RESEARCH ARTICLE OPEN ACCESS

Intelligent Handwritten Digit Recognition using Artificial Neural Network

Saeed AL-Mansoori
Applications Development and Analysis Center (ADAC), Mohammed Bin Rashid Space Center (MBRSC), United Arab Emirates

ABSTRACT
The aim of this paper is to implement a Multilayer Perceptron (MLP) Neural Network to recognize and predict handwritten digits from 0 to 9. A dataset of 5000 samples were obtained from MNIST. The dataset was trained using gradient descent back-propagation algorithm and further tested using the feed-forward algorithm. The system performance is observed by varying the number of hidden units and the number of iterations. The performance was thereafter compared to obtain the network with the optimal parameters. The proposed system predicts the handwritten digits with an overall accuracy of 99.32%.

Keywords – Accuracy, Back-propagation, Feed-forward, Handwritten Digit Recognition, Multilayer Perceptron Neural Network, MNIST database.

I. INTRODUCTION

Handwritten digit recognition has been a major area of research in the field of Optical Character Recognition (OCR). Based on the input to the system, handwritten digit recognition can be categorized into online and offline recognition. In the online mode, the movements of a pen on a pen-based software screen surface were used to provide input into the system designed to predict the handwritten digits. Meanwhile, the offline mode uses an interface such as a scanner or camera as input to the system [1]. The conversion of an image based on the digit contained to letter codes for further use in a computer or text processing application is the prior step in an off-line handwriting recognition system. This form of data provides a static representation of any handwriting contained. The task of recognizing the handwriting of an individual from another is difficult as each personal possess a unique handwriting style. This is one reason as to why handwriting is considered as one of the main challenging studies. The need for handwritten digit recognition came about the time when combinations of digits were included in records of an individual. The current scenario calls for the need of handwritten digit recognition in banks to identify the digits on a bank cheque and also to collect other user account related information. Moreover, it can be used in post offices to identify pin code box numbers, as well as in pharmacies to identify the doctors’ prescriptions. Although there are several image processing techniques designed, the fact that the handwritten digits do not follow any fixed image recognition pattern in each of its digits makes it a challenging task to design an optimal recognition system. This study concentrates on the offline recognition of digits using an MLP neural network. Many methods have been proposed till date to recognize and predict the handwritten digits. Some of the most interesting are those briefly described below. A wide range of researches has been performed on the MNIST database to explore the potential and drawbacks of the best recommended approach. The best methodology till date offers a training accuracy of 99.81% using the Convolution Neural Network for feature extraction and an RBF network model for prediction of the handwritten digits [2]. According to [3] an extended research conducted for identifying and predicting the handwritten digits attained from the Concordia University database, Mexican hat wavelet transformation technique was used for preprocessing the input data. With the help of the back propagation algorithm, this input was used to train a multilayered feed forward neural network and thereby attained a training accuracy of 99.17%. Although higher than the accuracies obtained for the same architecture without data preprocessing, the testing for isolated digits was estimated to be just 90.20%. A novel approach based on radon transform for handwritten digit recognition is reported in [4]. The radon transform is applied on range of theta from -45 to 45 degrees. This transformation represents an image as a collection of projections in various directions resulting in a feature vector. The feature vector is then fed as an input to the classification phase. In this paper, authors are used the nearest neighbor classifier for digit recognition. An overall accuracy of 96.6% was achieved for English handwritten digits, whereas 91.2% was obtained for Kannada digits. A comparative study in [5] was conducted by training the neural network using back-propagation algorithm and further using PCA for feature extraction. Digit recognition was finally carried out using the thirteen algorithms, neural network algorithm and FDA algorithm. The FDA algorithm proved less efficient with an overall accuracy of 77.67%, whereas the back-propagation algorithm with PCA for its feature extraction gave an accuracy of 91.2%. In 2014 [6], a novel approach using SVM binary classifiers and unbalanced decision trees was presented. Two classifiers were proposed in this study, where one uses the digit characteristics as input and the other using the whole image as such. It was observed that a handwritten digit recognition accuracy of 100% was achieved. In [7] authors presented the rotation variant feature vector algorithm to train a probabilistic neural network. The proposed system has been trained on samples of 720 images and tested on samples of 480 images written by 120 persons. The recognition rate was achieved at 99.7%. The use of multiple classifiers reveals multiple aspects of handwriting samples that help to better identify hand-written characters. A DWT classifier and a Fourier transform classifier aids to a better decision making ability of the entire classification system [8]. Data pre-processing plays a very significant role in the precision of the handwritten character identification. It has been proved that the practice of using different data processing techniques coupled together have led to a better trained neural network and also to improve computational efficiency of the training mechanism. The choice of data preprocessing technique and the training algorithm are extremely important for better training but can only be determined on a trial and error basis [9]. In this paper, the proposed handwritten digit recognition algorithm is based on gradient descent back-propagation. The rest of the paper is organized as follows. Section II briefly introduces the database used in this study. The network architecture and training mechanism is described in detail in section III. Section IV presents simulation results to demonstrate the performance of the proposed mechanism. Finally, the conclusion of the work is given in section V.

