World Applied Sciences Journal 7 (8): 1071-1076, 2009
ISSN 1818-4952
© IDOSI Publications, 2009

Parallel Processing of ART1 Neural Network Algorithm and Application for Recognition of Color Images

[1]Rashad J. Rasras, [2]Ibrahim E.L. Emary and [1]D.E. Skopin

[1]Faculty of Engineering Technology/ Albalqa' Applied University, Jordan
[2]Faculty of Information Technology, King Abdulaziz University, Jeddeh, Kingdom of Saudi Arabia

Abstract: It is well known, that standard ART1 network may process only binary images. In this paper we develop a method for a parallel processing of ART1 neural network algorithm, which allow to recognition of color images and speed up of the image processing and recognition.

Key words: Neural networks . parallel processing . image recognition

INTRODUCTION

In work [19] The increment of color depth (number of bits per single pixel) allows the ART1 network to recognize more colors in one hand but in other hand it causes increasing of network structure, computation complicity and processing time needed for image recognition in sequential implementation of learning algorithm and recognition process. To reduce processing time, we have converted a sequential structure of the network to the structure that can be implemented in N numbers of independed processors. In practical point of view we have choose distributed memory systems that can be easily constructed on the *N* sequential computers connected to single tcp/ip local area network using Message Passing Interface (MPI) [18]. The logical view of a machine supporting the message-passing paradigm consists of *p* processes, each with its own exclusive address space. Instances of such a view come naturally from clustered workstations and non-shared address space multicomputer. The MPI is a library for C++ and Fortran compilers that also includes a tool for processors loading, synchronization and data interchange between different members of parallel processing system. The firs step in conversion of ART1 neural network to parallel computing system is network decomposition.

PARALLEL IMPLEMENTATION ARCHITECTURE OF ART-1

Taking into account the fact that loading of each processor should be approximately equal and to reduce number of breaking links, we have decomposed the network according to the Fig. 1. The R is reset unit of the network, b_{ij} are bottom-up weights, t_{ji} are top-down weights.

Let denote as n-size of input vector of ART1 neural network, m-maximum number of clusters in recognition unit, N-number of tasks (parallel processors). In general case the number of processors less than input vector length, number of clusters in recognition layer less than length of input vector. In general case each processor has m/N cluster nodes and n interface and input nodes. We have add one limitations to the parallel processing algorithm: the ratio m/N should be integer number. The parallel processing systems includes as minimum one processor, for this case we have sequential system. The processor number zero let denote as root, this processor will be a source of data, serve for synchronization of processes, allow to distribute a data between other members of parallel processing system, accept intermediate results, produce and printout final results. Since we have single reset unit, we will allocate this unit in the root, also root will be involved to computations.

The sequential learning algorithm of ART1 is well known in modern literature [1, 2]; here we will explain a modified algorithm for parallel processing system. Standard work with ART1 neural network includes 2 main steps: learning the network (execution of training algorithm) and image recognition (clustering algorithm).

The notation we use as follows:

n Number of components in the input vector
m Maximum number of clusters that can be formed
b_{ij} Bottom-up weights (from F_1(b) unit X_i to F_2 unit Y_j)
t_{ji} Top-down weights (from F_2 unit Y_j to F_1 unit X_i)
s Binary input vector
x Activation vector for F1(b) layer rank current Processor's number
N Number of processors

Corresponding Author: Dr. Rashad J. Rasras, Faculty of Engineering Technology/Albalqa' Applied University, Jordan

