Hand Written English Character Recognition using Row-wise Segmentation Technique (RST)

Rakesh Kumar Mandal
University of North Bengal
PO: NBU, Distt: Darjeeling
West Bengal

N R Manna
University of North Bengal
PO: NBU, Distt: Darjeeling
West Bengal

ABSTRACT
Due to the limitations in single layer ANN researchers started losing interest in ANN during 1970s. Later on the development of multiple layer neural networks led to the development of many efficient techniques to recognize hand written/printed characters with great accuracies and also making the technology complex and costly. In this paper an effort was made to recognize hand written English alphabets using single layer ANN. This approach makes the ANN simple, easy to implement and understand. Row-wise segmentation technique was developed and used here to achieve optimum accuracy. This paper is an approach to develop a method to get the optimized results using the easily available resources. Row-wise segmentation helps to extract out some common features among distinct handwriting styles of different people.

General Terms
Character recognition, Input pattern matrix, Weight matrix, Target group, Row wise Segmentation, Training, Learning rule, Distorted patterns, Counter.

Keywords
ANN, Segmentation, Perceptron, Pattern Recognition

1. INTRODUCTION
In this fast corporate world it has become cumbersome to waste time in getting the documents typed from a typist. Hand written character recognition software recognizes the characters written by an executive or somebody else. But it is very difficult to design software, which is capable of identifying the characters with great accuracy. Basically, identification of the characters is one type of pattern recognition process. Variation in handwriting leads to great difficulty in identifying the character patterns. Different writing styles lead to the distortion in patterns from the standard patterns used to train the network, giving false results. A strong generalization method is required to identify the distorted patterns. Multiscale Training Technique (MST) is used in many places to solve the generalization problem [1, 2, and 3]. Results of MST depend largely on resolution of the character images. Image resolution and the training speed have to be optimized to achieve the highest percentage of accuracy.

Work has been done on identification of the characters in Devnagri script by combining multiple feature extraction techniques like intersection, shadow feature, chain code histogram and straight line fitting, [4].

Another approach towards feature extraction technique is to calculate only twelve directional feature inputs depending upon the gradients, where the features of the hand written characters are the directions of the pixels with respect to their neighboring pixels, [5].

Hybrid methods are also applied to recognize the hand written characters. One such method is a prototype learning/matching method that can be combined with support vector machines (SVM) in pattern recognition, [6].

K-nearest neighbor methods can be used to recognize the patterns, [7]. In K-nearest neighbor method the pattern is obtained by looking into k number of nearest patterns having the least Euclidean distance with that of the pattern. Artificial Neural Nets (perceptron learning) are used to train the nets and later using the nets identifying the characters, [8, 9, and 10]. But obtaining 100% accuracy is still a challenge for many such nets.

In this paper a concept of recognizing hand written character pattern has been developed and implemented called Row-wise segmentation technique. RST helps in minimizing errors in pattern recognition due to different handwriting styles to great extent.

In this method input pattern matrix is segmented row-wise into different groups. Target pattern is also grouped where each group is the numeric equivalent of the chronological position of each English alphabet. Each input segment is fully interconnected with each target group. Number of target groups is equal to the number of rows in the input matrix.

In general, the overall program has been divided into two parts, training and testing. Training requires the net to read segmented input patterns and testing requires the net to read any test character pattern, to read the produced target samples and to count the majority of samples and to find out the numeric equivalent of the sample to identify the character.

2. OVERVIEW OF THE PROPOSED HAND WRITTEN CHARACTER RECOGNITION PROGRAM AND METHODOLOGY
The standard characters are written on a piece of A4 size paper with uniform square sized boxes (Figure 1). Each character is written in one box. Blue/Black ballpoint pen is used to write the characters on the paper. 26 uppercase letters of English alphabets are taken for the training purpose. The characters are captured using a scanner.

Figure 1: Characters scanned from a scanner.

The captured images are used to generate input matrices. Each input matrix corresponding to an alphabet is segmented row-wise, each row forming an input pattern, which were fed to the perceptron neural network for the training. RST is used for setting the targets. Weights have been calculated using RST. Few English alphabets written by different people are taken for testing. Scilab has been used to test the results and the accuracy is measured.