II. Data Collection

In this study, a subset of 5000 samples was conducted from the MNIST database. Each sample was a gray-scale image of size (20×20) pixels. Figure 1 shows some sample images. The input dataset was obtained from the database with 500 samples of each digit from 0 to 9. As the common rule, a random 4000 samples of the input dataset were used for training and the remaining 1000 were used for validation of the overall accuracy of the system. A dataset of 100 samples for each isolated digit from 0 to 9 were further used for confirming the accurate predictive power of the network. The training set and the test set were kept distinct to help the neural network learn from the training dataset and thereafter predict the test set rather than merely memorizing the entire dataset and then reciprocating the same.

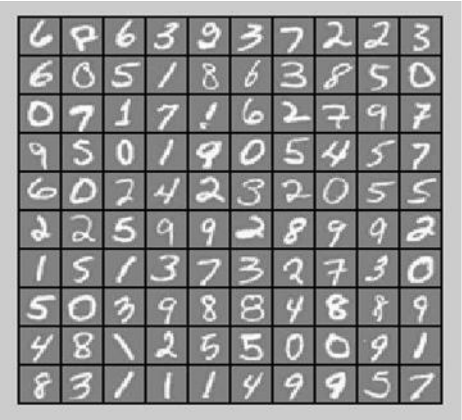

Figure1: Sample handwritten digits from MNIST

III. Proposed Methodology

A. Neural Network Architecture

Figure 2 illustrates the architecture of the proposed neural network. It consists of an input layer, hidden layer and an output layer. The input layer and the hidden layer are connected using weights represented as W_{ij}, where *i* represents the input layer and *j* represents the hidden layer. Similarly, the weights connecting the hidden and output layer are termed as W_{jk} , where, k represents the output layer. A bias of +1 is included in the neural network architecture for efficient tuning of the network parameters. In this MLP neural network, sigmoid activation function was used for estimating the active sum at the outputs of both hidden and output layers. The sigmoid function is defined as shown in equation 1 and it returns a value within a specified range of [0, 1].

$$g(x) = \frac{1}{1+e^{-x}} \quad (1)$$

Here, *g(x)* represents the sigmoid function and the net value of the weighted sum is denoted as *x*.

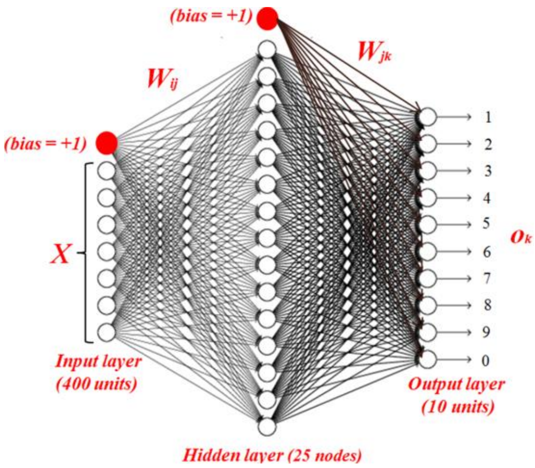

Figure2: The proposed neural network architecture

Input Layer

The 400 pixels extracted from each image is arranged as a single row in the input vector ***X***. Hence, vector ***X*** is of size (5000×400) consisting of the pixel values for the entire 5000 samples. This vector ***X*** is then given as the input to the input layer. Considering the above mentioned specifications, the input layer in the neural network architecture consists of 400 neurons, with each neuron representing each pixel value of vector ***X*** for the entire sample, considering each sample at a time.

