

# Generalized net model of forest fire detection with ART2 neural network

**Todor Petkov, Stanimir Surchev and Sotir Sotirov**

“Prof. Assen Zlatarov” University – Bourgas  
1 “Prof. Yakimov” Blvd., Burgas 8010, Bulgaria  
e-mails: [todor\\_petkov@btu.bg](mailto:todor_petkov@btu.bg), [stanimir\\_surchev@btu.bg](mailto:stanimir_surchev@btu.bg),  
[ssotirov@btu.bg](mailto:ssotirov@btu.bg)

**Abstract:** This paper thoroughly describes the use of unsupervised adaptive resonance theory ART2 neural network for the purposes of forest fire detection. In order to train the network, the pixel value of red color is regarded as learning vector. At the end the trained network was tested by the values of a picture and determines the design, or how to visualize the converted picture. As a result we had the same picture with colors according to the network. Here we use the generalized net to prepare a model that describes the process of the color recognition.

**Keywords:** Generalized Nets, Neural Networks, Adaptive Resonance Theory

**AMS Classification:** 68Q85.

## 1 Introduction

Adaptive resonance theory (ART) [3, 4] was introduced by Stephen Grossber in 1976. In this work, the ART2 [2, 3, 4] neural network [8], slow learning algorithm [5, 6, 7] for image processing of a picture is used. ART2 is designed to perform operation over continuous valued input vectors. It consists of two layers – the first one has complex units or neurons that support a combination of normalization and noise suppression. The second layer is a competitive one. Both of them are fully connected with bottom-up and top-down weights. The bottom-up and top-down signals are needed for the reset mechanism. The neural network is learned by modification of the bottom-up and top-down weights.

Here are used symbols that have cross shape to detect the fire, crosses could be useful if the system where the fire burns is only able to makes pictures only in two colors. The crosses of the picture represent the fire and the rest of the picture is not important.

The slow learning algorithm according to [5, 7] can be expressed by the following steps:

## BEGINNING OF ALGORITHM

**Step 1.** Initialize parameters:

- $a, b, \Theta, c, d, e, \alpha, \delta$
- where:
- $a, b$  – fixed weights in the F1 layer.
- $\Theta$  - noise suppression parameter.
- $c$  – fixed weight used in testing for reset.
- $d$  – activation of the winning F2 unit.
- $e$  – small parameter using preventing division by zero.
- $\alpha$  – learning rate.
- $\rho$  – vigilance threshold.
- $b_j$  – initial bottom-up weights.
- $t_j$  – initial top-down weights.

**Step 2.** For each input vector “s” do Steps 3-11.

**Step 3.** Update F1 unit activation:

$$u_i = 0, \quad x_i = \frac{w_i}{e + \|w\|}, \quad q_i = 0$$

$$w_i = s_i, \quad p_i = 0, \quad v_i = f(x)_i$$

The activation function is  $f(x) = \begin{cases} x & \text{if } x \geq \Theta \\ 0 & \text{if } x < \Theta \end{cases}$

Update  $F_1$  activations again

$$u_i = \frac{v_i}{e + \|v\|}, \quad x_i = \frac{w_i}{e + \|w\|}, \quad q_i = \frac{p_i}{e + \|p\|}$$

$$w_i = s_i + a * u_i, \quad p_i = u_i, \quad v_i = f(x)_i + bf(q)_i$$

**Step 4.** Compute the signals to F2 units:

$$y_j = \sum_j b_j * p_i$$

**Step 5.** While reset is true, do Steps 6-7.

**Step 6.** Find F2 unit with largest signal. (Define  $J$  such that  $y_J \geq y_j$  for  $j = 1, 2, \dots, m$ .)

**Step 7.** Check for reset:

$$u_i = \frac{v_i}{e + \|v\|}, \quad r_i = \frac{u_i + cp_i}{e + \|u\| + c\|p\|}, \quad p_i = u_i + dt_{Ji}$$

If  $\|r\| < \rho - e$  then  $y_J = -1$  (inhibit  $J$ )

Reset is true; repeat Step 5.

If  $\|r\| \geq \rho - e$ , reset is false; proceed to Step 8

**Step 8.** Update weights for winning unit  $J$

$$t_{Ji} = \alpha du_i + \{1 + \alpha d(d-1)\}t_{Ji}$$

$$b_{iJ} = \alpha du_i + \{1 + \alpha d(d-1)\}b_{iJ}$$

## END OF ALGORITHM

The network consists of one cluster. It transforms the graphical information into one color that is used for learning vector the color is red [255 0 0]. Red color that the cluster is learned represents the pixel value of fire.

In a trained network, the picture is splitting up in matrix that consists of testing vectors with three elements. If the vector of the picture gets into the cluster, the pixel will obtain black color. If it does not get anywhere it will be colored in white. At the end of the process final picture will consists of two colors, black and white, regions that are colored in black will be marked in red symbols. As a result red crosses represent the region where the fire is.

To test the network it was used a picture that was taken from [9], Fig. 1. The result after applying it to the network is (Fig 2.) and then red symbols were applied in pseudo-chaotic way to the resulted picture where the black color is (Fig. 3).



