BAM (Bi-directional Associative Memory) Neural Network Simulator

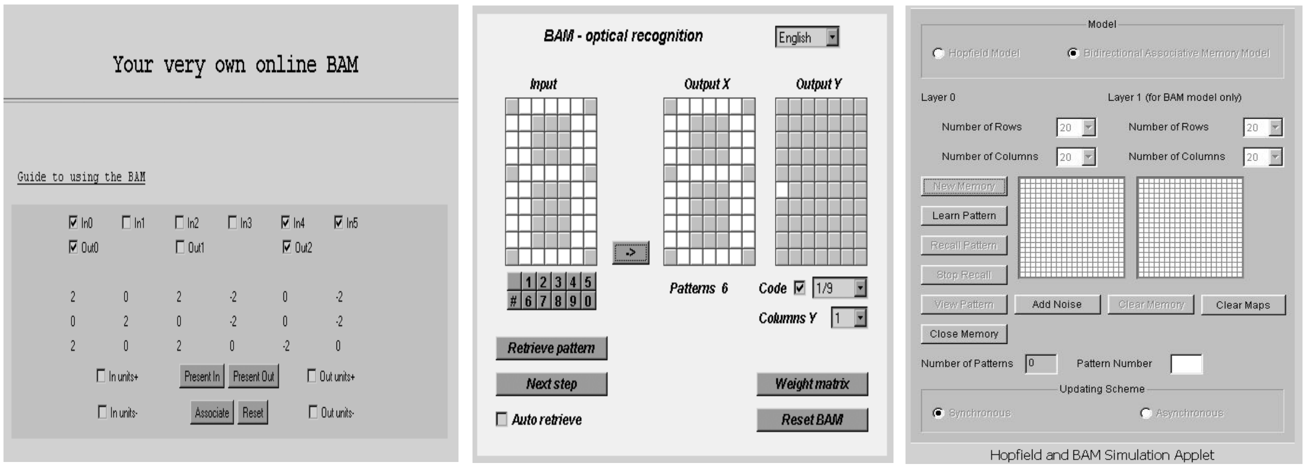
J. Zlateva, G. Todorov

Abstract: *On Windows platform implemented BAM (Bi-directional Associative Memory) neural network simulator is presented. The realization in two parts – main and user (interface) unit allows using it in the student education and as well as a part of other software applications, using this kind of neural network.*

Keywords*: Neural Network, Bi-directional Associative Memory (BAM), Simulator*

INTRODUCTION

There are some BAM neural network simulators available on the Internet (Fig. 1), but they don't offer possibilities for: choice of the signal propagation direction (Fig.1a), viewing the signal propagation history (Fig.1a, c), changes in the structure of the neural network (Fig.1b), free choice of the vector pairs in the training set (Fig.1b), viewing the weight matrix as a result from the training process (Fig.1c).



a) b) c)

Fig.1 Some BAM neural network simulators available on the Internet
a) Source - http://www.mdx.ac.uk/www/ai/garden/gleadall/BAM.html
b) Source - http://service.felk.cvut.cz/courses/36NAN/36NAN101/bam.html
c) Source - http://www.comp.nus.edu.sg/~pris/AssociativeMemory/DemoApplet

A BAM neural network simulator on Windows platform is implemented. The calculations of the weight matrix and the signal propagation are made using equations in [1]. The simulator is in two parts build:

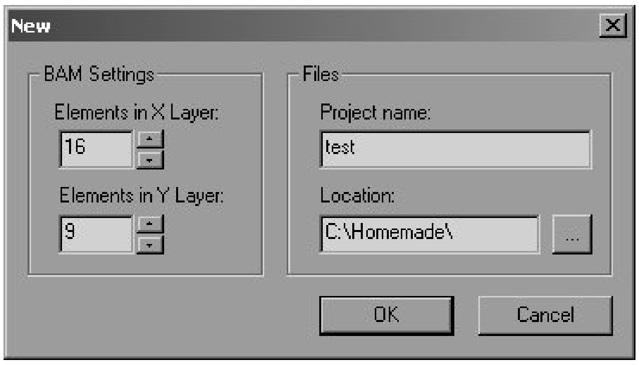
- Main unit – the BAM neural network simulator, that can be used in other software applications;
- Graphical user interface unit – full control and observation of the work of BAM neural network.