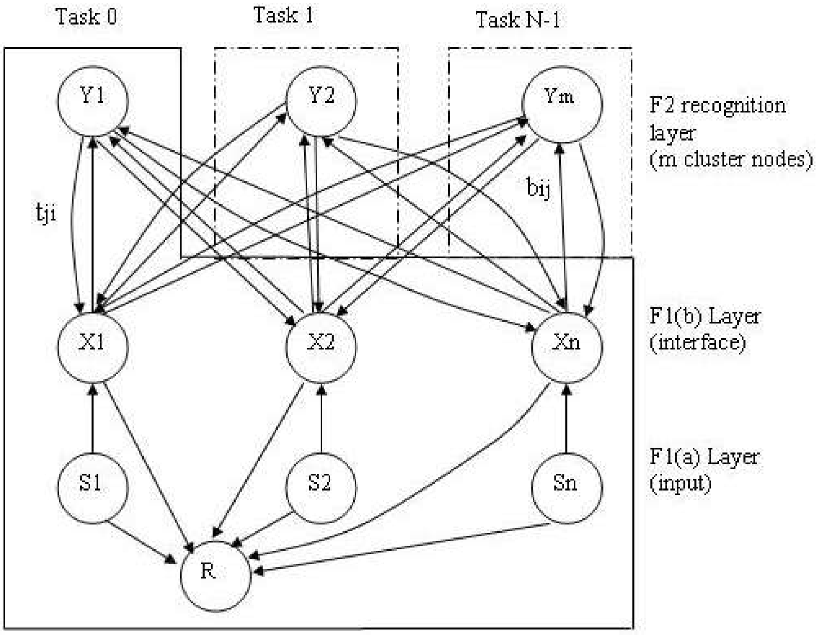


Fig. 1: The decomposition of ART1 network structure into N tasks

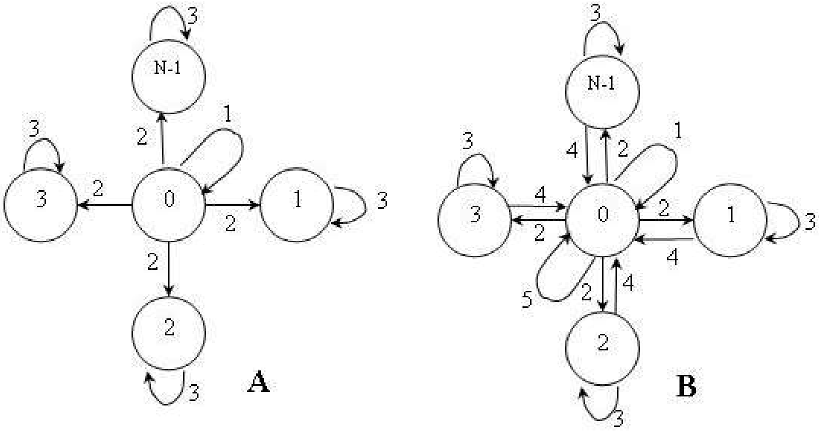


Fig. 2: The structure of parallel data processing during A-training algorithm, B-recognition algorithm

The block diagram above illustrates main steps of training and recognition algorithms:

The Fig. 2(A) illustrates the algorithm of data processing during training of ART1 network. It includes the steps 1-start up of the system, preparing a training set, 2-the one-to-all broadcast of training data set between all members of parallel processing system, step 3-computation of bottom-up and top-down weights, binary input and activation vectors. In the Fig. 6B represented the algorithm of recognition that includes steps: 1-loading image from the file by root processor, 2-one-to-all broadcast of image between other members of parallel processing system, 3-compute responses of cluster units, 4-all-to one reduce operation (each processor return values of cluster units), 5-searching of number of excited neurons and print out the number of patterns. In both cases the steps number one and five are sequential part of algorithms , all other parts are concurrent.

The training algorithm in details:

Step 1: Read the rank (current number of process), if rank is zero (process is root) load a training set from the file(s).

Step 2: Broadcast a training set to all other members of parallel processing system.

Step 2: If rank not equal zero receive training set from root (length of vector is n). Set activation of all F2 units to zero, $F_1(a)$ units to input vector s, compute the norm of s, send input signal from $F_1(a)$ to the $F_1(b)$ layer, for each F_2 node compute

$$y_j = \sum_i b_{ij} \cdot x_i$$

while reset is true recompute activation x of F1(b): $x_i = s_i \cdot t_{ij}$, compute the norm of vector x, update top-down and bottom-up weights.

Note: In sequential algorithm the top-down and bottom-up weights represented as matrices dimension n by m, in parallel realization of training algorithm we have other dimensions that equal n by m/N, where N-number of processors, so, computation and update of matrices takes N times less iterations.