Fig. 1. Original picture



Fig. 2. Picture after clusterization



Fig. 3. Result picture

## 2 GN-Model

Initially the following tokens enter the Generalized Net (GN) [1]:

- in place  $L_1$  – one token with initial characteristic “Input matrix of picture”;
- in place  $L_2$  – one token with initial characteristic “Structure of neural network and number of clusters”;
- in place  $L_3$  – one token with initial characteristic “Learning vectors”;
- in place  $L_{17}$  – one token with initial characteristic “Value of pseudo-chaotic distribution of symbols in the picture”.

On the first step in place  $L_{11}$  there is a token that starts the procedure. The GN, as illustrated on Fig. 4, consists of the following set of six transitions:

- $Z_1$  = “Converting picture to the input vector”
- $Z_2$  = “Neural Network training”
- $Z_3$  = “Determine cluster of input vector”
- $Z_4$  = “Determine values pixels of the picture”
- $Z_5$  = “Converting output vector to picture of two colors”
- $Z_6$  = “Applying symbols to the picture”

The six transitions are formally described, as follows.

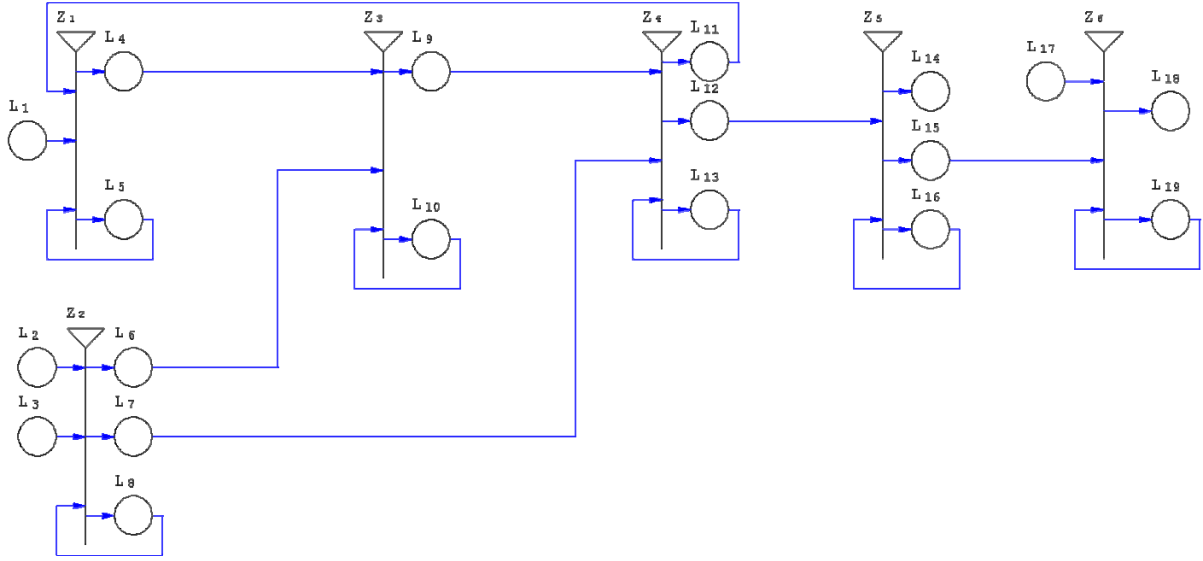


Figure 4. GN model of the process of forest fire detection using ART2 neural network

$$Z_1 = \langle \{L_1, L_5, L_{11}\}, \{L_4, L_5\}, R_1, \vee(L_1, \wedge(L_5, L_{11})) \rangle$$

where

$$R_1 = \begin{array}{c|cc} & L_4 & L_5 \\ \hline L_1 & False & True \\ L_5 & W_{5,4} & True \\ L_{11} & False & True \end{array},$$

where the predicate  $W_{5,4}$  = “There is an active signal for the next input vector”

The token that enter from place  $L_5$  in position  $L_4$  does not obtain new characteristic. The token that transfers from place  $L_{11}$  to place  $L_5$  obtains the characteristic “request for next input vector from the matrix”.

$$Z_2 = \langle \{L_2, L_3, L_8\}, \{L_6, L_7, L_8\}, R_2, \vee(L_8, \wedge(L_2, L_3)) \rangle$$

where

$$R_2 = \begin{array}{c|ccc} & L_6 & L_7 & L_8 \\ \hline L_2 & False & True & False \\ L_3 & False & True & False \\ L_8 & W_{8,6} & True & W_{8,7} \end{array},$$

where the predicates  $W_{8,6} = W_{8,7}$  = “The neural network is trained”.

The token that enters place  $L_8$  obtain characteristic “Neural Network”.

The token that moves from place  $L_8$  to place  $L_6$  does not obtain any new characteristic.