The implemented simulator offers possibilities for:

- Setting up a BAM neural network;
- Creation and editing of the training set;
- Creation and editing of the pattern to be recognized;
- Weight matrix calculation (training) and visualization;
- Choice of the input layer;
- Signal propagation;
- Visualization of the signal propagation history;
- Work with more BAM neural networks;
- Using the main unit in other software applications.

SETTING UP THE NEURAL NETWORK AND THE TRAINING SET

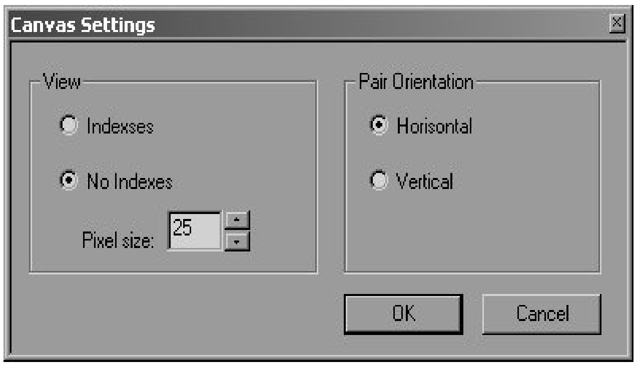
The number of the nodes in each of the two layers in the neural network can be chosen by the user (Fig. 2a) or can be loaded from a file previously created with the same simulator.



New

BAM Settings
Elements in X Layer: 16
Elements in Y Layer: 9

Files
Project name: test
Location: C:\Homemade\



Canvas Settings

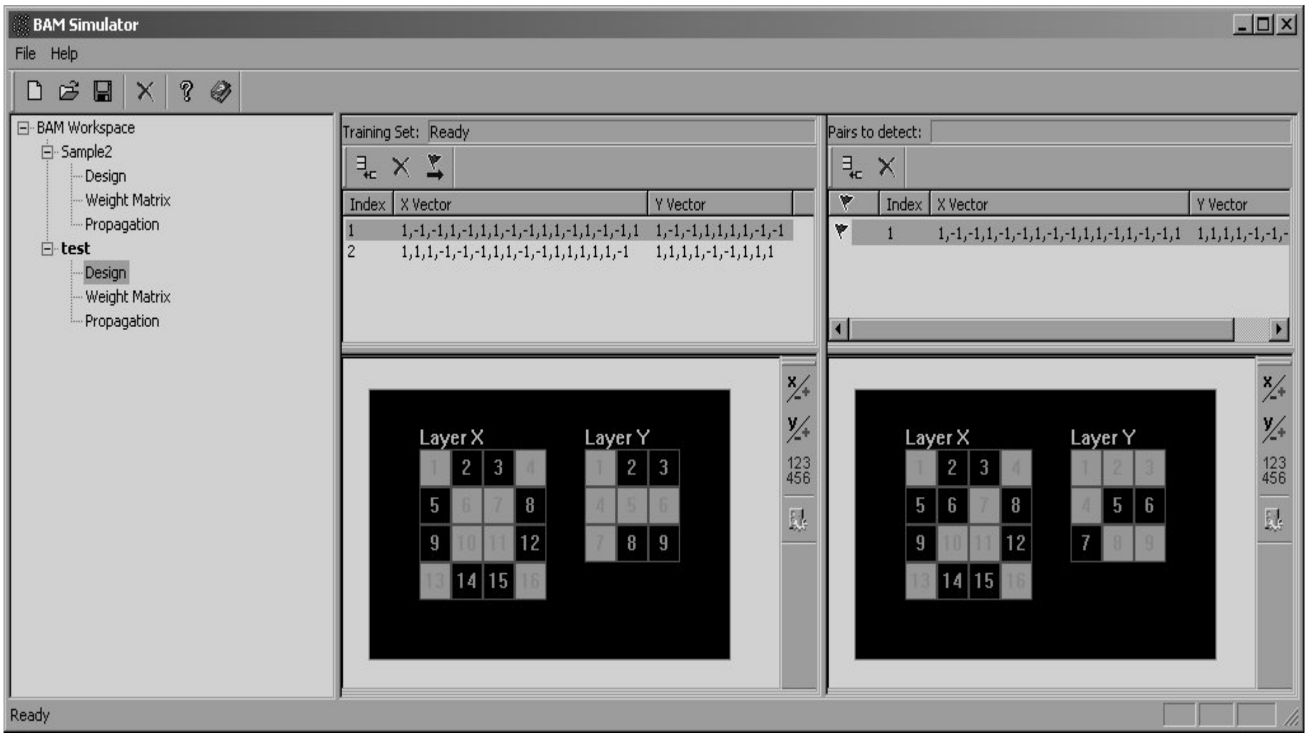
View
Indexes [✓ No Indexes]
Pixel size: 25