3. PREPARATION OF THE INPUT PATTERNS

The captured or scanned images are in RGB scale, which has to be converted into grayscale format and later into binary for further processing. Each character is logically encaged in a square of size [80 x 80] pixels by using Scilab programming logic. Finally, 26 different matrices are produced of size [80 x 80] in the binary format. All the 0s of the vectors are replaced by –1s for better learning. Using 0s takes more epochs (iterations). Each matrix is segmented row-wise. 80 rows form eighty such patterns for each matrix.

4. PERCEPTRON LEARNING METHOD

The classical perceptron neural network has been used to perform the character recognition (Figure 2). This is not only very simple to implement but also more powerful learning rule in comparison to some of the other learning rules, like, sigmoid learning rule, hebbs learning rule etc. Under suitable assumptions, its iterative learning procedure can be proved to converge to the correct weights. The output of the perceptron is y. Output is calculated by the activation function f, which is a function of the net output, y_out. So, y=f(y_out)
And,

$$y\_out = \sum_i x_i w_i \qquad (1)$$

Where, x_i is the input vector and w_i is the weight vector. The function f(y_out) takes the following values depending upon the values of y_out.

$$f(y\_out) = \begin{cases} 1 \text{ if } y\_out > \theta \\ 0 \text{ if } -\theta <= y\_out <= \theta \\ -1 \text{ if } y\_out < -\theta \end{cases} \qquad (2)$$

Here, θ is the threshold value, taken at random. For each training input, the net would calculate the response of the output unit.

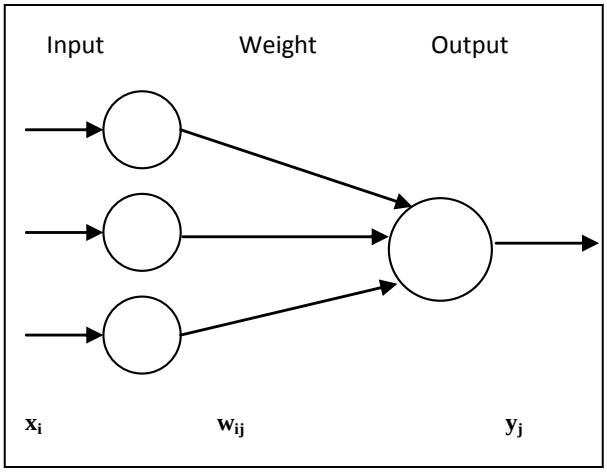

Figure 2: Simple perceptron to perform single classification.

The net would determine whether an error occurred for this pattern by comparing the calculated output with the target value. If an error occurs for a particular training input pattern, the weights would be changed according to the formula:

$$w_i(new) = w_i(old) + \alpha t_i x_i \qquad (3)$$

α is the learning rate, the value of which is taken at random, t_i is the target value, i.e. is the output expected from the net. The output y_i produced by the net is compared with t_i and the difference leads to the modification in weight given by equation (3). The process continues until y_i becomes equal to t_i.

5. ROW-WISE SEGMENTATION TECHNIQUE (RST)

The training of the net started with input vector of length 6400, (Figure 3).

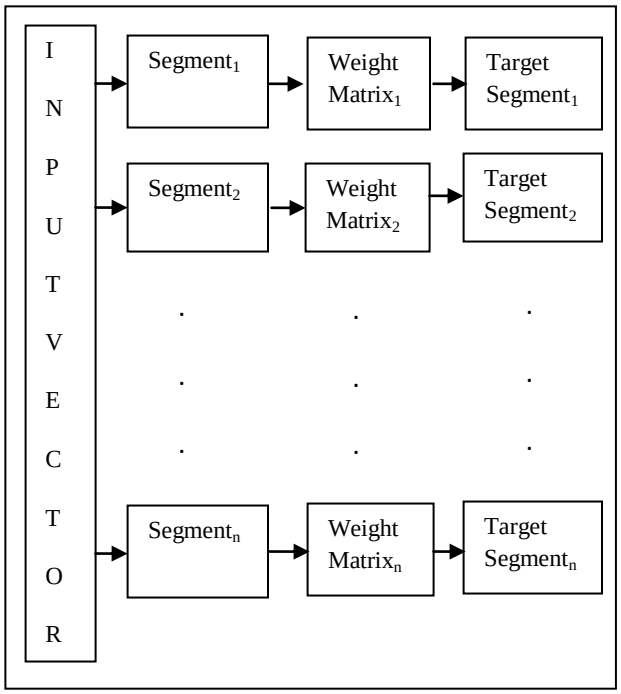

Figure 3: Conceptual Diagram of RST training.

