Influence of Introducing an Additional Hidden Layer on the Character Recognition Capability of a BP Neural Network having One Hidden Layer

Amit Choudhary [1], Rahul Rishi [2], Vijaypal Singh Dhaka [3], Savita Ahlawat [4]

[1] Deptt. of Comp. Sc., Maharaja Surajmal Institute,
New Delhi, India.

[2] Dept .of Comp. Sc. and Engg., TITS, Bhiwani,
Haryana, India.

[3] Dept .of Comp. Sc., IMS, Noida,
UP, India.

[4] Deptt. of Comp. Sc.and Engg., MSIT,
New Delhi, India.

***Abstract*— Objective of this paper is to study the character recognition capability of feed-forward back-propagation algorithm using more than one hidden layer. This analysis was conducted on 182 different letters from English alphabet. After binarization, these characters were clubbed together to form training patterns for the neural network. Network was trained to learn its behavior by adjusting the connection strengths on every iteration. The conjugate gradient descent of each presented training pattern was calculated to identify the minima on the error surface for each training pattern. Experiments were performed by using one and two hidden layers and the results revealed that as the number of hidden layers is increased, a lower final mean square error is achieved in large number of epochs and the performance of the neural network was observed to be more accurate.**

Keywords: Character Recognition, MLP, Hidden Layers, Back-propagation, Conjugate Gradient Descent

I. INTRODUCTION

Character recognition is the ability of a computer to receive and interpret handwritten input from sources such as paper documents, photographs, touch-panels, light pen and other devices. This technology is steadily growing toward its maturity. The domain of hand written text recognition has two completely different problems of On-line and Off-line character recognition.

On-line character recognition [1] involves the automatic conversion of characters as it is written on a special digitizer or PDA, where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching. That kind of data is known as digital ink and can be regarded as a dynamic representation of handwritten characters. The obtained signal is converted into letter codes which are usable within computer and text-processing applications.

On the contrary off-line character recognition involves the automatic conversion of character (as an image) into letter codes which are usable within computer and text-processing applications. The data obtained by this form is regarded as a static representation of handwritten character. The technology is successfully used by businesses which process lots of handwritten documents, like insurance companies. The quality of recognition can be substantially increased by structuring the document (by using forms).

The off-line character recognition is comparatively difficult, as different people have different handwriting styles and also the characters are extracted from documents of different intensity and background [2]. Nevertheless, limiting the range of input can allow recognition process to improve. For example, the ZIP code digits are generally read by computer to sort the incoming mail.

One of the most important types of feed forward neural network is the Back Propagation Neural Network (BPNN). Back Propagation is a systematic method for training multi-layer artificial neural network [3]. It is a multi-layer feed forward network using extend gradient-descent based delta-learning rule, commonly known as back propagation (of errors) rule. Back Propagation provides a computationally efficient method for changing the weights in a feed forward network, with differentiable activation function units, to learn a training set of input-output examples. Being a Gradient Descent Method it minimizes the total squared error of the output computed by the net. The network is trained by supervised learning method. The aim is to train the net to achieve a balance between the ability to respond correctly to the input characters that are used for training and the ability to provide good responses to the input that are similar. The total squared error of the output computed by net is minimized by a gradient descent method known as Back Propagation or Generalized Delta Rule.

The experiments conducted in this paper have shown the effect of an additional hidden layer on the learning and off-line character recognition accuracy of the neural network. Two experiments were performed. Experiment-1 employed a network having single hidden layer and Experiment-2 employed a network having two hidden layers. All other experimental conditions such Learning Rate, Momentum Constant (α), Activation Function, Maximum Training Epochs, Acceptable Error Level and Termination Condition were kept same for all the experiments.

The remainder of the paper is organized as follows: Section II deals with the overall system design and the various steps involved in the OCR System. Neural Network Architecture and functioning of proposed experiments are presented in detail in section III. Section IV provides various experimental conditions for all the experiments conducted under this work. Discussion of Results and interpretations are described in section V. Section VI presents the conclusion and also gives the future path for continual work in this field.

II. Overall System Design

A typical character recognition system is characterized by a number of steps, which include (1) Digitization / Image Acquisition, (2) Preprocessing, (3) Binarization (4) Feature Extraction, and (5) Recognition / Classification. Fig. 1 illustrates one such system for handwritten character recognition.

