



Recognition of Noisy Numerals using Neural Network

Sundar rajan N

Research Scholar,

CMJ University,

Mehalaya, India

Dr. Rajashekhar

Professor ECE Department

Bharath University, India

ABSTRACT: *Neural networks are known to be capable of providing good recognition rate at the present of noise where other methods normally fail. Neural networks with various architectures and training algorithms have successfully been applied for letter or character recognition. This paper uses MLP network trained using Levenberg-Marquardt algorithm to recognise noisy numerals. The recognition results of the noisy numeral showed that the network could recognize normal numerals with the accuracy of 100%, blended numerals at average of 95%, numerals added with Gaussian noise at the average of 94% and partially deleted numerals at 81% accuracy.*

Keywords: *NOISY NUMERAL, MLP NETWORK,*

1. INTRODUCTION

Recently, neural network becomes more popular as a technique to perform character recognition. It has been reported that neural networks could produce high recognition accuracy. Neural networks are capable of providing good recognition at the present of noise that other methods normally fail. Neural networks with various architectures and training algorithms have successfully been applied for letter or character recognition (Avi-Itzhak et al., 1995; Costin et al., 1998; Goh, 1998; Kunasaraphan and Lursinsap, 1993; Neves et al., 1997; Parisi et al., 1998; and Zhengquan and Sayal, 1998). In general, character recognition can be divided into two that are printed font recognition and handwritten recognition. Handwritten recognition is complex due to large variation of handwritten style whereas printed character recognition is also difficult due to increase number fonts. Neural network has been applied in both cases with respectable accuracy[1].

Neves et al. (1997) has applied neural network for multi-printed-font recognition and showed that neural network was capable of recognition accurate. Some useful features have been extracted from the multi-font character and have been used to train neural network. Zhengquan and Siyal (1998) applied Higher Order Neural Network (HONNs) to recognise printed characters that have rotated at different angles. A character was divided into 16×16 pixels and only 24 letters were used. Letters 'W' and 'Z' were not used to avoid confusion with the letters 'M' and 'N'. The HONN network consists of 256 input nodes and 24 output nodes were used to recognise the letters.

Recognition of handwritten letters is a very complex problem. The letters could be written in different size, orientation, thickness, format and dimension. These will give infinity variations. The capability of neural network to generalise and be insensitive to the missing data would be very beneficial in recognising handwritten letters. Costin et al. (1998) used neural network to recognise handwritten letters for language with diacritic signs. They used some extracted features from letters together with some analytical information to train their neural network. Kunasaraphan (1993) used neural network together with a simulated light intensity model approach to recognise Thai handwritten letters[2].

Recently, neural network has also been used to recognise vehicle registration plate numbers (Goh 1998 and Parisi et al. 1998). The current study explores the capability of neural network to recognise noisy numerals. Noisy numeral or character is the common problem in vehicle registration plate recognition where the characters captured by the camera may be noisy due to dirt, rain, reflection and so on. In the current study, Multi Layered Perceptron (MLP) network trained using Levenberg-Marquardt algorithm has been used to recognise noisy number 0 to 9.

2. NOISY NUMERAL

The input data for the current study is in form of image, which will be processed before it can be used as the input of neural network. For this initial study, the numerals were printed using *Time New Roman* font of size 96. The numerals were then made noisy by:

- i.) Adding large Gaussian noise
- ii.) Blending the numerals using image processing tools
- iii.) Deleting some parts of the numerals arbitrarily

The numerals were then made noisy by adding large Gaussian white noise. The Gaussian noises with different intensity were added to printed-numerals that have been scanned into computer format. Some samples of the numerals are shown in the Figure 1. The size of numeral images will be fixed at 50×70 pixels. If the scanned numeral is large or smaller, then it will be resized so that all numerals are having the same size[4].

The numeral images were divided into 5×7 segments as in Figure (2), where each segment consists of 10×10 pixels. The mean of the grey level for each segment were then used as input data to train MLP network. Therefore, the network would have 35 input nodes (one for each segment) and 10 output nodes, where each output node represents one numeral.

3. STRUCTURE ANALYSIS OF MLP NETWORK

The recognition performance of the MLP network will highly depend on the structure of the network and training algorithm. In the current study, Levenberg-Marquardt (LM) algorithm has been selected to train the network. It has been shown that the algorithm has much better learning rate than the famous back propagation algorithm (Hagan and Menhaj, 1994).

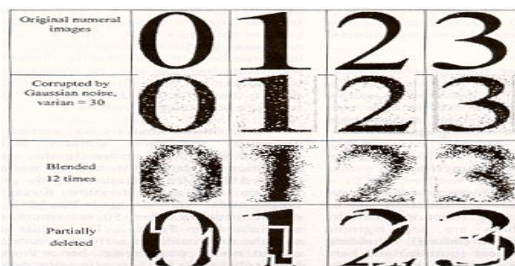


Figure 1: Samples of numerals with and without noise

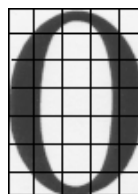


Figure 2: Numeral image is divided into 5×7

LM algorithm is an approximation of Gauss-Newton technique, which generally provides much faster learning rate than back propagation that is based on steepest decent technique.

