Volume 2, Issue 3, March 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Competitive Neural Network as Applied for Character Recognition

Apash Roy*, N. R. Manna
Department of Computer Science and Application,
University Of North Bengal, Darjeeling – 734013, W.B., India
mailaposh@gmail.com

Abstract- **Application of artificial neural network in the field of character recognition of printed or handwritten characters is already into its ease of popularity. Here an attempt has been made to recognize characters with the help of competitive neural network. The system consists of a network of competitive neurons, which can classify deviated patterns after training. It is found that the number of iteration for the training is surprisingly the number of different characters to be identified by the system if the value of learning constant is greater than 0.5. The results are quite satisfactory and high percentage of recognition is achieved.**

Keywords - **Artificial Neural Network, Pattern recognition, Character recognition, Competitive learning,**

1. Introduction

During a long time, recognition of handwritten characters for several languages is a popular topic of research for various researchers. Identification of different characters, handwritten or printed, remains a challenge till now.

The works in the field of character recognition were done by several probabilistic methods like Bayesian decision theory [4, 5, 6], k-nearest neighbour (k-NN) algorithm [4] etc.

Application of neural network [2, 3, 8] added a new dimension in this field. Several models of neural network were used for this purpose including Multilayer Perceptron (MLP)[13], back-propagation network[14], Multiscale Training Technique (MST) [4,10], Multiscale Training using competitive learning [17], row wise segmentation technique (RST) [18].

In this paper, an attempt has been made to explore the idea of competitive neural network to recognize handwritten characters. The proposed system consists of a network of competitive neurons, which can easily classify deviated patterns also after a proper training by some standard pattern.

2. Competitive learning

Competitive learning [1, 2] is one of the popular unsupervised mechanisms to train a network of processing units to achieve some pattern recognition task. Generally, a competitive learning network checks for a winner unit with maximum activation value. The weight vector of the winner is adjusted to bring its component closer to the input vector components and output for every winning unit is set to 1 while others are set to 0.

A vector of input X_i , i=1, 2, 3…... passed into the layer of every unit U_j, j=1, 2, 3…... with the synaptic weights W_{ij} (Fig. 1). According to the diagram, U_2 is the winner with maximum activation value. For this the weights have to be adjusted accordingly. The new weight vector for unit U_2 may be formed using the following equation:

$$W_{i2}(new) = W_{i2}(old) + c(X_i - W_{i2})$$

here,'c' is a small positive learning constant. Weights are to be adjusted in such a way that the angular distance between input vector and the weight vector for the winning unit is reduced.

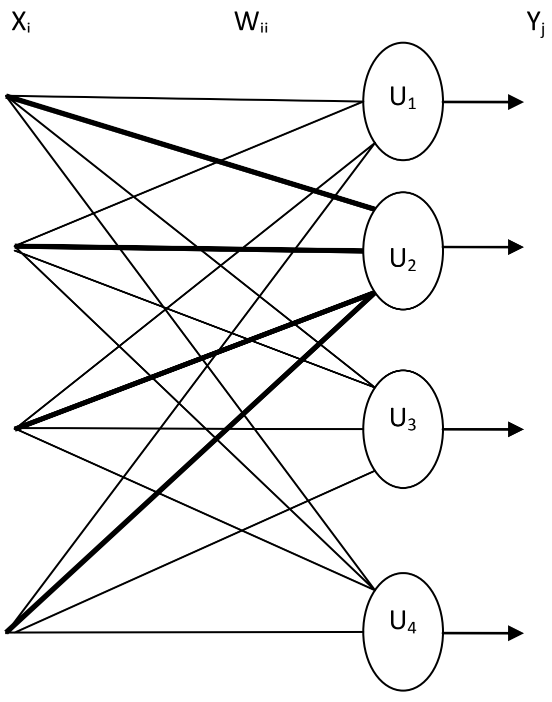


Fig 1: Competitive learning

.

3. Pre-processing

Character recognition through neural network needs several pre-processing. It includes the required steps to shape the input image into a form, suitable for processing. Here, input characters are generally written on some clean paper, within a standard size. These characters are captured by digital camera or a scanner. The captured characters are subsequently converted into uniform sized binary matrix, which will become the training or testing vector.

In this work, a 5X5 binary matrix has been used to represent each character of English alphabet as in Fig. 2.

Thus, input vector for **A** has been represented as 00100 01010 10001 11111 10001 and **X** as 10001 01010 0010001010 10001.

(a)	(b)
00100	**10001**
01010	01010
10001	00100
11111	01010
10001	**10001**

Fig 2: Binary vector for English character (a) **A**, (b) **X**

4. The Method

At the end of pre-processing, thus obtained 25 (5x5) different binary elements of input vectors for a particular character are fed into a network of processing units. The processing units are connected with each other as shown in the Fig. 3. Every element from the input layer to the processing units is connected by the weight vectors $\mathbf{W_{ij}}$, where i represents the number of connections from the input layer and j represents the jth processing unit. Initially its value is assigned randomly for each unit.