The recognition algorithm in details:

Step 1: Read the rank, if number equal zero (means root processor) load the image from file.

Step 2: Broadcast the image between all other members of parallel processing system

Step 3: Compute responses of cluster units, for all processors calculate the number of cluster units (inside F2 recognition units) as m/N per single processor, read the current processor's number (variable "rank")

```
for j=N1 to N2 do external loop, step is 1
    for i:=1 to n do internal loop, step is 1
    begin
        y[j]:=y[j]+b[j,i]*x[i];
    end;
```

As we can see, computation of clusters condition y_j takes N time less iterations to compare with sequential algorithm. Each processor covers the range of cluster units according to the equation below:

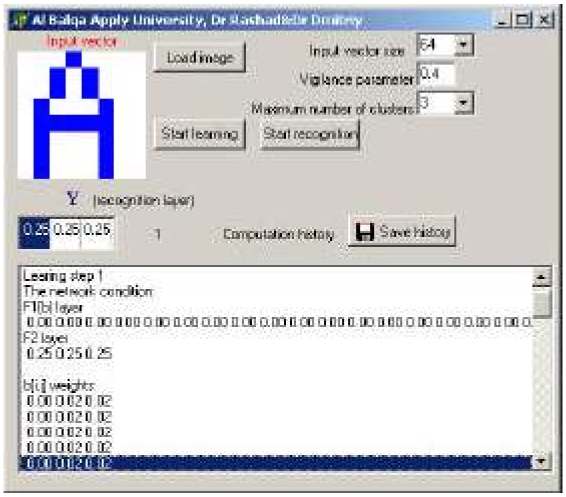
$$N1 = \frac{m}{N} \cdot \text{rank};$$

$$N2 = \frac{m}{N} \cdot (\text{rank} + 1) - 1$$

Step 4: All-to one reduce operation, during this step each processor return values of cluster units y_j to the root, that gather required information to single vector (length equal m)

Step 5: The root analyze a vector $y_{j,}$ search the maximum value inside the vector (number of excited neuron) and print out the number of pattern

Experemintal and practical results:



Input vector size 64

Vigilance parameter 0.4

Maximum number of clusters 3

0.25 0.25 0.25 1

```
Learing step 1
The network condition
F1(b) layer
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.
F2 layer
 0.25 0.25 0.25

b[i,j] weights
 0.00 0.02 0.02
 0.00 0.02 0.02
 0.00 0.02 0.02
 0.00 0.02 0.02
 0.00 0.02 0.02
```

Fig. 3: GUI of designed program

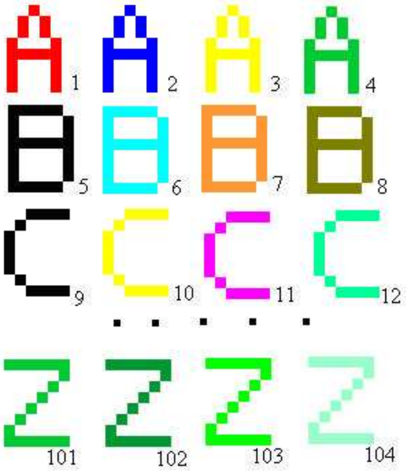


Fig. 4: The training and recognition set

The practical part of current work we have divided into two parts: testing ability of ART1 network to recognize color images (different forms and colors of patterns) and testing and speed up evaluation for parallel implementation of training and recognition algorithms. To study ability of the ART1 neural network in recognition of color images the program implemented a color interface of ART1 neural network has been created using Borland Delphi 6.0 compiler [15, 16]. The graphical user interface of elaborated program represented in the Fig. 3.