The token that transfers from place  $L_8$  to place  $L_7$  obtains the characteristic “Conditions for winner clusters”.

$$Z_3 = \langle \{L_4, L_6, L_{10}\}, \{L_9, L_{10}\}, R_3, \vee(L_6, \wedge(L_4, L_9)) \rangle$$

where

$$R_3 = \begin{array}{c|cc} & L_9 & L_{10} \\ \hline L_4 & False & True \\ L_6 & False & True \\ L_{10} & W_{10,9} & True \end{array},$$

where the predicate  $W_{10,9}$  = “There is a result from ART2 NN”;

The tokens that move from places  $L_4$  and  $L_{10}$  to place  $L_9$  obtain the characteristic “Winner cluster was determined”.

The token that transfers from place  $L_6$  to place  $L_{10}$  does not obtain any new characteristic.

$$Z_4 = \langle \{L_7, L_9, L_{13}\}, \{L_{11}, L_{12}, L_{13}\}, R_4, \vee(L_7, \wedge(L_9, L_{13})) \rangle$$

where

$$R_4 = \begin{array}{c|ccc} & L_{11} & L_{12} & L_{13} \\ \hline L_7 & False & False & True \\ L_9 & False & False & True \\ L_{13} & W_{13,11} & W_{13,12} & True \end{array},$$

where  $W_{13,11}$  = “The value pixel of picture was determined”;

$W_{13,12}$  = “The values of picture were determined”;

The tokens that enter from place  $L_9$  in place  $L_{13}$  do not obtain new characteristic.

The tokens that enter from places  $L_7$  and  $L_{13}$  in place  $L_{11}$  obtain characteristic “Request for next input vector”.

The token that enter from place  $L_{13}$  in place  $L_{12}$  obtain characteristic “Converted values of the picture”.

$$Z_5 = \langle \{L_{12}, L_{16}\}, \{L_{14}, L_{15}, L_{16}\}, R_5, \vee(L_{12}, L_{16}) \rangle$$

where

$$R_5 = \begin{array}{c|ccc} & L_{14} & L_{15} & L_{16} \\ \hline L_{12} & False & False & True \\ L_{16} & W_{16,14} & W_{16,15} & True \end{array},$$

where  $W_{16,14} = W_{16,15}$  = “The values of the picture were converted”.

The token that moves form place  $L_{12}$  to place  $L_{15}$  obtains the characteristic “Converted values of picture”. The token that transfers from place  $L_{16}$  to place  $L_{14}$  obtain characteristic “Visualizing Black/White picture”. The token that moves from place  $L_{16}$  to place  $L_{15}$  obtain characteristic “Black/White picture”.

$$Z_6 = \langle \{L_{15}, L_{17}, L_{19}\}, \{L_{18}, L_{19}\}, R_5, \vee(L_{16}, \wedge(L_{15}, L_{17})) \rangle$$

$$R_6 = \begin{array}{c|cc} & L_{18} & L_{19} \\ \hline L_{15} & False & True \\ L_{17} & False & True \\ L_{19} & W_{19,18} & True \end{array},$$

where  $W_{19,18}$  = “The symbols of the picture were applied”

The token that moves from place  $L_{15}$  to place  $L_{19}$  does not obtain new characteristic. The token that transfers from place  $L_{19}$  to place  $L_{18}$  obtains the characteristic “Visualizing picture with symbols”.

### 3 Conclusion

The neural network was learned by slow learning of ART2. A cluster was presented by red color, in case that a cluster hasn't won; the pixel is depicted in white. The neural network was tested by random picture in bitmap format. As a result of the training process, three figures are presented. The program code is implemented in Matlab environment.

### Acknowledgment

The authors are grateful for the support provided by the project DFNI-I-01/0006 “Simulating the behaviour of forest and field fires”, funded by the National Science Fund, Bulgarian Ministry of Education, Youth and Science.

### References

- [1] Atanassov, K., *Generalized Nets*, World Scientific, Singapore, 1991.
- [2] Carpenter, G. A., S. Grossberg. The ART of adaptive pattern recognition by a self-organizing neural network, *Computer*, Vol. 21, March 1988, No. 3, 77–88.
- [3] Grossberg, G., A. Carpenter *ART2: Self-organization of stable category recognition codes of analog input patterns*, Boston University, Center for Adaptive systems, 1987 Dec 1; 26(23), 4919–4930.
- [4] Grossberg, G. Adaptive pattern classification and universal recoding. II. Feedback, expectation, olfaction, and illusions, *Bioi. Cybernet.* Vol. 23, 1976, 187–202.
- [5] Sivanandam, S. N., S. N. Deepa, *Introduction to Neural Networks using Matlab 6.0*.
- [6] Krose B., P. van der Smagt, *An introduction to Neural Networks, Chapter 6*. 8th Ed., University of Amsterdam, 1996.
- [7] Fausett, L. *Fundamentals of Neural Networks; Architecture, algorithms and applications*, Prentice Hall, 1993.
- [8] McCulloch, W., W. Pitts, A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, Vol. 5, 1943, No. 4, 115–133.
- [9] [http://en.wikipedia.org/wiki/File:Mount\\_Carmel\\_forest\\_fire14.jpg](http://en.wikipedia.org/wiki/File:Mount_Carmel_forest_fire14.jpg)