To find the maximum activation value, activation values for every unit are to be calculated directly by summing up all the weighted inputs ($\sum W_{ij}x_i$ with respect to a particular unit j. This process is equivalent to calculating the minimum Euclidian distance [1] between W_{ij} and X_i for a particular unit j, such that

$$\mathbf{// X_i - W_{ij} // \quad = Min (\sqrt{} (X_i - W_{ij})^2)}$$

Or

$$\mathbf{Min ((X_i - W_{ij})^2)}$$

Once the winning unit is found, its weights are adjusted and output is set to 1.Throughout the learning session weights of different units are updated several times. When learning is complete for all training sets then the system is ready to identify the characters. Efficiency of learning depends on the size of character set i.e. the number of characters took part in the learning process. Usually, more the numbers better the performance. So, a number of input vectors are taken for each character in order to increase the efficiency of the system through learning.

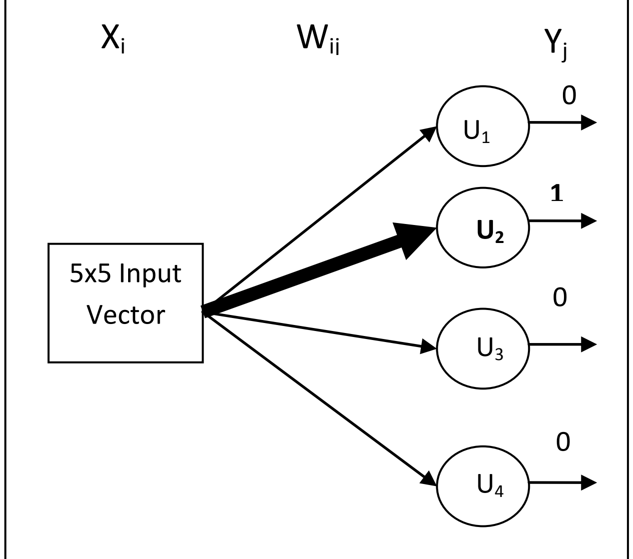


Fig3: Competitive learning network for character recognition

The learning process for the system lasts for the given input set repeatedly until a satisfactory condition is satisfied. A counter p is taken to count the number of iterations required for the complete learning. i.e. no possibility of changing output or weight vector.

The learning constant c has been taken with value 1 while adjusting the weights. So, actually the new weight vector components for the winner units are adjusted by value of the input vector component itself.

5. Experiment Results

The algorithm stated has been implemented in Borland C++ and tested for several characters of English alphabet. The testing alphabets are chosen in a specific manner. At first they are divided into two different groups. One which constructed with only some straight lines like A, E, F, H, I, K, L, M, N, T, V, W, X, Y, Z. While others like B, C, D, G, J, O, P, Q, R, S, U are built with curves. Now, Three character from each group are chosen randomly as test characters. The characters are A, H, E , B, G, O. After preprocessing, these are fed into the neural network for training. According to Table 1. the learning constant c produces single iteration for learning if the value is greater than 0.5. that is why its value has been taken as 1 for the experiment.

Table 1: Result with different values of learning

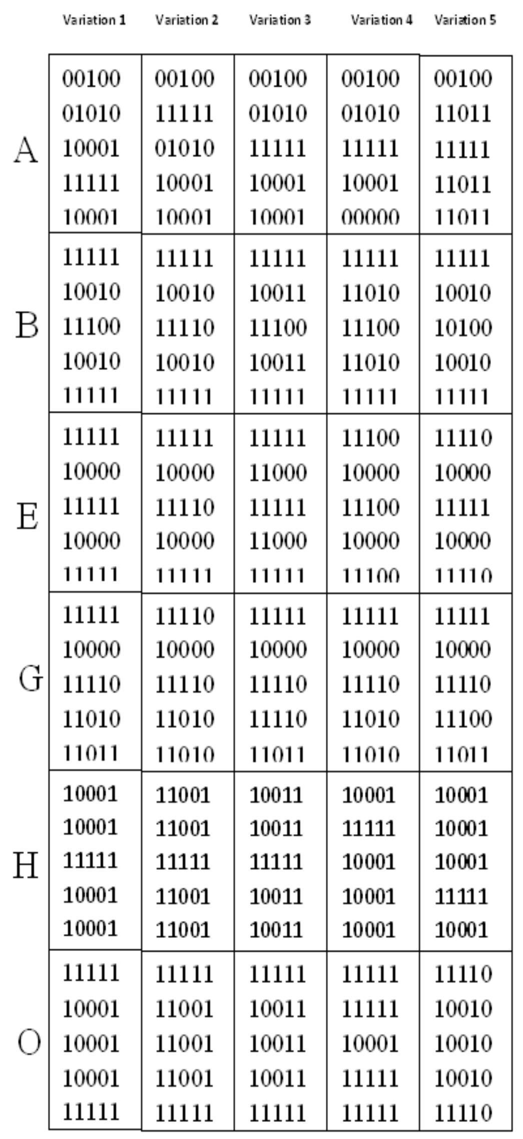
Value of 'c'	No of input characters	Iteration
0.01	6	67
0.02	6	17
0.03	6	12
0.04	6	8
0.05	6	8
0.06	6	7
0.07	6	5
0.08	6	6
0.09	6	4
0.1	6	4
0.2	6	3
0.3	6	2
0.4	6	2
0.5	6	1
0.6	6	1
0.7	6	1
0.8	6	1
0.9	6	1

Results of trained network on the characters as given in Table 3, are shown in Table 2.

