International Journal of Information and Computation Technology.
ISSN 0974-2239 Volume 3, Number 3 (2013), pp. 119-124

http://www. irphouse.com /ijict.htm

Evaluation of Neural Network Accuracy for Handwritten Character Recognition

Amit Choudhary[1] and Nikita Walia[2]

*[1]Computer Science Department, Maharaja Surajmal Institute,
C-4, Janakpuri, New Delhi, INDIA.*
*[2], Computer Science Department, Maharaja Surajmal Institute,
C-4, Janakpuri, New Delhi, INDIA.*

Abstract

The objective of this paper is to evaluate the character recognition capability of feed-forward back-propagation neural network using more than one hidden layer. This test has been conducted on 182 different upper case letters from English alphabet. After binarization, these characters have been clubbed together to form training patterns for the neural network. Network is trained to learn its behavior by adjusting the connection strengths on each iteration. The gradient descent of each presented training pattern is calculated to identify the minima on the error surface for each training pattern.

Keywords: BPANN; Hidden Layer; Character Recognition; OCR.

1. Introduction

The experiments conducted in this work have shown the effect of an additional hidden layer on the learning and off-line character recognition accuracy of the feed-forward back-propagation neural network. Two experiments have been performed. Experiment-1 employed a network having single hidden layer and Experiment-2 employed a network having two hidden layers. All other conditions such Learning Rate (η), Momentum Constant (μ), Activation Function and Termination Condition such as maximum training epochs allowed, acceptable error level etc. are kept same for both the experiments in this work. 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 is observed to be more accurate, but at the cost of training time.

2. Character Image Acquisition and Sample Preparation

All handwritten capital English characters are scanned into grey scale images [1-3]. These character images are first converted into binary images. One such image of character ‘A’ is shown in Fig. 1(a). These binary images are then resized to 8 × 6. The resized image of character ‘A’ is shown in Fig. 1(b).

Each character image is traced vertically column wise. The threshold parameter along with the grayscale image is made an input to the binarization program designed in MATLAB [4,5]. The output is a binary matrix which represents the image shown in Fig. 1(c). These images are then reshaped to a binary matrix of size 48×1 which is made as an input to the neural network for learning and testing as shown in Fig. 1(d). The resized characters have been clubbed together in a matrix of size 48×26 to form a sample. In a sample, each column corresponds to an English alphabet which is resized into 48×1 column vector [6].

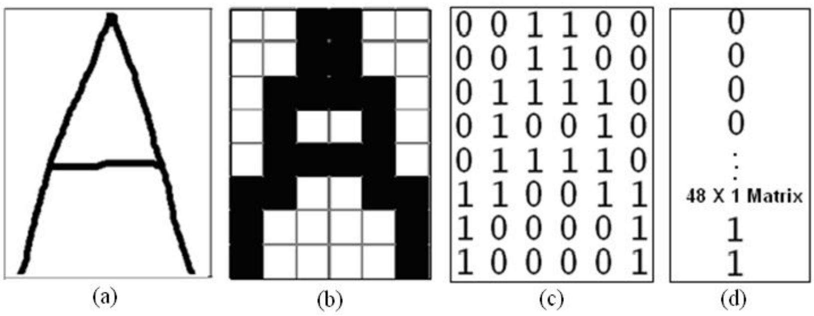

Fig. 1: (a) Binary Image of Character ‘A’ (b) Resized Binary Image of Character ‘A’; (c) Binary Matrix representation and (d) Reshaped Sample of Character ‘A’.

For sample creation, 182 (26×7=182) characters have been gathered form 7 people. After preprocessing, 5 samples are considered for training such that each sample is consisting of 26 characters (A-Z) and 2 samples are considered for testing the recognition accuracy of the network.

3. Implementation and Discusson of Results

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 is used for Experiment-1 and in Experiment-2; the process is repeated for the network having two hidden layers where each layer is having 10 neurons.

3.1 Number of Epochs

The results of the learning process of the network in terms of the number of training iterations, depicted as epochs are represented in Table 1.

Table 1: A Comparison of Training Epochs of the Network for both Experiments.

	Experiment-1(N_{HL}=1)	Experiment-2 (N_{HL}=2)
Sample No.	Epoch1	Epoch2
Sample1	186	521
Sample2	347	623
Sample3	551	717
Sample4	695	832
Sample5	811	960

In Table 1, 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.