The program allows to load up 200 input vectors with different vigilance parameter in the range 0.1...1 with maximum number of clusters equal 200. To control a learning and recognition process the full computation history list with step by step explanation of calculation sets were included to the program. During experimental work we chose input vector size equal

Table 1: Testing the recognition ability of the network

Test number	Vigilance parameter	Learning sequence	Input sequence	The network outputs (Number of cluster)	Note
1.	0.4	1,2,3,4	1,4,3,2	1,3,3,2	Small vigilance parameter caused clustering into 3 clusters
2.	0.75	1,2,3,4	1,4,3,2	1,4,3,2	All patterns recognized correctly
3.	0.75	5,6,7,8	8,6,5,7	5,6,5,7	Pattern 8 (dark grey) recognized as pattern 5 (black)
4.	0.75	101,102, 103,104	104,101, 103, 102	102,101, 101,101	All patterns clustered into two clusters (see explanation below)

64 pixels to be able to process images with 8X8 bits size. The structure of ART1 interface that illustrated in the Fig. 1 used during experimental work. For simplification a work and analysis we chose three-bits per pixel color encoding.

For testing the accuracy of neural network recognition the set of samples representative letters of the Latin alphabet (A... Z) have been constructed. The set included 26 different chars, 4 random colors per each char, it's totally 104 symbols located in external graphical files (BMP format). The appearance of sample chars inside the set showed in Fig. 8, note that each char was numbered (Fig. 4).

Experimental work has been broken into two parts: learning of a neural network and recognition. During learning the training set (chars recorded in BMP files numbered in ascending order) forwarded to the network input. During recognition initial numbers of files were shuffled in random order and then sent to the input again. The output of the network (one output per one input file) is the number of recognized cluster.
The input parameters of the network:

1. Input vector size = 64
2. Vigilance parameter (was vary from 0.4 up to 0.8)
3. Maximum number of clusters = 104
4. The source of input image-external bitmap files

The output of the network : number of cluster (or clusters if input char belong to few learning images)

We have provided two researches, first of them purposed to estimate abilities of the network to recognize the patterns with same object by different colors and second one to estimate statistical properties of the network at various values of vigilance parameter. The result of first research represented in Table 1, input sequence means the numbers of input samples (Fig. 5).

In first test we used lower vigilance parameter equal 0.4 and result shows that ever in low vigilance it is possible to recognize the color of pattern. In this test the red, green and blue A chars recognized correctly but yellow char (sample number 3) clustered as green (sample number 4). We can explain this fact that binary representation of red, green and blue color is 100,010,001 accordingly (all bits is different), yellow 110-first bit cross linked with middle bit of green color. In second test all chars recognized correctly due moderate vigilance parameter. The third test shows ability of network for color interpolation, so pattern 8 (dark grey) interpolated as pattern 5 (black). The number four also shows color interpolation by color interface, test patterns 101-103 interpolated as green, last pattern number 104 recognized as white color because source color too bright (Fig. 4).

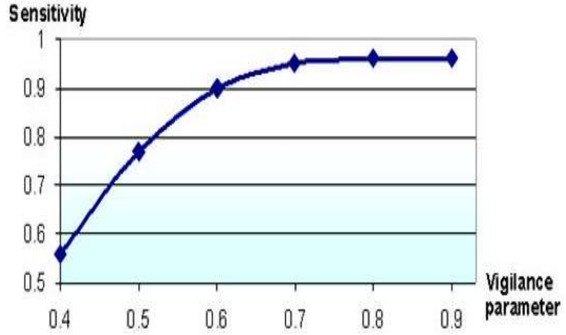


Fig. 5: Relation between the value of vigilance parameter and sensitivity of recognition

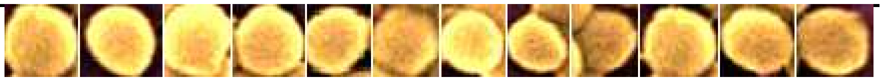
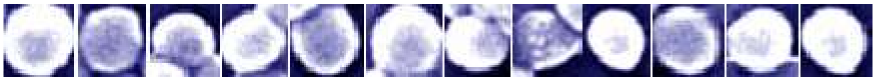
To study the recognition accuracy of the network we generated additional set of Latin alphabet (A...Z), but only native colors were included to this set (red, green, blue, yellow). The statistical parameter sensitivity calculated according to equation 1 [13, 14]

$$\text{Sensivity} = \frac{\text{Number of true positives}}{\text{Number of truepositives} + \text{Number of false Negatives}} \qquad (1)$$

The results shows that the network is able to recognize different objects in vigilance parameter ranged 0.7-0.8. Less value of vigilance reducing sensitivity of network, on the other hand moderate vigilance cause increment of time for training and recognition algorithms.