Table 2: Test Results

Character	No. of Variation	Identified
A	5	4
H	5	4
E	5	4
B	5	5
G	5	5
O	5	3

Table 3 Different variations of testing characters



	Variation 1	Variation 2	Variation 3	Variation 4	Variation 5
A	00100 01010 10001 11111 10001	00100 11111 01010 10001 10001	00100 01010 11111 10001 10001	00100 01010 11111 10001 00000	00100 11011 11111 11011 11011
B	11111 10010 11100 10010 11111	11111 10010 11110 10010 11111	11111 10011 11100 10011 11111	11111 11010 11100 11010 11111	11111 10010 10100 10010 11111
E	11111 10000 11111 10000 11111	11111 10000 11110 10000 11111	11111 11000 11111 11000 11111	11100 10000 11100 10000 11100	11110 10000 11111 10000 11110
G	11111 10000 11110 11010 11011	11110 10000 11110 11010 11010	11111 10000 11110 11110 11011	11111 10000 11110 11010 11010	11111 10000 11110 11100 11011
H	10001 10001 11111 10001 10001	11001 11001 11111 11001 11001	10011 10011 11111 10011 10011	10001 11111 10001 10001 10001	10001 10001 10001 11111 10001
O	11111 10001 10001 10001 11111	11111 11001 11001 11001 11111	11111 10011 10011 10011 11111	11111 11111 10001 11111 11111	11110 10010 10010 10010 11110

6. Conclusion:

This work is a small contribution towards the field of character recognition through artificial neural network and may be useful in the field of pattern reorganization where exact number of classes are not predefined. However, the efficiency of the system may be increased by considering two other aspects viz. increased size of the input matrix and some extra layers for training. Feature extraction technique from the input matrix may add some extra efficiency.

References

[1] George F Luger "Artificial Intelligence" Pearson Education Ltd., third Indian Reprint-2003

[2] B. Yegnanarayana [Artificial Neural Network] ,PHI Learning Pvt. Ltd., 2004.

[3] Shashank Araokar [Visual Character Recognition using Artificial Neural Networks]

[4] Velappa Ganapathy, and Kok Leong Liew [Handwritten Character Recognition Using Multiscale Neural Network Training Technique], World Academy of Science, Engineering and Technology 39 2008

[5] Wu, P.H. (2003), Handwritten Character Recognition, B.Eng (Hons) Thesis, the School of Information Technology and Electrical Engineering, the University of Queensland.

[6] Liou, C.Y. & Yang, H.C. (1996), "Hand printed Character Recognition Based on Spatial Topology Distance Measurement", IEEE Transactions On Pattern Analysis and Machine Intelligence, Vol. 18. No. 9, pp 941-945.

[7] Brown, E.W. (1993), Applying Neural Networks to Character Recognition, Available:

[8] M. Greenberger, 1962[Management and the Computer of the Future].http://www.ccs.neu.edu/home/feneric/charrecnn.html (Accessed: 2007, October 11th).

[9] Robinson, G. (1995), The Multiscale Technique, Available: http://www.netlib.org/utk/lsi/pcwLSI/text/node123.html (Accessed: 2007, October 11th).

[10] Handwritten Character Recognition, Available: http://tcts.fpms.ac.be/rdf/hcrinuk.htm (Accessed: 2007, October 11th).

[11] Rivals I. & Personnaz L. A statistical proecedure for determining the optimal number of hidden neurons of a neural model. Second International Symposium

on Neural Computation (NC.2000), Berlin, May 23-26 2000.

[12] Bishop M. (1995). Neural Networks for Pattern Recognition. Oxford Univ. Press, Oxford-U.K.

[13] LeCun Y., Bottou L., Orr G. B., Muller K. R. (1998a). Eficien backprop. In Orr G. and K. Miller, editors, Neural Networks: Tricks of the Trade. Springer.

[14] Alexander J. Faaborg, (May 14, 2002) [Using Neural Networks to Create an Adaptive Character Recognition System]

[15] Bansal, V., Sinha, R.M.K., [Partitioning and Searching Dictionary for Correction of Optically Read Devnagari Character Strings], Document Analysis and Recognition, 1999. ICDAR'99, Proceedings of the Fifth International.

[16] Apash Roy, N R Manna, "Character Recognition using Competitive Neural Network with Multi-scale training", UGC Sponsor National Symposium on EMERGING TRENDS IN COMPUTER SCIENCE (ETCS 2012) on 20-21 January 2012.

[17] Rakesh Kumar Mandal, N R Manna; "Hand Written English Character Recognition using Row-wise Segmentation Technique (RST)", International Symposium on Devices MEMS, Intelligent Systems & Communication (ISDMISC) 2011 Proceedings published by International Journal of Computer Applications® (IJCA).