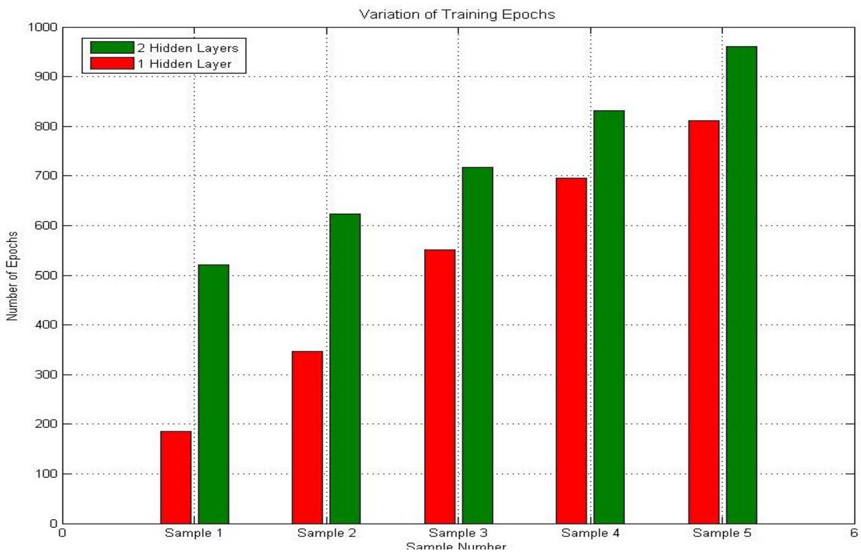

Fig. 2: Comparison of Number of Epochs for Network with One and Two Hidden Layers.

In Fig. 2, it is clear that small number of epochs are sufficient to train a network when one hidden layer is used. 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 1. The network converges slowly when two hidden layers are used in the experiment.

3.2 Error Estimation

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, 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 its maximum allowed count. The network performance achieved is shown in Table 2.

Table 2: The Error Levels Attained by the Network Trained in Both Experiments.

	Experiment-1(N_{HL}=1)	Experiment-2(N_{HL}=2)
Sample No.	Error1	Error 2
Sample1	0.00016534	0.000123139
Sample2	0.00056838	0.00037402
Sample3	0.00083115	0.00055085
Sample4	0.00091238	0.00083480
Sample5	0.00187574	0.00121815

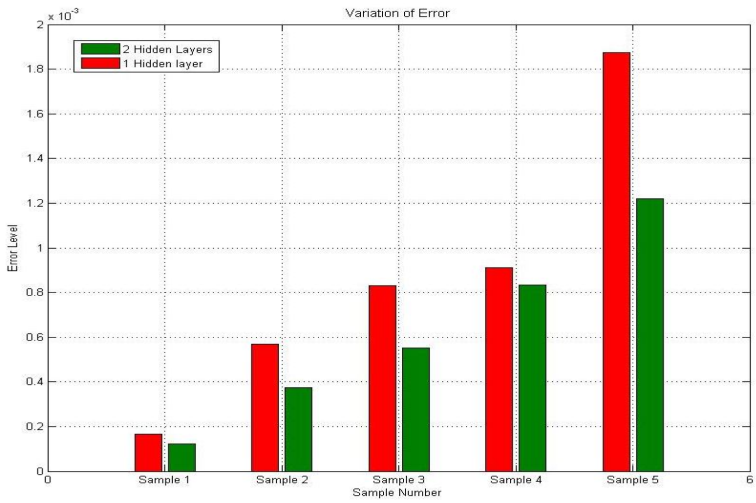

Fig. 3: Comparison of Error Values for Network with One and Two Hidden Layers.

The comparison of the error levels attained by the neural network with one hidden layer and two hidden layers is graphically shown in Fig. 3.

3.3 Testing

The character recognition accuracy of both networks with one and two hidden layers is shown in Table 3. The networks are tested with two samples. These samples are new for both the networks because they have not been trained with these samples.

It has been observed from Table 3 that in Experiment-2 employing MLP with two hidden layers, the recognition rates are better than MLP with one hidden layer.

Table 3: A Comparison of Character Recognition Accuracy.

Sample No. (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

4. Conclusion

The proposed method for the handwritten character recognition using the descent gradient approach, showed the remarkable enhancement in the performance when two hidden layers are used. It is clear from Table 3 that the recognition accuracy is best in Experiment-2 where MLP with two hidden layers is used.

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, 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. If the accuracy of the results is a critical factor for an character recognition application, then the network having two 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.

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] A. 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 MLP architecture for off-line cursive numeral recognition,” International Journal on Computer Science and Engineering, vol. 2, no.1s, pp. 1-7, 2010.