A NOVEL APPROACH FOR ALPHANUMERIC CHARACTER RECOGNITION USING BACK PROPAGATION ALGORITHM

A.Ratna Raju
Department of Computer Science and Engineering, MGIT, Hyderabad (India)

ABSTRACT

Classification methods based on learning from examples have been widely applied to character recognition from the 1990s and have brought forth significant improvements of recognition accuracies. A Neural network is a machine that is designed to model the way in which the brain performs a particular task or function of interest. The network is usually implemented by using electronic components or is simulated in software on a digital computer. Speed, accuracy are crucial to the practical use of handwriting recognition. Besides, extensibility is also an important concern as we move from one domain to another which requires the character set to be extended. In this paper we propose a modified approach for handwritten character recognition using back propagation algorithm. The results are quite encouraging in terms of percentage of characters being successfully recognized. One advantage of proposed scheme is that the system is quite tolerant to changing conditions and inputs. The system consistently learns. Moreover the recognition ratio is excellent in the proposed system.

Keywords: ANN, BPN Knowledge, Neuron, Neural Network, Supervised, Pattern Recognition

I. INTRODUCTION

Over the years, computerization has taken over large number of manual operations, one such example is offline handwritten character recognition, which is the ability of a computer system to receive and interpret handwriting input present in the form of scanned images. One of the most classical applications of the Artificial Neural Network is the Character Recognition System. This system is the base for many different types of applications in various fields, many of which we use in our daily lives. Cost effective and less time consuming, businesses, post offices, banks, security systems, and even the field of robotics employ this system as the base of their Operations. Handwritten character recognition is a difficult problem due to the great variations of writing styles, different size (length and height) and orientation angle of the characters. Handwritten Character recognition is an area of pattern recognition that has become the subject of research during the last some decades. Neural network is playing an important role in handwritten character recognition. Many reports of character recognition in English have been published but still high recognition accuracy and minimum training time of handwritten English characters using neural network is an open problem[7]. Therefore, it is a great important to develop an automatic handwritten character recognition system for English language. In this paper, efforts have been made to develop automatic handwritten character recognition system for English language with high recognition accuracy and minimum training and classification time. In the early stage of OCR (optical character recognition) development, template matching based recognition techniques were used [5]. Character recognition is the process to classify the input character according to the predefine character class. With increasing the interest of computer applications, modern society needs the input text into computer readable form. This research is a simple approach to implement that dream as the initial step to convert the input text into computer readable form. In this paper we use a modified approach for handwritten character recognition. This method, by its self organizing properties, is able to infer relationships and learn more as more inputs are presented. One of the most classical applications of the Artificial Neural Network is the Character Recognition System. This system is the base for many different types of applications in various fields, many of which we use in our daily lives. Cost effective and less time consuming, businesses, post offices, banks, security systems, and even the field of robotics employ this system as the base of their Operations. Handwritten character recognition is a difficult problem due to the great variations of writing styles, different size (length and height) and orientation angle of the characters. Handwritten Character recognition is an area of pattern recognition that has become the subject of research during the last some decades. Neural network is playing an important role in handwritten character recognition [3]. Many reports of character recognition in English have been published but still high recognition accuracy and minimum training time handwritten English characters using neural network is an open problem. Therefore, it is a great important to develop an automatic handwritten character recognition system for English language. In this paper, efforts have been made to develop automatic handwritten character recognition system with high recognition accuracy and minimum training and classification time. Characters consist of line segments and curves. Different spatial arrangements of these elements form different characters. In order to recognize a character, we should find out the structural relationships between the elements which make up the character. However, in practice, the syntactic and structural approach [2,3] suffers from several drawbacks. One of the major concerns is the need for robust extraction of primitives [4].

