Generalized Net Model of the Cognitive and Neural Algorithm for Adaptive Resonance Theory 1

Todor Petkov, Sotir Sotirov

Prof. Dr. Assen Zlatarov University
Burgas, Bulgaria
E-mails: t-petkov@mail.bg, ssotirov@btu.bg

Received: July 03, 2013 **Accepted: December 06, 2013**

Published: December 20, 2013

Abstract: *The artificial neural networks are inspired by biological properties of human and animal brains. One of the neural networks type is called ART [4]. The abbreviation of ART stands for Adaptive Resonance Theory that has been invented by Stephen Grossberg in 1976 [5]. ART represents a family of Neural Networks. It is a cognitive and neural theory that describes how the brain autonomously learns to categorize, recognize and predict objects and events in the changing world. In this paper we introduce a GN model that represent ART1 Neural Network learning algorithm [1]. The purpose of this model is to explain when the input vector will be clustered or rejected among all nodes by the network. It can also be used for explanation and optimization of ART1 learning algorithm.*

Keywords: *Neural Networks, Generalized Nets, ART1.*

Introduction

Every learning system must be able to adapt in changing environment (it must be plastic). Constant changes can lead to the instability of the system, due to the fact, that the system can obtain information by learning, only when the old information is forgotten [3, 4].

The type of neural network that solves this problem is called ART [5, 6]. The essence of ART's predictive power is its ability to autonomously carry out fast, incremental, unsupervised and supervised learning in response to the changing world, keeping previously learned memories at the same time [4]. More generally, we can quickly learn about new environment, even if it has not been told about the difference of the rules in each environment. In a remarkable degree, humans can rapidly learn new facts without being forced to just as rapidly forget what they already know. As a result, we can confidently go out into the world without fearing that, in learning to recognize a new friend's face, we will suddenly forget the faces of our family and friends [4]. The ART method consist two layers. The first one is known as Comparison layer and the second is called Recognition layer, fully connect with “bottom-up” and “top-down” weights and reset module which control the degree of similarity between both layers. There are two methods of learning – supervised and unsupervised.

Unsupervised learning method is used for neural networks ART1, ART2, ART3, … [7, 8]. Supervised method is known as ARTMAP [7, 8].

The ART1 learning algorithm is described below, it operates with binary input vectors [3]. According to [2, 9, 10] learning algorithm is:

1. Notations

The set of input vectors

$$I(x)=\{x(1),x(2),\ldots,x(N)\},\ x(t)\in R^n,\ t=1,\ 2,\ \ldots,$$

where t (time) is a natural number.

Basically ART1 consists of two layers:

F1 is the input (Comparison field) with n-neurons $F1_1$, $F1_2$, …, $F1_n$;
F2 is the output (Recognition field) with M-neurons $F1_1$, $F2_2$, …, $F2_M$.

Weights from neuron i of $F1$ layer towards the neuron j of $F2$ layer

$$"w_{i,j}(t),\ i=1,\ 2,\ \ldots,\ N;\ j=1,\ 2,\ \ldots,\ M;\ W(t)=\left(w_{i,j}(t)\right)"$$

Each column in matrix $W(t)$ is column vector $w_j(t)\in R^n,\ j=1,\ 2,\ \ldots,\ M$.

$$"w_j(t)=\left(w_{1j}(t),w_{2j}(t),\ \ldots,\ w_{ij}(t)\right)^{\mathrm{T}}"$$

Weights from neuron j of $F2$ towards the neuron i of $F1$

$$"v_{j,i}(t),\ j=1,\ 2,\ \ldots,\ M;\ i=1,\ 2,\ \ldots,\ n;\ V(t)=\left(v_{j,i}(t)\right)"$$

Each row in matrix $V(t)$ is column vector $"v_j(t)\in R^n,\ j=1,\ 2,\ \ldots,\ M"$

$$"v_j(t)=\left(v_{j1}(t),v_{j2}(t),\ \ldots,\ v_{ji}(t)\right)^{\mathrm{T}};"$$

$o(t)\in R^M$ – firing vector that consist one component for each neuron in $F2_j$.

Vector $\theta\in R^M$ is null vector of the space.