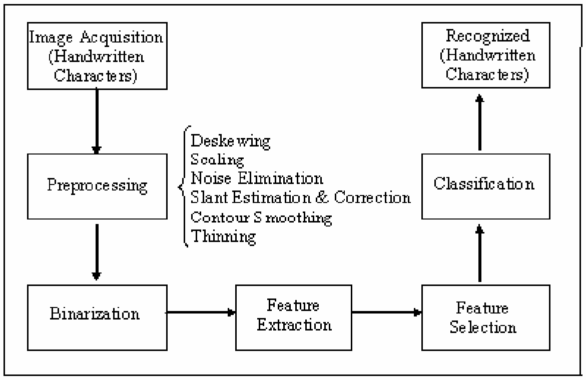

Figure 1. Typical Off-Line Character Recognition System

The steps required for typical off-line character recognition are described here in detail:

A. Preprocessing

Preprocessing aims at eliminating the variability that is inherent in cursive and hand-printed characters. The preprocessing techniques that have been employed in an attempt to increase the performance of the recognition process are as follows:

Deskewing is the process of first detecting whether the handwritten word has been written on a slope and then rotating the word if the slope's angle is too high so the baseline of the word is horizontal. Some examples of techniques for correcting slope are described by Brown and Ganapathy [4].

Scaling sometimes may be necessary to produce words of relative size

Noise (small dots or blobs) may be introduced easily into an image during image acquisition. Noise elimination in character images is important for further processing; therefore, these small foreground components are usually removed. Madhvanath, Kleinberg, and Govindaraju [9] also analyzed the size and shape of connected components in a word image and compared them to a threshold to remove the noise.

Slant estimation and correction is an integral part of any word image preprocessing. The slope can be estimated through analysis of the slanted vertical projections at various angles [5].

Contour Smoothing is a technique to remove contour noise which is introduced in the form of bumps and holes due to the process of slant correction.

Thinning is a process in which the skeleton of the word image is used to normalize the stroke width.

B. Binarization

All hand printed characters are scanned into grey scale images. Each character image is traced vertically after converting the gray scale image into binary matrix [6]. The threshold parameter along with the grayscale image is made an input to the binarization program designed in MATLAB. The output is a binary matrix which represents the image shown in Fig. 2(c).

Every character is first converted into a binary matrix and then resized to 8 X 6 matrixes as shown in Fig. 2(c) and reshaped to a binary matrix of size 48 X 1 which is made as an input to the neural network for learning and testing. Binary matrix representation of character 'A' can be defined as in Fig. 2(c). The Resized characters were clubbed together in a matrix of size 48 X 26 to form a sample [6]. In the sample, each column corresponds to an English alphabet which was resized into 48 X 1 input vector.

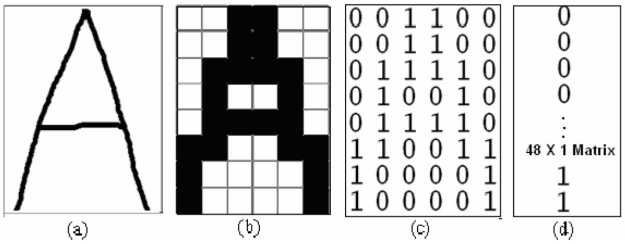

Figure 2. (a) Grey scale image of Character 'A' (b) Binary Representation of Character 'A'; (c) Binary Matrix representation and (d) Reshaped sample of Character 'A'.

For sample creation, 182 characters were gathered form 35 people. After pre-processing, 5 samples were considered for training such that each sample was consisting of 26 characters (A-Z) and 2 samples were considered for testing the recognition accuracy of the network.

C. Feature Extraction and Selection

Feature extraction is a process of studying and deriving useful information from the filtered input patterns. The derived information can be general features, which are evaluated to ease further processing. The selection of features is very important because there might be only one or two values, which are significant to recognize a particular segmented character.

D. Classification

Classification is the final stage of the character/numeral recognition. This is the stage where an automated system declares that the inputted character belongs to a particular category. The classification techniques here we have used is a feed forward back propagation neural network.

III. Neural Network Architecture Used In The Recognition Process

To accomplish the task of character classification and mapping, the multi layer feed forward artificial neural network is considered with nonlinear differentiable function 'Tansig' in all processing units of output and hidden layers. The processing units in the input layer, corresponds to the dimensionality of the input pattern, are linear. The number of output units corresponds to the number of distinct classes in the pattern classification. A method has been developed, so that network can be trained to capture the mapping implicitly in the set of input output pattern pair collected during an experiment and simultaneously expected to modal the unknown system for function from which the predictions can be made for the new or untrained set of data [3]. The possible output pattern class would be approximately an interpolated version of the output pattern class corresponding to the input learning pattern close to the given test input pattern. This method involves the back propagation-learning rule based on the principle of gradient descent along the error surface in the negative direction.