II. RELATED WORK

An Abductive Approach to Handwritten Character Recognition for Multiple Domains[3] describes an approach to automated hand-written character recognition that applies domain-specific knowledge such that a partial solution can be refined through top-down guidance. Specifically, the recognition task is one in which features derived from the input data are explained through higher-level hypotheses using abduction. Top-down guidance is used to improve accuracy. This approach has been applied to several domains. Individual character recognition is 70% without top-down guidance but improves to 99% with guidance. The layered abduction approach applied here is very different from other, highly successful hand-written recognition approaches that use neural networks or genetic algorithms.

Advantages of CHREC over other such systems are its ability to express uncertainty in its solutions when there is sufficient uncertainty, to apply a greater variety of knowledge types and sources, to use forms of less direct reasoning, and to apply noise hypotheses to account for pixels in the bitmap that should be ignored. CHREC continues to be expanded. A learning algorithm was implemented to learn new character recognizers. With this in place, CHREC learned lower case characters and performed reasonably well on the postal domain, around 96% character recognition accuracy. A fourth domain is being implemented, that of recognizing English sentences. A database of 500English words are being used along with a very restricted syntax. This domain will afford two layers of top-down guidance – words in the dictionary will correct individual character errors while legal syntaxes will correct incorrectly recognized words. It is too early to report on CHREC's performance in this domain, but it is hoped to equal that of the bank check domain. Results will be reported when they become available.

Handwritten Character Recognition Using Multiscale Neural Network Training Technique [2] Multiscale neural training with modifications in the input training vectors is adopted in this paper to acquire its advantage in training higher resolution character images. Selective thresholding using minimum distance technique is proposed to be used to increase the level of accuracy of character recognition. A simulator program (a GUI) is designed in such a way that the characters can be located on any spot on the blank paper in which the characters are written. The results show that such methods with moderate level of training epochs can produce accuracies of at least 85% and more for handwritten upper case English characters and numerals. When the resolution of the character images grows larger, neural network training tends to be slow due to more processing for larger input matrix. If the character images have lower resolution, the training process is much faster. However, some important details might be lost. Hence, it is a tradeoff between image resolution and training speed to recognize hand written characters. To optimize between these two parameters, it has been shown that one can adopt the multiscale training technique with modifications in input vectors as it provides faster training speed for different image resolutions.

III. FEED FORWARD BACK PROPAGATION NEURAL NETWORK

A feed-forward network is a neural network where synapses between the neurons do not form a directed cycle. A typical feed-forward network consists of an array of layers containing an input layer, followed by at least one hidden layer and an output layer. A layer typically consists of hundreds of neurons, no two of which are connected to each other. Directed synapses connect every neuron in one layer to all neurons in the next layer. Each synapse has a **weight** associated with it. Whenever data is propagated from its source neuron to a target neuron, the synapse multiplies the data by its weight. The weights of synapses in a network collectively play a major role in functioning of the network. Neural network training mainly refers to the suitable adjustment of the synapse weights. Back-propagation algorithm is a commonly used supervised algorithm to train feed-forward networks. Structure of a neuron in a back-propagation network is shown below.

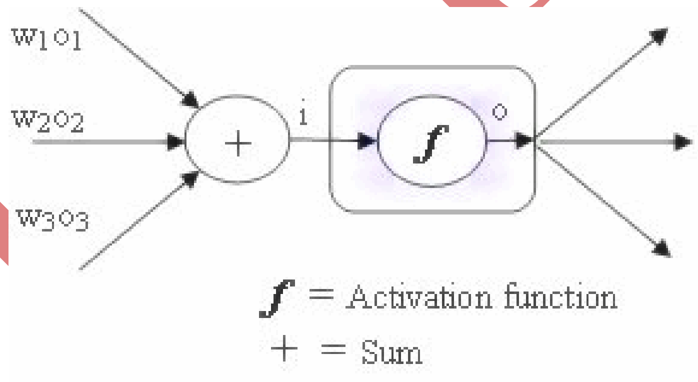

Figure 1. Structure of a neuron in a back-propagation network is shown below.