Second part of research was dedicated to study a properties of parallel implementation of ART1 structures in distributed memory parallel processing

Table 2: The training set for recognition of microscopic images

Description of the object	Number of training sets	Images
Staphylococcus aureus	1-12	
Streptococcus pyogenes	13-24	

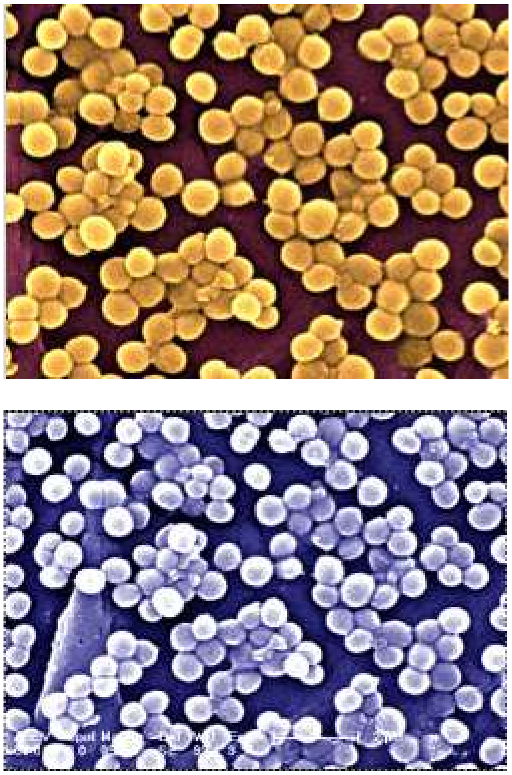


Fig. 6: Microscopic images of gram-positive bacteria: Staphylococcus aureus (left image) streptococcus pyogenes (right image). Zoom equal 1500

systems. We have used MPICH system (version 1.2.5 for NT) to construct parallel processing cluster. The cluster includes 8 sequential computers. The program implemented a parallel realization of ART1 training and recognition algorithm that was created by Visual C++ 2005 compiler, console application, the model of the parallel processing system-SPMD [1]. The program was designed to count the numbers of microorganisms in high resolution microscopic images [17] Fig. 6.

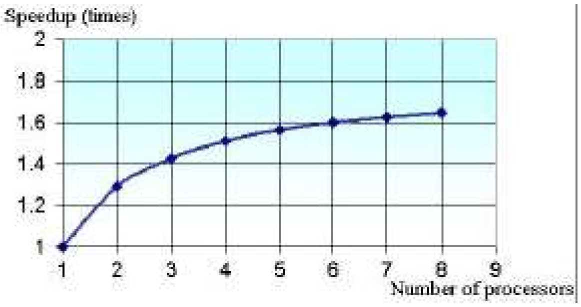


Fig. 7: Relation between number of processors and speedup of parallel program

On the base of source images a training set for ART1 neural network was formed. As we can see in Fig. 6 both bacteria has approximately same forms but different colors according to different Gram's method [17]. We have chosen 12 patterns per each image (patterns number 1-12 for left image, 13-24 for right image). Each pattern contains the images resolution equal 36X36 pixels, the length of input vector equal 3888 bits. The training set for ART1 neural network represented in the Table 2.

The program was distributed between all computers of parallel processing systems, evaluation of time for training and recognition algorithms of the network was done for number of processors inside cluster are: 1 (single sequential computer), 2 (distributed parallel processing system), 3, 4, 6, 8. The cases for number of processors equal 5 and 7 was skipped according to the limitation of parallel algorithms (see previous chapter). The speedup of parallel processing system was evaluated according to the equation 2:

$$\text{SpeedUp} = \frac{\begin{array}{c}\text{Time of task execution}\\ \text{for sequential program}\end{array}}{\begin{array}{c}\text{Time of task executiin}\\ \text{for parallel program}\end{array}} \qquad (2)$$

where sequential program is particular case of parallel processing system for number of processors equal 1. Evaluation of execution time was done according to standard MPI function MPI_Wtime(), measurement of times was provided after broadcast of training sets and recognizable images. Relation between speedup and number of processors in parallel system represented in the Fig. 7.