The network has 48 input neurons that are equivalent to the input character's size as we have resized every character into a binary matrix of size 8 X 6. The number of neurons in the output layer is 26 because there are 26 English alphabets. The number of hidden neurons is directly proportional to the system resources. The bigger the number more the resources are required. The number of neurons in a hidden layer was kept 10 for optimal results.

The neural network was exposed to 5 different samples as achieved in Section II. Actual output of the network was obtained by "COMPET" function [6]. This is a competitive transfer function which puts '1' at the output neuron in which the maximum trust is shown and rest neuron's result into '0' status. The output is a binary matrix of size 26×26 because each character has 26×1 output vector. First 26×1 column stores the first character's recognition output, the following column will be for next character and so on for 26 characters. For each character the 26×1 vector will contain value '1' at only one place. For example character 'A' if correctly recognized, will result in [1, 0, 0, 0 …all …0] and character 'B' will result in [0, 1, 0, 0 … all …0].

The difference between the desired and actual output is calculated for each cycle and the weights are adjusted during backpropagation. This process continues till the network converges to the allowable or acceptable error.

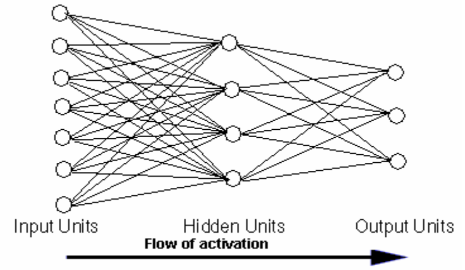

Figure 3. Feed Forward Neural Network with one Hidden Layer.

In the feed forward phase of operation, the signals are sent in forward direction and in back propagation phase of learning, the signals are sent in the reverse direction [3].

The training algorithm of back propagation involves four stages:

a). Initialization of weights: During this stage some random values are assigned for initialization of weights.

b). Feed Forward: During Feed Forward stage, each input unit receives an input signal and transmits this signal to each of the hidden units. Each hidden unit then calculates the activation function and sends its signal to each output unit. The output unit calculates the activation function to form the response of the net for the given input pattern.

c). Back Propagation of Errors: During back propagation of errors, each output unit compares its computed activation value (output) with its target value to determine the associated error for that input pattern with that unit. Based on the error, the error factor for each unit is computed and is used to distribute the error at each output unit back to all units in the previous layer. Similarly the error factor is computed for each hidden unit.

d). Updation of the Weights and Biases: During final stage, the weights and biases are updated for the neurons at the previous levels to lower the local error.

IV. Experimental Conditions

The various parameters and their respective values used in the learning process of all the three experiments with one and two hidden layers are shown in Table II.

TABLE I. EXPERIMENTAL CONDITIONS OF THE NEURAL NETWORK

PARAMETERS	VALUE	
Input Layer		
No. of Input neurons	48	
Transfer / Activation Function	Linear	
Hidden Layer		
No. of neurons	10	
Transfer / Activation Function	TanSig	
Learning Rule	Momentum	
Output Layer		
No. of Output neurons	26	
Transfer / Activation Function	TanSig	
Learning Rule	Momentum	
Learning Constant	0.01	
Acceptable Error (MSE)	0.001	
Momentum Term (α)	0.90	
Maximum Epochs	1000	
Termination Conditions (N_{HL})	Based on minimum Mean Square Error or maximum number of epochs allowed	
Initial Weights and biased term values	Randomly generated values between 0 and 1	
Number of Hidden Layers	Experiment-1	N_{HL} =1
	Experiment-2	N_{HL} =2

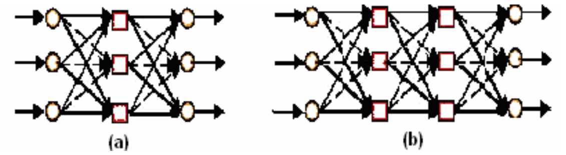

Figure 4. MLP having (a) One Hidden Layer(48-10-26) (b) Two Hidden Layers (48-10-10-26)

V. Discussion Of Results And Interpretations

The system is simulated using a feed forward neural network system that consists of 48 neurons in input layer, 10 neurons in hidden layer and 26 output neurons. The characters are resized into 8×6 binary matrixes and are exposed to 48 input neurons. The 26 output neurons correspond to 26 upper case letters of English alphabet. The network having one hidden layer was used for Experiment-1 and in Experiment-2, the process is repeated for the network having two hidden layers, each having 10 neurons as shown in Fig. 4(b).