Each neuron has an input, output and an error value associated with it. In case of neurons in the input layer, the input value is set by the external environment. All other neurons will have their inputs evaluated as the sum of data transferred to them by each source synapse. Output at a neuron is obtained by applying a differentiable function on the input. This function is referred to as Activation Function. The output value so obtained is propagated forwards via associated target synapses. Finally, the set of output values of neurons in output layer is presented as the network output.

Neuron Input = Sigma (source synapse weight * source neuron output)

Neuron Output = Activation Function (Neuron Input)

Error at a neuron is a measure of difference between desired output and the actual obtained output. It is evaluated at the output layer, and is back-propagated to the inner layers. Hence the name 'back-propagation algorithm'. The algorithm trains the network such that the mean squared error is minimized.

3.1 BACK PROPAGATION ALGORITHM OVERVIEW

Training an artificial neural network is nothing but providing it with a training set and allowing it to learn by adjusting weights of its synapses. A training set is a collection of training samples.

Training Set = Set of training samples

A training sample is a pair of a sample input vector and a desired output vector. In case of unsupervised training, the output vector should be null. The length of input vector should be same as the number of neurons in the input layer, and the output vector length should be equal to the number of neurons in output layer.

Training Sample = (input vector, desired vector)

Back-propagation algorithm is a commonly used supervised algorithm to train feed-forward networks. It was first introduced by Paul Werbos in his book 'The Roots of Back-propagation'. The basic idea is to determine how the neural network behaves for a sample input, compare how different it is from the desired behavior and then adjusting the weights of synapses to minimize the difference. This process is repeated for all training samples in the set multiple times to ensure proper training.

3.2. Back propagation training algorithm

1. Initialize the weights of networks
2. Choose a random training sample, and assign input vector to the input neurons
3. Propagate all neurons in the forward direction to obtain output at the output layer
4. Evaluate error values at each neuron in the output layer as the difference between obtained output and the desired output of the training sample chosen.
5. Evaluate Mean Squared Error value. This value reflects the effectiveness of training done so far.
6. Back propagate the errors, all the way up to the input layer
7. Calculate delta (weight update) for all synapses
8. Update the weights all synapses such that the sum-squared value of error is minimized.
9. Now, choose another random sample and repeat the process. In this fashion, train all samples in some random order. A training epoch is a cycle through all the samples in the training set. Typically, many training epochs are required to train a back-propagation network.
10. Check if the stopping criterion has reached. If not, continue with the next training epoch.

    The stopping criterion is usually a limit to acceptable mean squared error or a limit to the number of training cycles to use.

    Learning Rate

    Learning rate is one of the parameters which govern how fast a neural network learns and how effective the training is. The effect of learning rate on training is explained below. Consider a neural network which is undergoing the process of learning. Let us assume that the weight of some synapse in the partially trained network is 0.3. When the network is introduced with a new training sample, the training algorithm demands the synapse to change its weight to 0.7 (say) so that it can learn the new sample appropriately. If we update the weight straightaway, the neural network will definitely learn the new sample, but it tends to forget all the samples it had learnt previously. This is because the current weight (0.3) is a result of all the learning that it has undergone so far. So we do not directly change the weight to 0.7. Instead, we increase it by a fraction (say 25%) of the required change. So, the weight of the synapse gets changed to 0.4 and we move on to the next training sample. This factor (0.25 in this case) is called Learning Rate. Proceeding this way, all the training samples are trained in some random order. When this training cycle is repeated a large number of times, the neural network ultimately learns all samples effectively.

Operating Environment

Software Requirements used are Microsoft .Net framework 2.0, Microsoft C# .Net language, Microsoft Visual Studio 2005 IDE, Microsoft Windows 2000 SP4 or higher.

Hardware Requirements used are P4 processor, 512MB of main memory (RAM) and 40GB Hard disk.

Design and Implementation Constraints

The software is designed in such a way that the user can easily interact with the screen because they are GUI components and he just have to enter the character then the software itself does the preprocessing and pattern matching to produce the output. Software is designed in such a way that with little modifications it can be extended to the real time.