The input vector is segmented into different groups. The length of each group is 80. The target pattern is also grouped into 80 identical patterns. Each input segment is fully interconnected with the corresponding target group. Each peer-to-peer input-target set is having separate weight matrices. The weight matrices are initially set to zero. After being trained for several epochs, the final weight that produces the correct output is obtained. The net is tested by providing sample characters written in different handwriting styles, (Figure 4). Different characters give different output vectors. The output vectors are observed to find out the total number of matching patterns for a character using a counter. Majority of similar patterns present in the output is considered for the identification of the character. For example, if the pattern present in majority in the output is [-1,-1,-1, 1,-1], replacing -1s by 0s gives [00010]. The decimal equivalent of [00010] is 2, which is the numeric equivalent of the chronological position of character B, if the alphabets are arranged alphabetically.

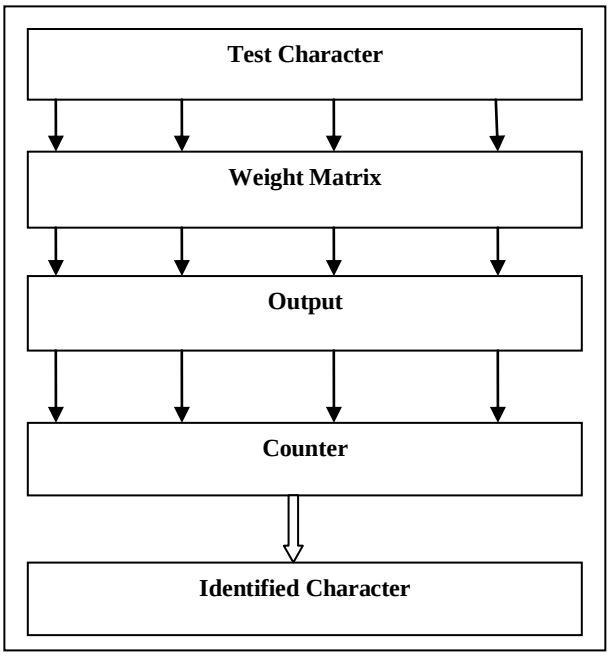

Figure 4: Conceptual diagram of RST testing.

6. RESULT ANALYSIS

The method has been tested in Scilab software. Few characters were taken initially, (Figure 5).

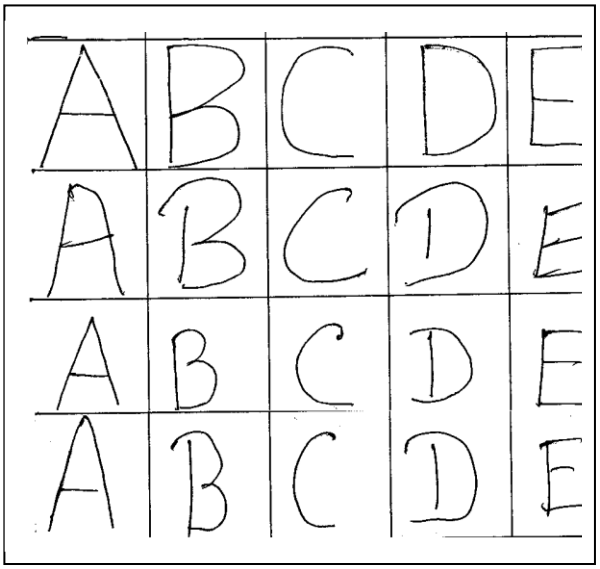

Figure 5: Characters taken for testing.