2. Initialization of parameters

- n – number of neurons in $F1$ layer;
- M – number of neurons in $F2$ layer;
- N – number of input vectors;
- ρ – vigilance threshold.

$$w_{i,j}^0=w_{i,j}(1)=\frac{1}{1+N},v_{j,i}^0=v_{j,i}(1)=1\ \text{for}\ i=1,\ 2,\ \ldots,\ n;\ j=1,\ 2,\ \ldots,\ M \tag{1}$$

where $w_{i,j}^0,v_{j,i}^0$ are initialized weights at $t = 0$.

2.1. At the beginning of the work, the fire vector $o(1)$ has all of its components equal to 1, i.e.,

$$o_j=1\ \text{in}\ F2. \tag{2}$$

2.2. Read an input vector *x*(*t*).

2.3. Compute the activation values for all the enabled neurons *j* of *F*2:

$$"net(j) = \langle w_j(t), x(t) \rangle" \tag{3}$$

Select the winner neuron that has

$$"k, 1 \le k \le M \max \{ net_j(t) | o_j = 1 \} = net_k(t)". \tag{4}$$

If the maximum is not unique and there is a tie for the winning neuron *k* in *F*2 layer, then choose a special rule to break the tie.

2.4. Process the vigilance test for $F2_k$ output neuron:

$$"r = \frac{\langle v_k(t), x(t) \rangle}{\| x(t) \|}" \tag{5}$$

If $r \ge \rho$ there is match go to step 2.5 otherwise go to step 2.6.

2.5. Update the vectors $v_k(t)$ and $w_k(t)$ as follows:

$$"v_k(t+1) = v_k(t) * x(t)" \tag{6}$$

$$"w_k(t+1) = \frac{v_k(t+1)}{0.5 + \| v_k(t+1) \|}" \tag{7}$$

$$"v_j(t+1) = v_j(t), w_j(t+1) = w_j(t) \text{ for } j = 1, 2, \ldots, M, j \ne k" \tag{8}$$

Put output $o_j(t+1) = 1, j = 1, M$ and store input pattern, go to step 2.7.

2.6. Disable the output neuron $F2_k$ i.e. $"o_k(t) = 0"$ and go to 2.3. When $o(t) = \theta$ the network rejects the input vector *x*(*t*) and it is stored in reject set *RI*(*x*) go to step 2.7.

2.7. Put the old weights on the positions of new weights and enable all the output neurons, namely

$$"v_j(t+1) = v_j(t), w_j(t+1) = w_j(t), o_j(t+1) = 1 \text{ for } j = 1, 2, \ldots, M" \tag{9}$$

2.8. Step 2.2 has been repeated again with the input set *I*(*x*).

2.9. Print the content of classes and the matrixes $WandV$.

We introduced a static structure of ART1 Neural Network here is described ART1 learning algorithm.

GN-model

The GN-model is presented in Fig. 1. Initially the following tokens enter the generalized net:

- in place L_1 – a token with an initial characteristic "Input vectors" – $x_0^\alpha = "I(x)"$;
- in place L_6 – a token with an initial characteristic "Number of neurons in F1" – "$x_0^\beta = n$";
- in place L_{17} – a token with an initial characteristic "Vigilance threshold" – $x_0^\gamma = "\rho"$;
- in place L_{22} – a token with an initial characteristic "Initialized weights "$v_{j,i}^0$" in $F2$, Number of neurons in $F2$, Special rule (S)" – $x_0^\delta = "v_{j,i}^0, M, S"$.

The GN is introduced by the set of transitions:

$$\mathrm{A} = \{Z_1, Z_2, Z_3, Z_4, Z_5\}$$

where the transitions describe the following processes:
Z_1 = "Distribution of the input vectors";
Z_2 = "Initializing of weights $w_{i,j}^0$ in $F1$";
Z_3 = "Determination of weights $w_{i,j}$ in $F1$";
Z_4 = "Determination of resonance state";
Z_5 = "Determination of weights $v_{j,i}$ in $F2$".

GN-model consists of five transitions with the following description:

$$Z_1 = \langle \{L_1, L_5, L_9, L_{25}\}, \{L_2, L_3, L_4, L_5\}, R_1, \vee(\wedge(L_5, L_9), \wedge(L_5, L_{25}), L_1, L_5)\rangle$$

$$R_1 = \begin{array}{c|cccc} & L_2 & L_3 & L_4 & L_5 \\ \hline L_1 & False & False & False & True \\ L_5 & W_{5,2} & W_{5,3} & W_{5,4} & True \\ L_9 & False & False & False & True \\ L_{25} & False & False & False & True \end{array},$$

where:
$W_{5,2} = W_{5,3} = W_{5,4}$ = "There are input vectors".