Pair Orientation
[✓ Horisontal] Vertical

a) b)

Fig. 2 Setting up the neural network structure and the two-dimensional representation

For a better comprehension of the vectors, which have many components with values of 1 or –1, a two-dimensional representation of these vectors is also used (Fig. 3). In the two-dimensional representation is possible after pushing the button on user's choice (Fig. 2b) to visualize or not the position numbers of the nodes in each layer, to change the dimensions of the kernel component or to exchange the mutual position of the x and y vector.



BAM Simulator

Training Set: Ready

Index	X Vector	Y Vector
1	1,-1,-1,1,-1,1,1,-1,-1,1,1,-1,1,-1,-1,1	1,-1,-1,1,1,1,1,-1,-1
2	1,1,1,-1,-1,-1,1,1,-1,-1,1,1,1,1,1,-1	1,1,1,1,-1,-1,1,1,1

Pairs to detect:

Index	X Vector	Y Vector
1	1,-1,-1,1,-1,-1,1,-1,-1,1,1,-1,1,-1,-1,1	1,1,1,1,-1,-1,-

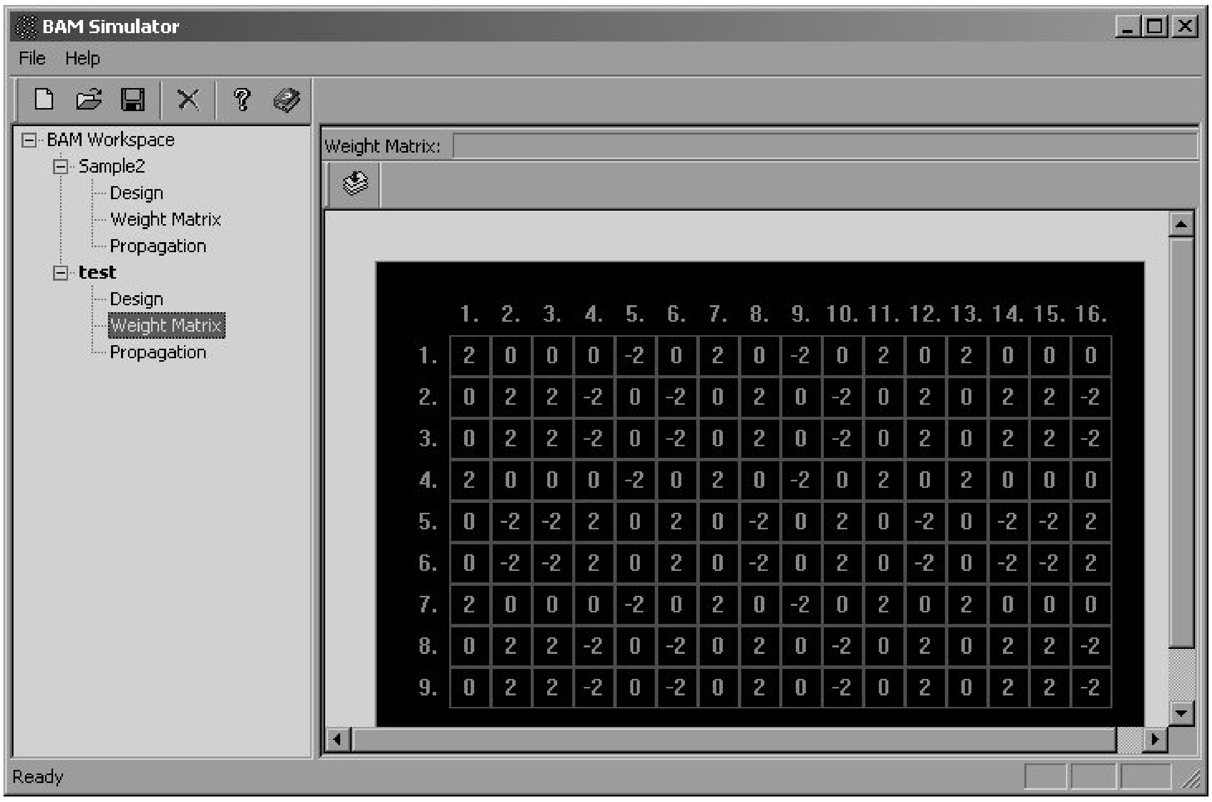
Fig. 3 Creation of the vector pairs in the training set and choosing the initial state

By pushing the button for adding a new x-y vector pair to the training set, the components of the x – and y - vector receive initial values of –1 and simultaneously all components of the vectors are visualized by a dark fields in the two-dimensional representation. The fields' values can be edited with the mouse