i>e</i>	<i>e</i>	<i>e</i>				<i>e</i>
m_7	<i>fals</i>	<i>fals</i>	<i>fals</i>	$\neg V_5$	$V_5 \& \neg V_6$	V_6	<i>fals</i>
	<i>e</i>	<i>e</i>	<i>e</i>				<i>e</i>
n_2	<i>fals</i>	<i>fals</i>	<i>fals</i>	<i>fals</i>	<i>false</i>	<i>fals</i>	<i>true</i>
	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>		<i>e</i>	

where

V_4 = "the adjoint neuron has assigned the input signal",

V_5 = "the new pattern for learning must be applied",

V_6 = "there exist adjoint neurons with unassigned outputs".

In the place \ddot{X}_s the new values of the parameters δ_i are assigned to the adjoint neuron outputs, and the weights can be corrected, in details according to the following rule

$$\Delta w_{i(j)(l)} = \eta \delta_{j(l)} x_{i(l-1)}.$$

Other details of the transitions, places and tokens are very similar to those considered in the previous sections and we will not repeat them.

5. Conclusions

The process of the backpropagation learning algorithm of the multilayer neural networks can be simplified by introducing the adjoint neurons. Such neurons enrich the ordinary neuron capabilities with some mechanism for error backpropagation and self-modification of the connection weight changes. It seems that the new structure of the neurons (the neuron combined with the adjoint neuron) gives the possibilities for hardware implementation of the neural network including the mechanism of the backpropagation learning algorithm or other learning algorithms based on gradient descent.

6. References

- [1] Atanassov, K. (1984) On the Concept "Generalized Net, *AMSE Review*, 1, 3, 39-48.
- [2] Atanassov, K., (1987) Generalized Index Matrices. *Comptes Rendus de l'Academie Bulgare des Sciences*, 40, 11, 15-18.
- [3] Atanassov, K., (1991) *Generalized Nets*. World Scientific, Singapore, New Jersey, London.
- [4] Atanassov, K. (Ed.) (1993) *Applications of Generalized Net*. World Scientific, Singapore, New Jersey, London.

- [5] Atanassov, K., (1997) *Generalized Nets and Systems Theory*. „Prof. M. Drinov" Academic Publishing House, Sofia.
- [6] Atanassov, K., (1998) *Generalized Nets in Artificial Intelligence*. Vol. 1: *Generalized nets and Expert Systems*. „Prof. M. Drinov" Academic Publishing House, Sofia.
- [7] Atanassov, K., Aladjov, H. (2000) *Generalized Nets in Artificial Intelligence*. Vol. 2: *Generalized nets and Machine Learning*. "Prof. M. Drinov" Academic Publishing House, Sofia.
- [8] Christofides, N., (1975) *Graph Theory*. Academic Press, New York.
- [9] Krawczak, M. (2000) Backpropagation versus Dynamic Programming Approach. *Bulletin of Polish Academy of Sciences*, 48, 2, 167-180.
- [10] Krawczak, M. (2002) Adjoint Multilayer Neural Networks. In: *Automatic Control and Management*, J. Gutenbaum (ed.). Polish Academy of Sciences, Systems Research, Special Issue, 153-166.
- [11] Krawczak, M. (2003) *Multilayer Neural Systems and Generalized Net Models*. Academic Press EXIT, Warsaw, Poland.
- [12] Krawczak, M., Aladjov, H. (2002) Generalized Net Model of Backpropagation Algorithm. *Proc. of 3rd International Workshop on Generalized Nets*, Sofia, Bulgaria, October 2002, 32-36.
- [13] Oppenheim, A. V., Schafer, R. W. (1989) *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice Hall.
- [14] Pitts, W., McCulloch, W. S. (1947) How We Know Universals: the Perception of Auditory and Visual Forms. *Bulletin of Mathematical Biophysics*, 9, 127-147.

- [15] Radeva, V., Krawczak, M., Choy, E. (2002) Review and Bibliography on Generalized Nets Theory and Applications. *Advanced Studies in Contemporary Mathematics*, 4, 2, 173-199.
- [16] Wan, E., Beaufays, F. (1996) Diagrammatic Derivation of Gradient Algorithms for Neural Networks. *Neural Computation*, 8, 1, 182-206.