The token α from place L_1 that enters in place L_5 does not change its characteristic. The token α' from place L_5 splits in to three identical tokens entering in places L_2, L_3, L_4. The tokens ζ, δ' from places L_9 and L_{25} enter in place L_5 obtaining a characteristic:

$$x_{cu}^{\alpha'} = "\alpha' + 1".$$

$$Z_2 = \langle \{L_6, L_8\}, \{L_7, L_8\}, R_2, \vee(L_6, L_8)\rangle$$

$$R_2 = \begin{array}{c|cc} & L_7 & L_8 \\ \hline L_6 & False & True \\ L_8 & W_{8,7} & True \end{array},$$

where:
$W_{8,7}$ = "The weights are initialized".

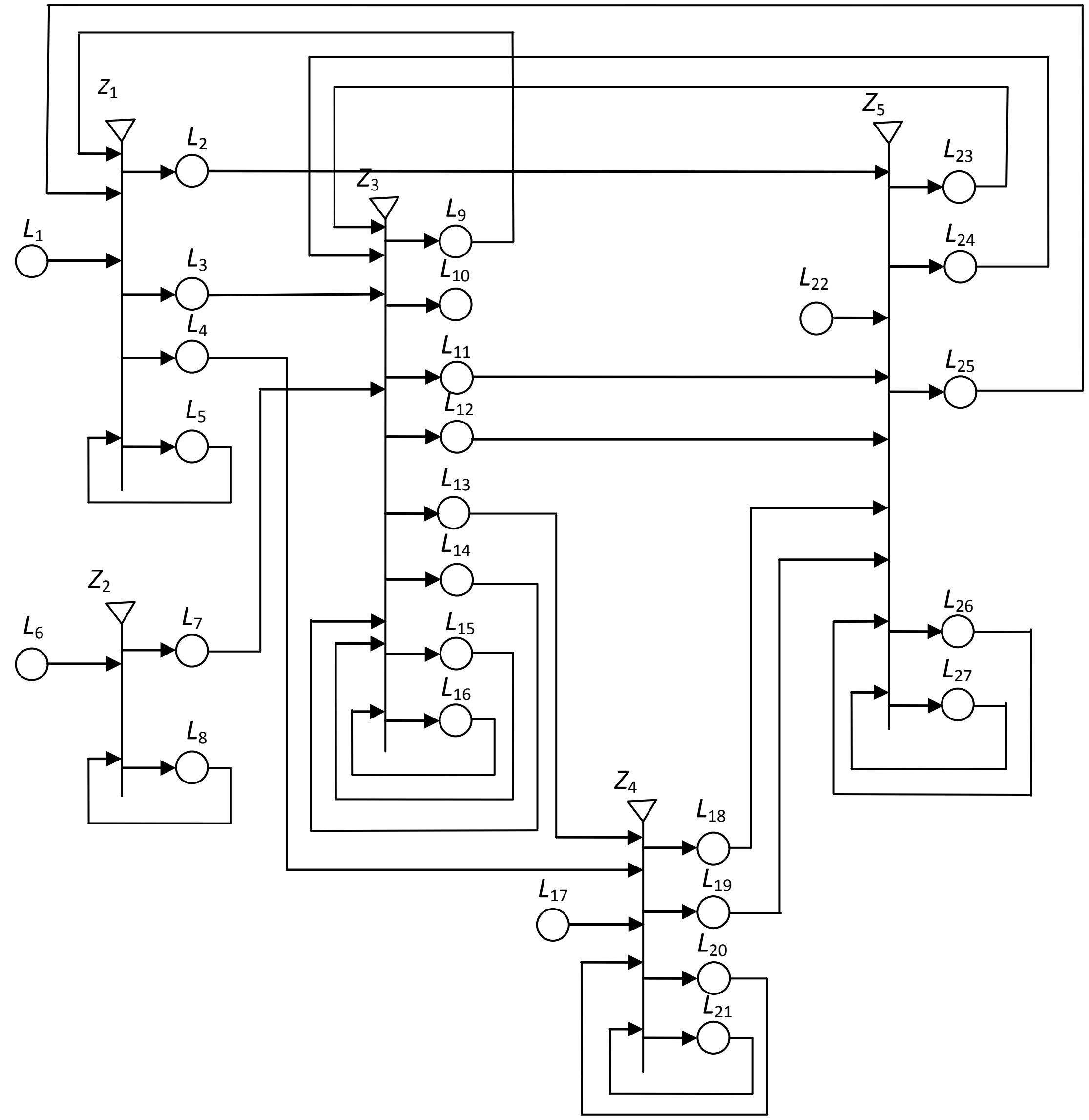

Fig. 1 GN-model

The token β from place L_6 enters in place L_8 obtaining a characteristic:

$$x_{cu}^{\beta'} = \frac{1}{1+n}.$$

The token β' in place L_8 enters in place L_7 keeping its characteristic.

$$Z_3 = \langle\{L_3, L_7, L_{14}, L_{15}, L_{16}, L_{23}, L_{24}\}, \{L_9, L_{10}, L_{11}, L_{12}, L_{13}, L_{14}, L_{15}, L_{16}\}, R_3,$$
$$\vee(\wedge(L_3(\vee(L_7, L_{23})), L_{14}, L_{15}, L_{16}, L_{24})\rangle$$

$$R_3 = \begin{array}{c|cccccccc} & L_9 & L_{10} & L_{11} & L_{12} & L_{13} & L_{14} & L_{15} & L_{16} \\ \hline L_3 & False & False & False & False & False & False & False & True \\ L_7 & False & False & False & False & False & False & False & True \\ L_{14} & False & False & False & False & False & True & False & W_{14,16} \\ L_{15} & False & W_{15,10} & False & W_{15,12} & W_{15,13} & W_{15,14} & True & False' \\ L_{16} & W_{16,9} & W_{16,10} & W_{16,11} & False & False & False & False & True \\ L_{23} & False & False & False & False & False & False & True & False \\ L_{24} & False & False & False & False & False & False & W_{24,15} & True \end{array}$$

where:
$W_{14,16}$ = "Weights $W_k(t + 1)$ are updated";
$W_{15,10} = W_{16,9} = W_{16,10} = W_{24,15}$ = "$o(k)= \theta$";
$W_{15,12} = W_{15,14}$ = "Weights $V_k(t + 1)$ are updated";
$W_{15,13}$ = "The value of winning neuron is defined";
$W_{16,11}$ = "The values to the neurons are defined".

The token α' from place L_3 splits in two identical tokens that enter in places L_{15} and L_{16}. The tokens α', β' from places L_3 and L_7 enter in place L_{16} obtaining a characteristic:

$$x_{cu}^{\zeta} = "W_j * x(t)".$$

The token ζ splits in two identical tokens one of which enters in place L_{11} and in place L_{16} stays "α', β', ζ". The tokens δ''' and δ'' from place L_{23} enter in place L_{15} and unite with token α' from place L_3 obtaining a characteristic:

$$x_{cu}^{\eta} = "V_k(t) * x(t)".$$

In place L_{15} stays "α', η". The token μ' from place L_{23} unites with token η from place L_{15} and obtains a characteristic:

$$x_{cu}^{\eta'} = "update(V_k(t) * x(t))".$$

The token ν from place L_{14} unites with token ζ from place L_{16} and obtains a characteristic:

$$x_{cu}^{\zeta} = "w_j(t)".$$

The token α' from places L_{15} and L_{16} enters in place L_{10} keeping its characteristic. The token η' from places L_{15} splits in two identical tokens that enter in places L_{12} and L_{14}, keeping its characteristic in place L_{12}. The token η' from place L_{15} enters in place L_{14} obtaining a characteristic:

$$x_{cu}^{\nu} = \frac{v_k(t+1)}{0.5 + \left\| v_k(t+1) \right\|}.$$

The token η from place L_{15} enters in place L_{13} keeping its characteristic. The token ζ from place L_{16} enters in place L_9 also keeping its characteristic. The token σ' from L_{24} enters in place L_{16}, obtaining a characteristic x_{cu}^{ζ}. The token σ'' from L_{24} splits in two identical tokens that enter in places L_{15} and L_{16}, obtaining characteristics:

$$x_{cu}^{\alpha'} = RI\left(x\right).$$

$$Z_4 = \langle\{L_4, L_{13}, L_{17}, L_{20}, L_{21}\}, \{L_{18}, L_{19}, L_{20}, L_{21}\}, R_4, \vee(\wedge(L_4, L_{13}), \wedge(L_{17}, L_{21}), L_{20})\rangle$$