in which case the fields in the two-dimensional representation corresponding to +1 – components in the vector pair become lithe. All components of the so received x- or y vector can be inverted by pushing the button or . The vector pair deletion from the training set can be performed also interactively.

Could be defined many variants for the initial state of the neural network but the actual selected one is marked with a flag in the frame "Pairs to detect" (Fig. 3). Very useful is the possibility to copy some of the training pairs from the frame "Training Set" to the frame "Pairs to detect" and after that to change some of the fields in the vectors, producing the desired closeness to the initial state of the neural network.

NETWORK TRAINING

The training process starts with a pushing the button . The result is the calculated weight matrix and it can be shown with the simulator (Fig. 4).



	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
1.	2	0	0	0	-2	0	2	0	-2	0	2	0	2	0	0	0
2.	0	2	2	-2	0	-2	0	2	0	-2	0	2	0	2	2	-2
3.	0	2	2	-2	0	-2	0	2	0	-2	0	2	0	2	2	-2
4.	2	0	0	0	-2	0	2	0	-2	0	2	0	2	0	0	0
5.	0	-2	-2	2	0	2	0	-2	0	2	0	-2	0	-2	-2	2
6.	0	-2	-2	2	0	2	0	-2	0	2	0	-2	0	-2	-2	2
7.	2	0	0	0	-2	0	2	0	-2	0	2	0	2	0	0	0
8.	0	2	2	-2	0	-2	0	2	0	-2	0	2	0	2	2	-2
9.	0	2	2	-2	0	-2	0	2	0	-2	0	2	0	2	2	-2