Hidden Layer

As per various studies conducted in the field of artificial neural network, there do not exist a fixed formula for determining the number of hidden neurons. Researchers working in this field have however proposed various assumptions to initialize a cross validation method for fixing the hidden neurons [10]. In this study, geometric mean was initially used to find the possible number of hidden neurons. Thereafter, the cross validation technique was applied to estimate the optimal number of hidden neurons. Figure 3 shows a comparative study of the neural network training and testing for 5000 samples with respect to various hidden neurons.

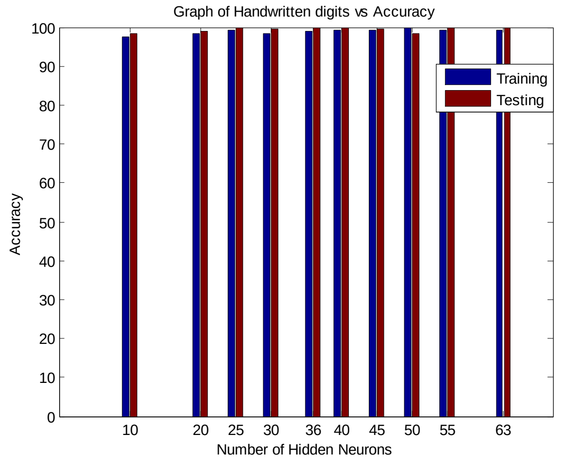

Figure3: Variation of accuracy w.r.t number of hidden neurons

As observed, a neural network with hidden neuron numbers 63, 55, 45, 40, 36 and 25 produced almost similar accuracies for testing and training. A neural network of hidden neuron number 25 was fixed for further training and testing to reduce the cost during its real time realization.

Output Layer

The targets for the entire 5000 sample dataset were arranged in a vector Y of size (5000×1). Each digit from 0 to 9 was further represented as y_k with the neuron giving correct output to be 1 and the remaining as 0. Hence, the output layer consists of 10 neurons representing the 10 digits from 0 to 9.

$$\begin{array}{cccccccccc}
\left[\begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix}\right] \\
y_k^{(1)} \; y_k^{(2)} \; y_k^{(3)} \; y_k^{(4)} \; y_k^{(5)} \; y_k^{(6)} \; y_k^{(7)} \; y_k^{(8)} \; y_k^{(9)} \; y_k^{(10)}
\end{array} \tag{2}$$

B. Gradient descent back propagation algorithm

The proposed design uses the gradient descent back-propagation algorithm for training 4000 samples and feed-forward algorithm for testing the remaining 1000 samples. The aim behind using back-propagation algorithm is to minimize the error between the outputs of the output neurons and the targets.

Step 1: Initialize the parameters

The parameters to be optimized in this study are the connecting weights, W_{jk} and W_{ij} where W_{jk} denotes the connecting weights between the hidden and output neurons and W_{ij} is the connecting weights between the input and hidden layers. The weights were randomly initialized along with other parameter as shown in Table 1.

Table1: Initialization of parameters

Parameters	**Values assigned**
W_{ij}	-0.121 to 0.121
W_{jk}	-0.121 to 0.121
Learning parameter, η	0.1

Step 2: Feed-forward Algorithm

The 400 pixels of a sample were provided as input x_i (x_i=x) at the input layer. The input was further multiplied with the weights connecting the input and hidden layer to obtain the net input to the hidden layer represented as x_j.

$$x_j = W_{ij} x_i \tag{3}$$

Sigmoid of the net input was calculated at the hidden layer to obtain the output of the hidden layer, o_j.

$$o_j = g(x_j) \tag{4}$$

where $g(x_j)$ is the sigmoid function. Weighted sum of o_j was further provided as input to output layer represented as x_k.

$$x_k = W_{jk} o_j \tag{5}$$

The sigmoid of x_k was calculated to obtain the output of the output layer which was thereby the final output of the network.

$$o_k = g(x_k) \quad (6)$$

Once the final output was obtained, it was compared with the target representation y_k to obtain the error to be minimized. Mean square error,

$$MSE = \frac{1}{2}\sum_{k=1}^{K}(o_k - y_k)^2 \quad (7)$$

where, K=10, o_k denotes the output of the output neurons and y_k is the target representation of the output neurons. The same was repeated for the entire 4000 samples for training and the average mean of the entire dataset was calculated to obtain the overall training accuracy.