$$R_4 = \begin{array}{c|cccc} & L_{18} & L_{19} & L_{20} & L_{21} \\ \hline L_4 & False & False & False & True \\ L_{13} & False & False & False & True \\ L_{17} & False & False & True & False’ \\ L_{20} & W_{20,18} & W_{20,19} & True & False \\ L_{21} & False & False & W_{21,20} & True \end{array}$$

where:
$W_{20,18}$ = "$\rho \le r$";
$W_{20,19} = \neg W_{20,18}$;
$W_{21,20}$ = "Resonant state is calculated".

The tokens α' and η from places L_4 and L_{13} unite in place L_{21}, obtaining a characteristic:

$$x_{cu}^{\lambda} = "\frac{V_k}{\left\|x\left(t\right)\right\|}".$$

The tokens λ and γ from places L_{17} and L_{21} unites in place L_{20}, obtaining a characteristic:

$$x_{cu}^{\mu} = "\left(\rho = r\right)".$$

In place L_{21} stays λ, in all time of life of GN. The token μ from place L_{20} splits in two tokens μ', μ'', that enter in places L_{18} and L_{19}. The token μ' that enters in place L_{18} obtains a characteristic:

$$x_{cu}^{\mu'} = "\rho \le r".$$

The token μ'' that enters in place L_{19} obtains a characteristic:

$$x_{cu}^{\mu''} = "\rho > r".$$

$$Z_5 = \langle\{L_2, L_{11}, L_{12}, L_{18}, L_{19}, L_{22}, L_{26}, L_{27}\}, \{L_{23}, L_{24}, L_{25}, L_{26}, L_{27}\}, R_5,$$
$$(\vee(\wedge(L_2, L_{11}, L_{22}), \wedge(L_{22}, L_{27}), \wedge(L_{12}, L_{26}), \wedge(L_{19}, L_{27})), L_{18})\rangle$$

$$R_5 = \begin{array}{c|ccccc} & L_{23} & L_{24} & L_{25} & L_{26} & L_{27} \\ \hline L_2 & False & False & False & False & True \\ L_{11} & False & False & False & False & True \\ L_{12} & False & False & False & True & False \\ L_{18} & True & False & False & False & True \\ L_{19} & False & False & False & False & True \\ L_{22} & False & False & False & True & True \\ L_{26} & W_{26,23} & False & W_{26,25} & True & False \\ L_{27} & False & W_{27,24} & False & W_{27,26} & True \end{array} ,$$

where:
$W_{26,23}$ = "Weights and winning neuron are defined";
$W_{26,25}$ = "Weight matrices are updated;
$W_{27,24}$ = "Winning neuron is disabled";
$W_{27,26}$ = "Winning neuron is defined".

The token δ from place L_{22} splits in two tokens – δ' enters in place L_{27} and δ'' enters in place L_{26} obtaining characteristics:

$$\delta' = M, S; \;\; \delta'' = v_{j,i}^{0} \; .$$

In place L_{27} stays δ', in all time of life of GN. The token α' from place L_2 enters in place L_{27}, obtaining a characteristic:

$$x_{cu}^{\delta'} = o(t) = 1 .$$

The token ζ from place L_{11} unites with token δ' in place L_{27} obtaining a characteristic:

$$\left\{ \frac{x_{cu}^{\varepsilon} = "\max(k),\; o(k) = 1"}{x_{cu}^{\omega} = "\max(k),\; o(k) = 1 \text{ if } = S"} \right\} .$$

The tokens ε and ω from place L_{27} enter in place L_{26} obtaining a characteristic:

$$x_{cu}^{\delta''} = \max(k), V(k)" .$$

The tokens δ'' and δ''' from place L_{26} enter in place L_{23} keeping their characteristics. The token η' from place L_{12} unites with token δ'' in place L_{26} obtaining a characteristic:

$$x_{cu}^{\psi} = v_j(t) .$$

The token μ' from place L_{18} splits in two identical tokens that enter in places L_{23} and L_{27}, in place L_{23} obtaining a characteristic:

$$x_{cu}^{\mu'} = "o_k(t) = update" .$$

The token μ' from place L_{18} enters in place L_{27} obtaining a characteristic:

$$\left\{\frac{x_{cu}^{\varepsilon} = \text{"}\max(k),\ o(k) = 0\text{"}}{x_{cu}^{\omega} = \text{"}\max(k),\ o(k) = 0 \ \text{ if } = S\text{"}}\right\}.$$