The number of nodes in input, hidden and output layers will determine the network structure. Furthermore, hidden and output nodes have activation function that will also influence the network performance. The best network structure is normally problem dependent, hence structure analysis has to be carried out to identify the optimum structure. In the current study, the number of input and output nodes were fixed at 35 and 10 respectively, since the numeral images have been divided into 35 segments and the target outputs are 10 numerals. Therefore, only the number of hidden nodes and activation functions need to be determined. Mean squared error (MSE) will be used to judge the network performance to perform numeral recognition.

For this analysis the numeral images without noise were used to determine the structure of the network. The analysis will be divided into three that are used to determine number of hidden node, type of activation function and sufficient training epoch.

3.1 Type of Activation Function

Both hidden and output nodes have activation function and the function can be different. There are five functions that are normally used, which are log sigmoid (*logsig*), tangent sigmoid (*tansig*), saturating linear (*satlin*), pure linear (*purelin*) and hard limiter (*hardlin*). For this analysis, the number of hidden nodes was taken as 9 and the maximum epoch was set to 50. The value of MSE at the final epoch was taken as an indication of the network performance. When the activation functions for the output nodes were selected as log sigmoid and the activations of the hidden nodes were changed, the MSE values as in Table 1 were obtained.[6]

The results in Table 1 suggest that the best-hidden node activation function is *tansig*. The results also revealed that in general sigmoidal functions (*logsig* and *tansig*) are generally much better than linear functions such as *purelin*, *satlin* and *hardlin*. In case of *hardlin* and *purelin*, the training process stopped after 10 epochs because the minimum gradient of the learning rate has been reached. In other words, there was no significant learning after 10 epochs when the two functions were used. Based on these results the hidden node activation function has been selected to be *tansig*.

Table 1: MSE variation with different activation function of hidden nodes

Activation Function of Hidden Nodes	MSE
Logsig	2.5854×10^{-11}
Tansig	1.9590×10^{-12}
Purelin	0.2814
Satlin	0.0448
Hardlin	0.1750

A similar analysis was carried out to determine the most suitable activation function for output nodes. In this case the activation function for hidden nodes was selected as *tansig* determined by the previous analysis and the activation function for output nodes were changed. The MSE values of the network after 50 maximum epochs were shown in Table 2. The results showed that the most suitable activation function for output nodes was *logsig*; hence *logsig* has been selected as the activation function of the output nodes.[5]

Table 2: MSE variation with different activation function of output nodes

Activation Function of Hidden Nodes	MSE
Logsig	1.9590×10^{-12}
Tansig	0.0385
Purelin	0.0348
Satlin	2.5349×10^{-11}
Hardlin	0.9000

3.2 Number of Hidden Nodes

The number of hidden nodes will heavily influence the network performance. Insufficient hidden nodes will cause underfitting where the network cannot recognise the numeral because there are not enough adjustable parameter to model or to map the input-output relationship. However, excessive hidden nodes will cause overfitting where the network fails to generalise. Due to excessive adjustable parameters the network tends to memorise the input-output relation hence the network could map the training data very well but normally fail to generalise. Thus, the network could not map the independent or testing data properly. One way to determine the suitable number of hidden node is by finding the minimum MSE calculated over the testing data set as the number of hidden nodes is varied (Mashor, 1999).

In this analysis, the activation functions for hidden and output nodes have been selected as in previous section. The target MSE was set to 1.0×10^{-11} and the maximum training epoch was set to 50 epochs. However, some training processes have been stopped before 50 epochs because minimum gradient of the learning rate has been reached. The number of hidden nodes was increased from 1 to 14 and the MSE values as in Table 3 were obtained. It is obvious from the table that the minimum MSE occurs at 9 hidden, hence the number of hidden nodes has been selected as 9.[7]

3.3 Number of Training Epochs

Training epoch also plays an important role in determining the performance of neural network[7]. Insufficient training epoch causes the adjustable parameters not to converge properly. However, excessive training epochs will take unnecessary long training time. The sufficient number of epochs is the number that will satisfy the aims of training. For example if our objective is to recognise the numeral, then the number of training epoch after which the network is capable to recognise the numeral accurately can be considered as sufficient. Sometimes this objective is hard to be achieved even after thousand of training epochs. This could be due to a very complex problem to be mapped or inefficient training algorithm. In this case, the profile of MSE plot for the training epochs should indicate that further training epoch could no longer beneficial. Normally, the MSE plot would have a saturated or flat line at the end.

MSE is the performance index that was used in this study to determine the training degree of the network. In general, the lower the value of MSE the better the network has been trained. The MSE values of the network that was used in this study were shown in Figure 3. The target MSE was set to 1.0×10^{-11} and this objective was achieved in 50 epochs where the MSE value already reached 1.959×10^{-12} . The final value of MSE is very small which indicates that the network is capable of mapping the input and output accurately.