In structured sections, the experiments and their outcomes at each stage are described.

A. Gradient Computation

The gradient descent is the characteristic of error surface. If the surface is not smooth, the gradient calculated will be a large number and this will give a poor indication of the "true error correction path". On the other hand, if the surface is relatively smooth, the gradient value will be a smaller one. Hence the smaller gradient is always the desirable one. This rationale is based on the knowledge of the shape of the error surface.

For each trial of learning, the computed values of gradient descent are shown in Table II.

TABLE II. Comparison of gradient values of the network for both experiments.

	Experiment-1(N_{HL}=1)	Experiment-2 (N_{HL}=2)
Sample No.	Gradient1	Gradient2
Sample1	1981400	1419834
Sample2	5792000	3714695
Sample3	7018400	5834838
Sample4	7173900	6157572
Sample5	6226395	6317917

It has been observed in Table II that in Experiment 2 using MLP with two hidden layers, the gradient values are much smaller than in MLP with one hidden layer used in Experiment 1.

B. Number of Epochs

The results of the learning process of the network in terms of the number of training iterations, depicted as Epoch, are represented in Table III.

TABLE III. Comparison Of Network Training Epochs Between The Two Learning Trails For Both The Experiments

	Experiment-1(N_{HL}=1)	Experiment-2 (N_{HL}=2)
Sample	Epoch1	Epoch2
Sample1	186	521
Sample2	347	623
Sample3	551	717
Sample4	695	832
Sample5	811	960

In the above table, Epoch1 and Epoch2 represent the number of network iterations for a particular sample when presented to the neural network having one hidden layer and two hidden layers respectively.

In Table III, it is clear that small number of epochs are sufficient to train a network when we use one hidden layer. As the number of hidden layers is made two, the number of epochs required to train the network also increases as observed in Experiment 2 of Table III. We can say that the network converges slowly when a two hidden layers are used in the experiment.

Although, the network with two hidden layers requires more time during learning, the gradient values are found to be quiet low as shown earlier in Table II. Hence, the error surface will be smooth and the network's probability of getting struck in the local minima will be low.

C. Error estimation

The network performance achieved is shown in Table IV. For both experiments with one and two hidden layers, it is evident that the error is reduced when two hidden layers are used in the network. In other words, we can say that with the increase in the number of hidden layers, there is an increase in probability of converging the network before the number of training epochs reaches it maximum allowed count.

TABLE IV. Error level attained by the neural network trained with both methods

	Experiment-1(N_{HL}=1)	Experiment-2(N_{HL}=2)
Sample	Error1	Error 2
Sample1	0.00006534	0.000023139
Sample2	0.00056838	0.00037402
Sample3	0.00083115	0.00055085
Sample4	0.00091238	0.00083480
Sample5	0.00487574	0.00121815

D. Testing

The character recognition accuracy of both networks with one and two hidden layers is shown in Table V. The networks were tested with 2 samples. These samples were new for both the networks because they were never trained with these samples. The recognition rates for these samples are shown in Table V.

TABLE V. Character Recognition Accuracy

Sample (Number of characters in test sample)	Experiment-1 (N_{HL}=1)		Experiment-2 (N_{HL}=2)	
	Correctly Recognised	Accuracy (%)	Correctly Recognised	Accuracy (%)
Sample-6 (26)	17	65.38	23	88.46
Sample-7 (26)	20	80	22	84.61

It has been observed in Table V that in Experiment 2 employing MLP with two hidden layers, the recognition rates are better than MLP with one hidden layer used in Experiment-1.

When both the networks having one hidden layer and two hidden layers are being trained with Sample 1, the profiles of

MSE plot for the training epochs are drawn in Fig. 5 and Fig. 6 respectively.