CONCLUSION

It is known, that the standard ART1 network is intended for work with a binary input vector [1, 2] and this fact greatly reduce a contribution of ART1 structures in color image recognition because real images represented in color format and conversion to binary form caused information loss. We have shown that by bit slicing method is possible to represent a color pixel as the set of binary numbers and use this binary sequence as input of neural network.. For three bits color encoding, ART1 neural network has interpolation ability that allow to interpolate any input color to the nearest base color listed in Table 1. We have tested ability of the network for recognition of color images that gave coefficient of sensitivity approximately equal one in case of recognition of the objects with different forms and colors. The network also was implemented in parallel program of image recognition to count a number of different microorganisms that has approximately the same form but different colors. As researches showed speedup (time profit) of our parallel program become 1.62 for 8 processors system. The future researches is actual to concentrate in the problem of further speedup increment in example of shared memory systems.

REFERENCES

1. Carpenter, G.A. and S. Grosberg, 1987. Massively parallel architecture for self-organizing Neural pattern recognition, Machine// computer vision. Graphics and Image Processing, 37: 45115.
2. Faulset, L., 1994. Fundamentals of Neural Networks. Architectures, Algorithm and Applications. New Jersey, Prentice Hall Int., pp: 461.
3. Moore, B. and D. Touretzky, 1998. ART1 and pattern clustering, in //Proceedings of the connectionist models summer school. Morgan Caufmann, LosAltros, CA, pp: 174-185.
4. Blake, R.E., 1987. Development of an Incremental Graph Matching Device, Pattern Recognition Theory and Applications, NATO ASI Series, Springer, Berlin, 30: 355-366.
5. Toru Nakanishi and Yuji Sugiyama, 2004. Sealed-bid Auctions with Efficient Bids Using Secure Bit-slicing Conversion. IEICE Trans. Fundamentals, E87-A (10): 2533-2542.
6. Peter M. Maurer and William J. Schilp, Software Bit-Slicing: A Technique for Improving Simulation Performance\\Design, Automation and Test in Europe (DATE '99) p: 786.
7. Gonzalez, R.C. and R.E. Woods, 1992. Digital Image Processing. Reading, Massachusetts: Addison-Wesley, pp: 716.
8. Castleman, K.R., 1996. Digital Image Processing. Second Ed. Englewood Cliffs, New Jersey: Prentice-Hall.
9. van der Heijden, 1994. Image Based Measurement Systems, Wiley.
10. Schachter, B.J. and L.S. Davis, 1979. A. Rosenfeld, Some experiments in image segmentation by clustering of local feature values, Pattern Recognition, 11: 19-28.
11. Skopin, D.E., 2005. Optical topographer for diagnosis of support-locomotion system diseases //"Medical and ecological information technology. Kursk, pp: 48.
12. Liu, J. and Y.H. Yang, 1994. Multiresolution color image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 16 (7): 689-700
14. Chatfield, C., 2005. Statistics for Technology: A Course in Applied Statistics. Third Edition CRC press. ISBN: 0412253402
15. Kerman, F. and C. Miitchell, 2004. Programming and Problem Solving With Delphi. UK INTERNATIONAL ED, Pearson Higher Education Date Published: ISBN: 0321204417.
16. Rachele, Warren Learn Object Pascal with Delphi. Wordware Publishing inc. ISBN: 1556227191.
17. Madigan, M.T., J. Martinko and J. Parker, 2004. Brock Biology of Microorganisms, 10th Edition, Lippincott Williams and Wilkins. ISBN 0-13-066271-2.
18. Ananth Grama, Anshul Gupta and George Karypisp, 2003. Introduction to Parallel Computing, Second Edition Addison Wesley 0-201-64865-2.
19. Rashad, J. Rasras, Ibrahim E.L. Emary and D.E.E. Skopin, 2009. A new ART-1 neural network interface model adopted for color image recognition. Scientific Bulletin of University of Pitesti, series Electronic and Computer Science, Rominia, Vol: 1.