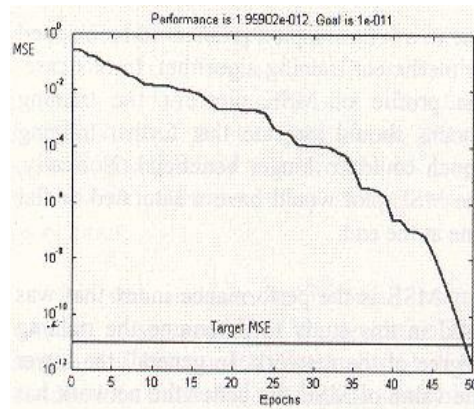


Figure 3: The variation of MSE with the training epochs

Table 3: The MSE values as the number of hidden nodes were increased

Number of Hidden Nodes	MSE
1	0.1421
2	0.0485
3	0.0508
4	0.0745
5	0.0809
6	0.0933
7	0.1183
8	0.0386
9	1.9590×10^{-12}
10	0.0039
11	0.1691
12	0.0025
13	0.0014
14	0.0014

4. NOISY NUMERAL RECOGNITION

Some analyses have been carried out to determine the best structure for the MLP network to perform the numerals recognition. It has been found that 9 hidden nodes were the optimum for the network. The training data consist of:

- i.) Normal numerals
- ii.) Numerals that have been blended for 3 and 12 times.
- iii.) Numerals that have been added with Gaussian noise with variance of 5 to 30.

The targets were all set to the normal numerals. The network was then trained using Levenberg-Marquardt algorithm and it converged properly after 50 epochs. [8]

The trained network was then tested with four types of numerals that were:

- i.) Normal numerals
- ii.) Numerals that have been blended for several times (3, 6, 9 and 12 times)
- iii.) Numerals that have been added with Gaussian noise with various intensity (variance from 5 to 75) and
- iv.) Numerals that have been partially deleted.

The samples of the numerals are shown in Figure 4.

The recognition results showed that the network could recognize normal numerals with the accuracy of 100%, blended numerals at average of 95%, numerals added with Gaussian noise at the average of 94% and partially deleted numerals at 81% accuracy.

Based on these results it can be concluded that the network can be used to recognize noisy numerals or letters. Hence, with some further studies this technique could be explored and used as part of vehicle registration plate recognition system.

5. CONCLUSION

This paper uses MLP network trained using Levenberg-Marquardt algorithm to recognise noisy numerals. The results of structure analysis suggest that the appropriate structure should have 9 hidden nodes with *tansig* function in hidden nodes and *logsig* function in output nodes. The appropriate training epochs was found to be 50 epochs in order to achieve the required gold. The recognition results of the noisy numeral showed that the network could recognize normal numerals with the accuracy of 100%, blended numerals at average of 95%, numerals added with Gaussian noise at the average of 94% and partially deleted numerals at 81% accuracy. Based on these results it can be concluded that the network can be used to recognize noisy numerals successfully.

REFERENCES

- [1] Avi-Itzhak H.I., Diep T.A. and Garland H., 1995, "High Accuracy Optical Character Recognition Using Neural Networks with Centroid Dithering", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, No.2, pp. 218-223.
- [2] Costin H., Ciobanu A. and Todirascu A., 1998, "Handritten Script Recognition System for Languages with Diacritic Signs", *Proc. Of IEEE Int. Conf. On Neural Network*, Vol. 2, pp. 1188-1193.
- [3] Goh K. H., 1998, "The use of Image Technology for Traffic Enforcement in Malaysia", *MSc. Thesis*, Technology Management, Staffordshire University, United Kingdom.
- [4] Hagan M.T. and Menhaj M., "Training Feedback Networks with the Marquardt Algorithm", *IEEE Trans. on Neural Networks*, Vol. 5, No. 6, pp. 989-993.
- [5] Kunasaraphan C. and Lursinsap C., 1993, "44 Thai Handwritten Alphabets Recognition by Simulated Light Sensitive Model", *Intelligent Eng. Systems Through Artificial Neural Network*, Vol.3, pp. 303-308.
- [6] Mashor M.Y., 1999, "Some Properties of RBF Network with Applications to System identification", *Int. J. of Computer and Engineering Management*, Vol. 7 No. 1, pp. 34-56.
- [7] Neves D.A., Gonzaga E.M., Slaets A. and Frere A.F., 1997, "Multi-font Character Recognition Based on its Fundamental Features by Artificial Neural Networks", *Proc. Of Workshop on Cybernetic Vision*, Los Alamitos, CA, pp. 116-201.
- [8] Parisi R., Di Claudio E.D., Lucarelli G. and Orlandi G., 1998, "Car Plate Recognition by Neural Networks and Image Processing", *IEEE Proc. On Int. Symp. On Circuit and Systems*, Vol. 3, pp. 195-197.
- [9] Zhengquan H. and Siyal M.Y., 1998, "Recognition of Transformed English Letter with Modified Higher-Order Neural Networks", *Electronics Letter*, Vol. 34, No.2, pp. 2415-2416.
- [10] Vats, Kuldeep, et al. "Simulation and performance analysis of OLSR routing protocol using OPNET." *International Journal* 2.2 (2012).