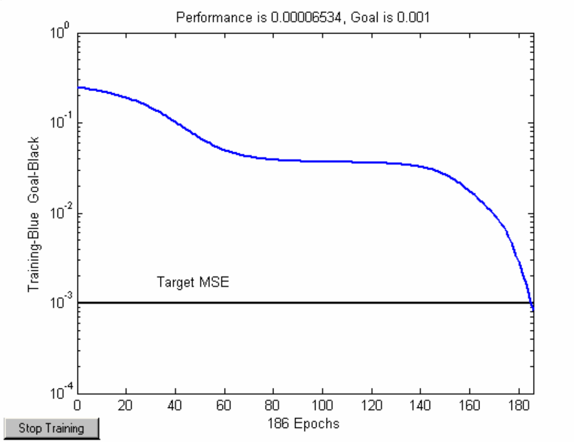

Figure 5. MSE Plot for the Network with One Hidden Layer

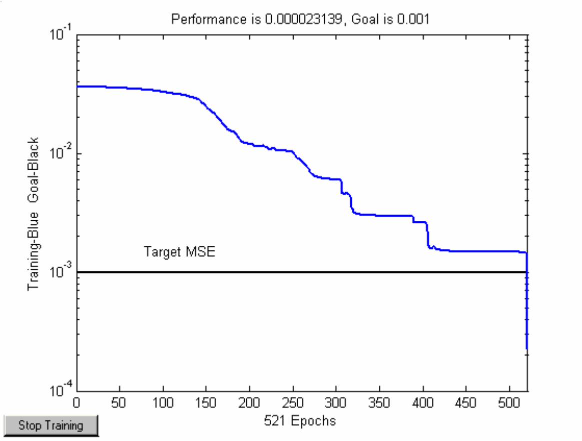

Figure 6. MSE Plot for the Network with Two Hidden Layers

The increase in the number of hidden layers results in the computational complexity of the network. As a result, the time taken for convergence and to minimize the error may be very high. After strong analysis, three relationships between the number of hidden layers, number of epochs and MSE are established.

$$N_{HL}\ \alpha\ N_{E} \qquad (5.1)$$

where N_{HL} is the number of hidden layers and N_E is the number of epochs.

The number of training epochs is inversely proportional to the minimum MSE.

$$N_{E}\ \alpha\ \frac{1}{MSE} \qquad (5.2)$$

The number of hidden layers is inversely proportional to the minimum MSE.

$$N_{HL}\ \alpha\ \frac{1}{MSE} \qquad (5.3)$$

where *MSE* is the mean square error.

VI. Conclusion and Future Scope

The proposed method for the handwritten character recognition using the descent gradient approach, showed the remarkable enhancement in the performance when two hidden layers were used. As shown in Table-II, the results of both the experiments for the different character sample represent that the gradient values were found to be least when two hidden layers were used in the network. Smaller the gradient values, smoother will be the error surface and the probability that the neural network will get struck in the local minima will be the least. Smaller gradient values indicate that the error correction is downy and accurate. It is clear from Table-V that the recognition accuracy is best in Experiment-2 where MLP with two hidden layers was used.

Eq.5.1 implies that the number of hidden layers is proportional to the number of epochs. This means that as the number of hidden layers is increased, the training process of the network slows down because of the increase in the number of epochs. However, Eq.5.3 implies that the training of the network is more accurate if more hidden layers are used. This accuracy is achieved at the cost of network training time as indicated by Eq.5.2.

If the accuracy of the results is a critical factor for an character recognition application, then the network having many hidden layers should be used but if training time is a critical factor then the network having single hidden layer (with sufficient number of hidden units) should be used.

Nevertheless, more work needs to be done especially on the test for more complex handwritten characters. The proposed work can be carried out to recognize English words of different character lengths after proper segmentation of the words into isolated character images.

References

[1] A. Bharath and S. Madhvanath," FreePad: a novel handwriting-based text input for pen and touch interfaces", Proceedings of the 13th international Conference on Intelligent User Interfaces, pp. 297-300, 2008.

[2] Bhardwaj, F. Farooq, H. Cao and V. Govindaraju," Topic based language models for OCR correction", Proceedings of the Second Workshop on Analytics For Noisy Unstructured Text Data, pp. 107-112, 2008.

[3] S. N. Sivanandam, S. N. Deepa," Principals of Soft Computing", Wiley-India, New Delhi, India. pp. 71-83, 2008.

[4] M. K. Brown and S. Ganapathy," Preprocessing techniques for cursive script word recognition", Pattern Recognition, pp. 447–458, 1983.

[5] D. Guillevic and C. Y. Suen," Cursive script recognition: A sentence level recognition scheme", Proceedings of the 4th International Workshop on the Frontiers of Handwriting Recognition, pp. 216–223, 1994.

[6] A. Choudhary, R. Rishi, S. Ahlawat, V. S. Dhaka, "Optimal feed forward MLParchitecture for off-line cursive numeral recognition," International Journal on Computer Science and Engineering, vol. 2, no.1s, pp. 1-7, 2010.