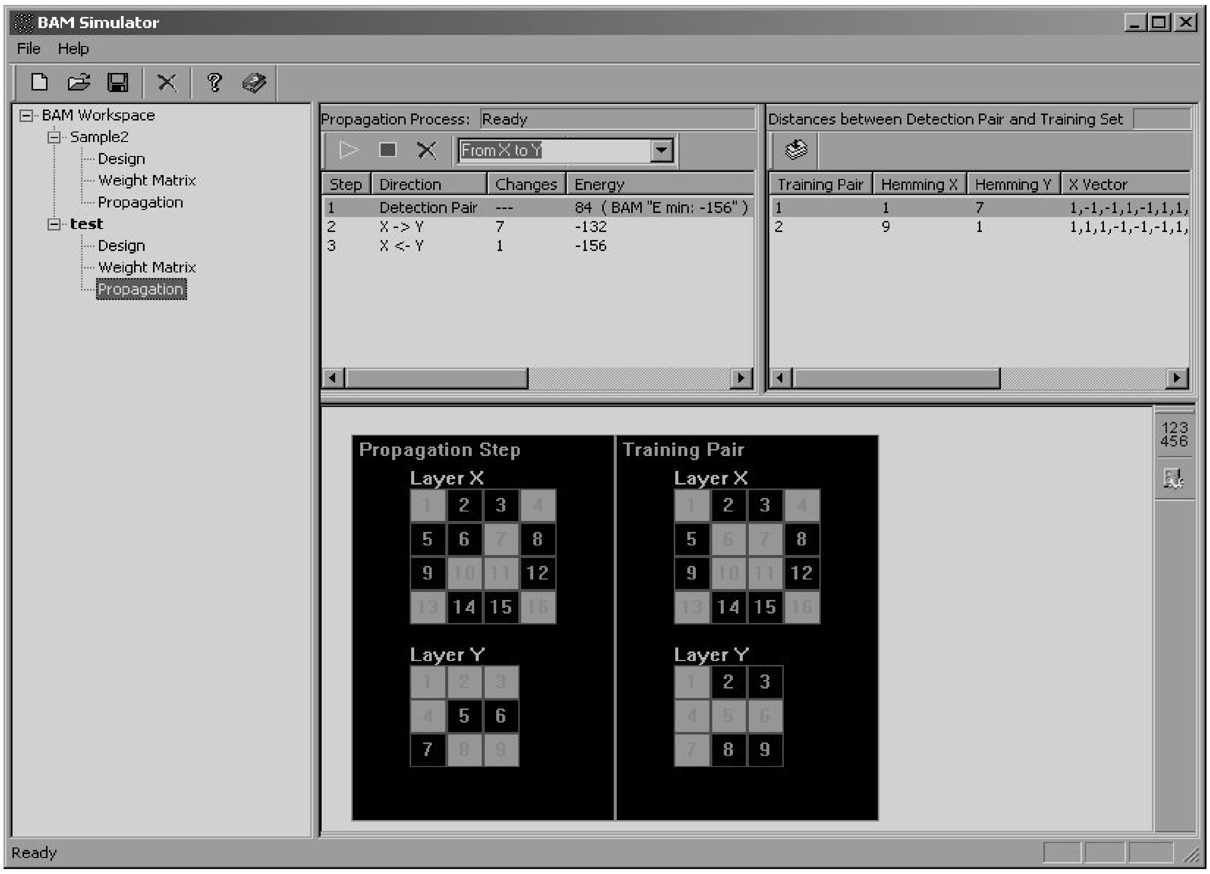
Fig. 4 The weight matrix – the result of the training process

RECOGNITION PROCESS

After the choice of the input layer, the user starts the recognition process with pushing of the button . At each step of the identifying process the currently direction of the signal propagation and the value of the calculated network energy are displayed (Fig. 5 - frame "Propagation Process"). The last row from the list of steps in the same frame (step 3 including the initial state in the case shown) is the result from the recognition.

In the frame on the right side (Fig. 5 - frame "Distances between Detection Pair and Training Set") is visible the closeness (Hamming distance) between the chosen initial state of the layers in the neural network and every vector pair in the training set.