The accuracy of the method was measured by testing its capability of identifying the character samples taken from few people, written in different handwriting styles.

6.1 Neural Network parameters used in the experiment

Number of neurons in the input unit=6400

Number of neurons in each input pattern segment=80

Number of input pattern segments=80

Number of Neurons in the output unit = 400

Number of Neurons in each output segment = 5

Dimension of each weight matrix = 80 x 5

Training Algorithm used = perceptron

Number of Epoch = 3, $\theta = 0.2$, $\alpha = 1$

t_i = {A(-1,-1,-1,-1,1), B(-1,-1,-1,1,-1),

C(-1,-1,-1,1,1), D(-1,-1,1,-1,-1), E(-1,-1,1,1,-1)}

The accuracy of the system is shown in Table 1. The five characters A, B, C, D and E were taken initially for testing.

The response of the experiment shows very good results for the first five alphabets taken from the handwritten samples of our people.

7. DISCUSSION

The results show that RST training allows very fast convergence. The number of epochs required for the training is very less. The accuracy of identifying the characters is also very good.

However, the method is tested for some sample characters, which are totally deviated from their positions. The method failed to produce correct outputs for the same as the total deviations from the position produces completely different training pattern and hence a completely different target pattern. Experiments have to be carried out to increase the efficiency of RST and also provide more generalized method.

Table 1. Identification of characters taken from different people in different hand writing styles.

Alphabets	Number of samples	Correctly recognized	Percentage of accuracy
A	4	3	75%
B	4	2	50%
C	4	4	100%
D	4	3	75%
E	4	4	100%
Total Accuracy			80%

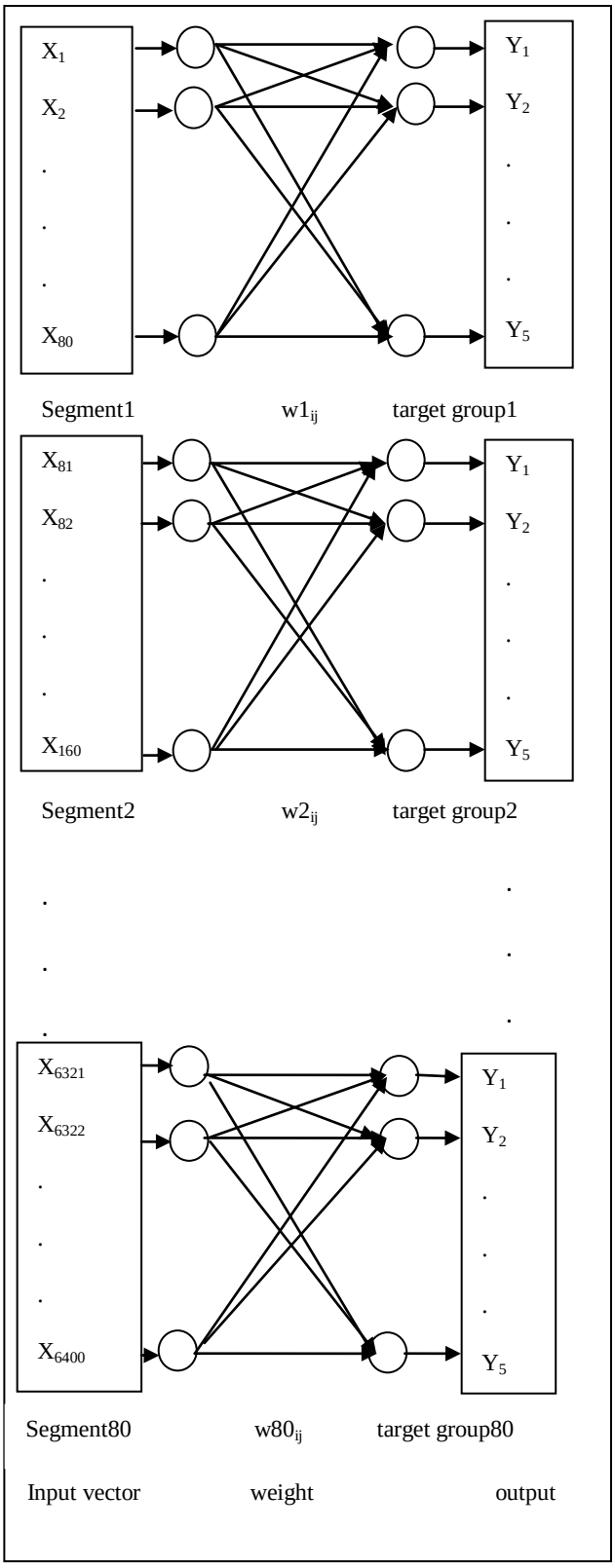

Figure 6: RST using perceptron.