Assumptions and Dependencies

User has to train the neural network before expecting the output. Training is done through the GUI form.

Inputs

The input to our project is a single alphabet. We give the input by drawing the alphabet with the mouse on a given screen.

Outputs

The output of our project is a character after recognized as a specific alphabet. It gives the output based on the matching probability of the given character with the characters or alphabets already trained with.

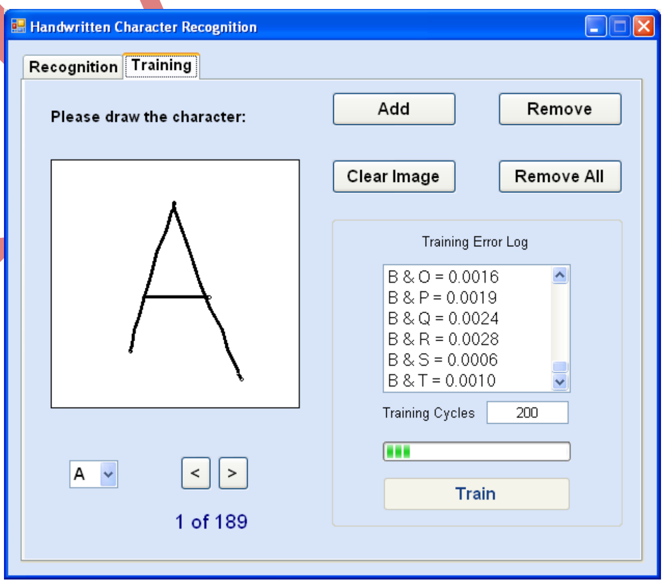

Figure 2. The screen illustrates training process for a character

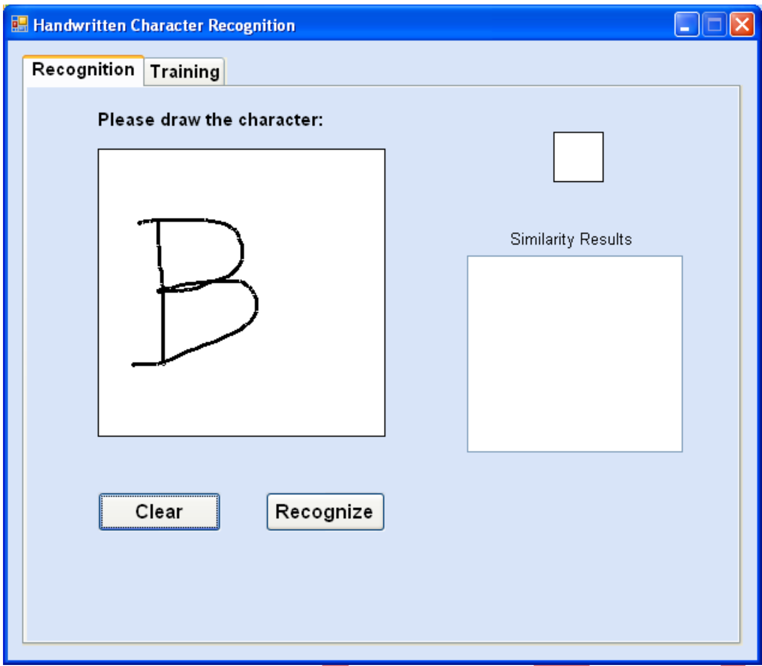

Figure 3. The screen illustrates character given for recognition

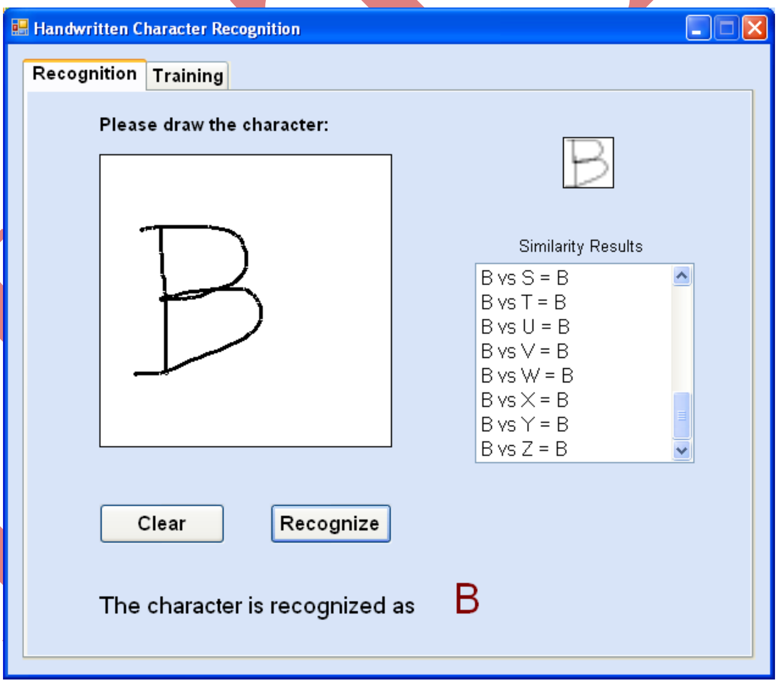

Figure 4.. The screen illustrates output generated after recognition

IV. CONCLUSION

Handwriting recognition is not a new technology, but it has gained public attention until recently. The ultimate goal of designing a handwriting recognition system an accuracy rate of 100 % is quite illusionary, because even human beings are not able to recognize every hand written text without any doubt. For example, most people can not even read their own notes. Therefore there is an obligation for a writer to write clearly. The disadvantage of back-propagation learning is that it is a time consuming process. Variations of this algorithm have been proposed with some more parameters (like momentum and jitter) to improve the learning speed. Another way of attempting to improve the learning speed is by using adaptive learning rates where the rate is appropriately adjusted after every training cycle to suit the conditions. Increasing the learning rate with the decrease in network error often helps to speed up the learning process. Various adjustable learning rate algorithms have been proposed and studied. Enhancements to the project can be either software