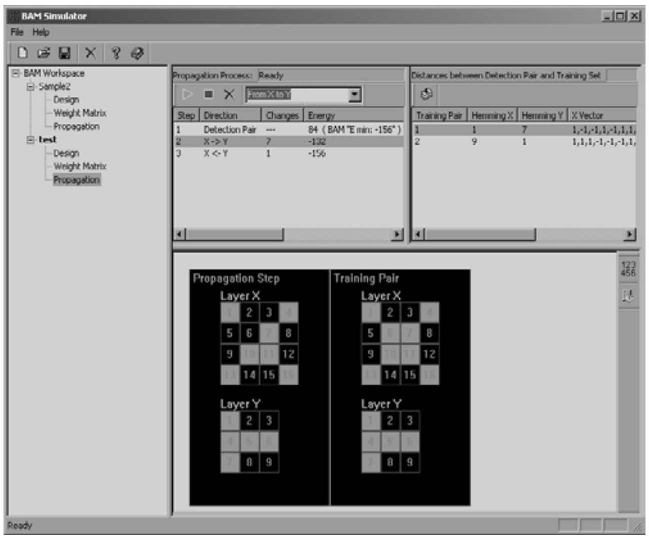
The history of the signal flow is recorded step by step. When a step with the mouse is selected, on the left side of the bottom right frame containing the two-dimensional representation of the vectors (Fig. 5 frame "Propagation step") the actual state of the both layers for this step is shown. In the right part of the same frame can be seen the chosen with the mouse vector pair of the training set. The last possibility give a chance to compare the result of the recognition with the expected one based on the Hamming distances.

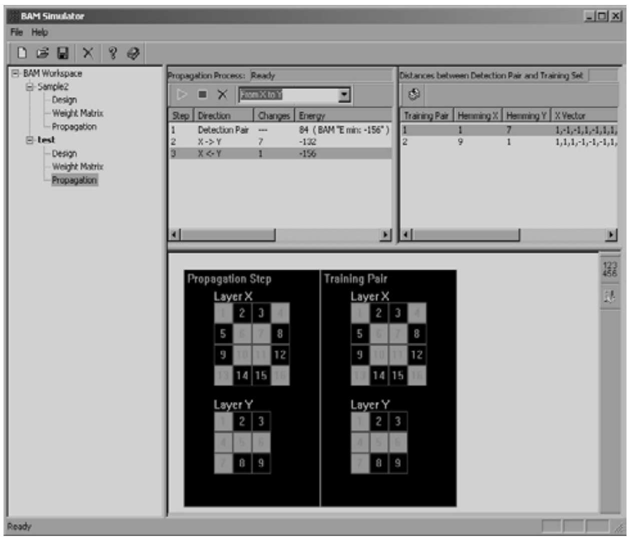


Step	Direction	Changes	Energy
1	Detection Pair	---	84 (BAM "E min: -156")
2	X -> Y	7	-132
3	X <- Y	1	-156

Training Pair	Hemming X	Hemming Y	X Vector
1	1	7	1,-1,-1,1,-1,1,1,
2	9	1	1,1,1,-1,-1,-1,1,

a) Initial state of the neural network





b) After the first signal propagation step c) After the second signal propagation step

Fig. 5 Recognition process and visualization of the history

CONCLUSIONS AND FUTURE WORK

The use of graphical user interface in the simulator forces the students to focus on the core of the learned matter.

The possibilities for change of all parameters in the neural network contribute to the rationalizing of the teaching material.

The possibility to include the main module as a part of other software applications extends the application field of the implemented simulator.

REFERENCES

[1] J.A. Freeman, D.M. Skapura, Neural Networks Algorithms, Applications and Programming Techniques, Addison-Wesley Publishing Company, 1992

ABOUT THE AUTHORS

Assoc. Prof. Julia Stojanova Zlateva, Department of Computer systems and technologies, University of Rousse “Angel Kanchev”, Phone: +359 82 888 681
E-mail: Jzlateva@ecs.ru.acad.bg

Georgi Todorov, E-mail: georg_todorov@abv.bg