The token μ'' in place L_{19} unites with token ε in place L_{27} obtaining a characteristic:

$$x_{cu}^{\sigma} = \text{"}o_k(t) = 0,\ o_M(t) = 1\text{"}.$$

The token μ'' in place L_{19} enters in place L_{27} obtaining a characteristic:

$$x_{cu}^{\omega'} = \text{"}o_k(t) = 0,\ o_M(t) = 1 \ \text{ if } = S\text{"}.$$

The token ψ from place L_{26} enters in place L_{25} keeping its characteristic. The token σ in place L_{27} enters in place L_{24} obtaining a characteristic:

$$\left\{\frac{x_{cu}^{\sigma'} = \text{"}net_{(j)} \ \text{ if } o_k(t) = 0\text{"}}{x_{cu}^{\sigma''} = \text{"}RI(x) \ \text{ if } o_M(t) = \theta\text{"}}\right\}.$$

The token ω′ in place L_{27} enters in place L_{26} obtaining a characteristic:

$$x_{cu}^{\delta'''} = \max(k), V(k) \ \text{ if } = S \ .$$

Conclusion

In the present work a GN-model to describe the learning algorithm of ART1 Neural Network is used. It has been shown, how the input vector is recognized and the process of acceptance or rejection (if there are not uncommitted neurons) from neural network. In that matter, the method can be used to solve the “stability-plasticity dilemma”.

This paper is in the second series of papers devoted on the ART1 neural network.

References

1. Atanassov K. (1991). Generalized nets, World Scientific, Singapore, New Jersey, London.
2. Carpenter G. A. (1988). The ART of Adaptive Pattern Recognition by Self Organizing Neural Network, Computer, 21, 77-88.
3. Carpenter G. A., S. Grossberg (1987). A Massively Parallel Architecture for a Self-organizing Neural Pattern Recognition Machine, Computer Vision, Graphics, and Image Processing, 37, 54-115.
4. Grossberg S. (2013). Adaptive Resonance Theory: How a Brain Learns to Consciously Attend, Learn, and Recognize a Changing World, Neural Networks, 37, 1-47.
5. Grossberg S. (1988). Neural Network and Natural Intelligence, Cambridge, MA: MIT Press.
6. Grossberg S. (1976). Adaptive Pattern Classification and Universal Recoding. II. Feedback, Expectation, Olfaction, and Illusions, Biological Cybernetics, 23, 187-202.
7. Grossberg S. (1976). Adaptive Pattern Classification and Universal Recoding. I. Parallel Development and Coding of Neural Feature Detectors, Biological Cybernetics, 23, 121-134.

8. Grossberg S. (1986). The Adaptive Self-organization of Serial Order in Behavior: Speech, Language, and Motor Control, in Pattern Recognition by Humans and Machines (Eds. Schwab E. C., H. C. Nusbaum), Vol. 1, Academic Press, New York.
9. Grossberg S. (1978). Do All Neural Networks Really Look Alike? A Comment on Anderson, Silverstein, Ritz, and Jones, Psycho Rev, 85, 592-596.
10. Popoviciu N. (2004). An Efficient ART1 Algorithm Based on Vector Computation, General Mathematics, 12(3), 37-52.
11. Krose B., P. Van der Smagt (1996). An Introduction to Neural Networks, Chapter 6, 8th Edition, University of Amsterdam.

Todor Petkov, M.Sc.
E-mail: t-petkov@mail.bg

M.Sc. Todor Petkov graduated from “Prof. Dr. Assen Zlatarov” University, Burgas under the department of “Computers, systems and technologies” in 2012. He is pursuing research on “Generalized net model of neural networks”. He has interest over neural networks, generalized nets and computers.

Assoc. Prof. Sotir Sotirov, Ph.D.
E-mail: ssotirov@btu.bg

Sotir Sotirov was graduated from Technical University in Sofia, Bulgaria, Department of Electronics. He received his Ph.D. degree from “Prof. Dr. Assen Zlatarov” University, Burgas in 2005. His current research interests are in the field of modeling of neural networks.