enhancements or hardware enhancements or both the hardware enhancements include replacement of existing hardware components with the latest and advanced components. The software enhancements include improvements in the training algorithm employed to train the neural network, improvements in the GUI developed, putting up additional controls, and so on. The software developed in this work can be extended to recognition of lower case characters, words, phrases and so on by increasing the adaptability of the neural network. It can also be extended for different languages.

REFERENCES

[1].Shashank Arokar : Visual Character Recognition using Artificial Neural Networks, MGM's college of Engineering and Technology, University of Mumbai, India.

[2]. Velappa Ganapathy, and Kok Leong Liew : Handwritten Character Recognition Using Multiscale Neural Network Training Technique, Proceedings of World Academy of Science, Engineering and Technology, Volume 29 May 2008 ISSN 1307-6884.

[3].Andrew T. Wilson: Off-line handwriting recognition using Artificial Neural Networks, University of Minnesota, Morris

[4]. Pelin Gorgel and Oguzhan OZTAS : Handwritten Character Recognition using Artificial neural Networks,Computer Engineering Department, Engineering Faculty, Istanbul University, Avcilar, Istanbul, Turkey, 2007.

[5].Alexander J. Faabor : Using Neural Networks to create an Adaptive Character Recognition System, Cornell University, Ithaca NY, 2002.

[6]. Kenji Nakayama, Osamu Hasegawa and Carlos Hernandez : Handwritten Alphabet and Digit Character Recognition using Skeleton Pattern Mapping with Structural Constraints, Dept. of Electrical and Computer Eng., Kanazawz univ. Japan.

[7]. Muhammad Faisal Zafar, Dzulkifli Mohamad and Razib M. Othman : Writer Independent OnlineHandwritten Character Recognition Using a Simple Approach, Faculty of Computer Science and Information Systems, UTM, 81310.