8. AN ALGORITHIMIC APPROACH FOR RST

Step 1. Consider a single layer neural network with the input matrix of order [80 x 80] and output matrix of order [80 x 5].

Step 2. Segment the input matrix row wise into 80 rows of length 80 each and group the output matrix into 80 groups each of length 5. Each input segment is fully interconnected with each of the corresponding output group (Figure 6). The dimension of each weight matrix for an input output combination is [80 x 5]. There are 80 such weight matrices.

Step 3. Initialize all weights to 0 and set α to 1 for simplicity.

Step 4. Apply perceptron learning algorithm to each pair of input output combination.

Step 5. Repeat Step 6 to Step 10 for each test character.

Step 6. Present the test character to the trained net.

Step 7. Set the counter to 0 for each known pattern (each pattern representing an alphabet) for the test character.

Step 8. Check the output matrix and each group for the known pattern.

Step 9. For each known pattern found in the output, increment the counter by one.

Step 10. Consider, the known pattern showing the maximum value of counter as the winner. If there are no more test characters stop and exit

9. CONLUSION

RST gives very good accuracy, if the characters were written in boxed sheets. In this paper the method applied used the logic of encaging the characters without using the boxed sheets but the logic provides static encaging. Problems in identifying the characters arises when the characters gets fully deviated from their positions on the sheet. Efficient algorithm is still to be explored to encage the characters written on any position of the paper. Generalizations among variations in sizes of the characters in the box also produces problem.

10. ACKNOWLEDGMENTS

Our thanks go to all the colleagues and friends who have contributed their valuable time and efforts to help us to develop this paper in an organized way.

11. REFERENCES

[1] Robinson, G. (1995), The Multiscale Technique, Available: http://www.netlib.org/utk/Isi/pcwLSI/text/node123.html.

[2] Handwritten Character Recognition, Available: http://tcts.fpms.ac.be/rdf/hcrinuk.htm.

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

[4] Sandhya Arora, Debotosh Bhattacharjee, Mita Nasipuri, Dipak kumar Basu and Mahantapas Kundu, Combining Multiple Feature Extraction Techniques for Handwritten Devnagri Character Recognition, Available: http://arxiv.org/ftp/arxiv/papers/1005/1005.4032.pdf, (Accessed: 26, July, 2010).

[5] Dayashankar Singh, Sanjay Kr. Singh and Dr. (Mrs.) Maitreyee Dutta, Hand Written Character Recognition Using Twelve Directional Feature Input and Neural Network, Available: http://www.ijcaonline.org/journal/number3/pxc387173.pdf, (Accessed: 26, July, 2010).

[6] Fu Chang, Chin-Chin Lin and Chun-Jen Chen, Applying A Hybrid Method To Handwritten Character Recognition, www.iis.sinica.edu.tw/~fchang/paper/931_chang_F.pdf, (Accessed: 26, July, 2010).

[7] L van der Maaten-2009, A New Benchmark Data Set for Handwritten Character Recognition, Available: www.tilburguniversity.nl/faculties/humanities/ticc/.../TR2009002.pdf, (Accessed: 2010, July 19th).

[8] Brown, E.W. (1993), Applying Neural Networks to Character Recognition, Available: http://www.ccs.neu.edu/home/feneric/charrecnn.html (Accessed: 2010, July 19th).

[9] Neural Networks, G.N. Swamy, G. Vijay Kumar, SCITECH.

[10] Fundamentals of Neural Networks, Architectures, Algorithms and Applications, Laurene Fausett, Pearson Education