Step 3: Calculate the gradient

The gradient value for both output and hidden layers were calculated for updating the weights. The gradient was obtained by evaluating the derivative of the error to be minimized.

$$\delta_k = o_k(1-o_k)(o_k - y_k) \quad (8)$$

$$\delta_j = o_j(1-o_j)\sum_{k=1}^{K}\delta_k W_{jk} \quad (9)$$

where δ_k and δ_j are the gradients of the output layer and hidden layer respectively.

Step 4: Update the weights

The weights were obtained as a function of the error using a learning parameter.

$$\Delta W_{jk} = -\eta\delta_k o_j \quad (10)$$

$$\Delta W_{ij} = -\eta\delta_j o_i \quad (11)$$

$$W_{jk_{new}} = W_{jk_{old}} + \Delta W_{jk} \quad (12)$$

$$W_{ij_{new}} = W_{ij_{old}} + \Delta W_{ij} \quad (13)$$

where, ΔW_{jk} denotes the weight updates of the weights connecting the hidden and output layer and ΔW_{ij} represents the weight updates of the weights connecting the input and hidden layer. Continue steps 1 to 4 for maximum number of iterations until an acceptable minimum error was obtained.

IV. Experimental Results

In this section, the performance of the proposed handwritten digit recognition technique is evaluated experimentally using 1000 random test samples. The experiments are implemented in MATLAB 2012 under a Windows7 environment on an Intel Core2 Duo 2.4 GHz processor with 4GB of RAM and performed on gray-scale samples of size (20×20). The accuracy rate is used as assessment criteria for measuring the recognition performance of the proposed system. The accuracy rate is expressed as follows:

$$\%Accuracy = \frac{No.of\ test\ samples\ classified\ correctly}{Total\ No.of\ samples} \times 100\% \quad (14)$$

The following sets of experiments are conducted in this section:

A. Estimate the maximum number of iterations providing the maximum accuracy.

The training and testing was continued at an interval of 50 until an acceptable accuracy was obtained. The relation between the number of iterations and the accuracy of training and testing were observed and recorded as shown in Table 2.

Table2: Number of iterations Vs Accuracy

Iteration #	*Training*	*Testing*
50	94.78	96.87
100	98.28	98.43
150	98.86	96.87
200	99.10	100
250	**99.32**	**100**

B. Test the accuracy of each handwritten digit from 0 to 9.

The Neural network trained for 250 iterations was used to test 100 samples of digits from 0 to 9 individually and the testing accuracies were observed as shown in figure 4.

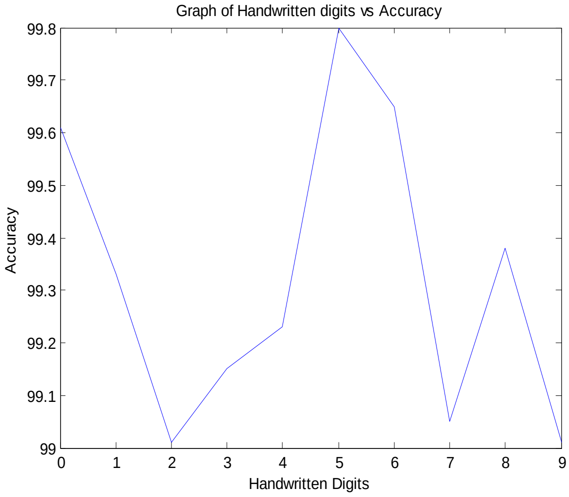

Figure4: Testing accuracy of handwritten digits

It was observed that digit 5 was predicted at the highest accuracy of 99.8% while digit 2 was predicted with the lowest accuracy of 99.04%.

C. Prediction of 64 test samples

To verify the predictive power of the system, 64 random samples out of 1000 samples were tested for various iterations ranging from 50 to 250. From figure 5, it was noticed that the accuracy at 50 iterations was low and hence the digits were predicted incorrectly. The incorrect predictions of the handwritten digits are circled as shown in figure5. Figure 6 illustrates the best results obtained at 250 iterations, where all handwritten digits were predicted correctly.

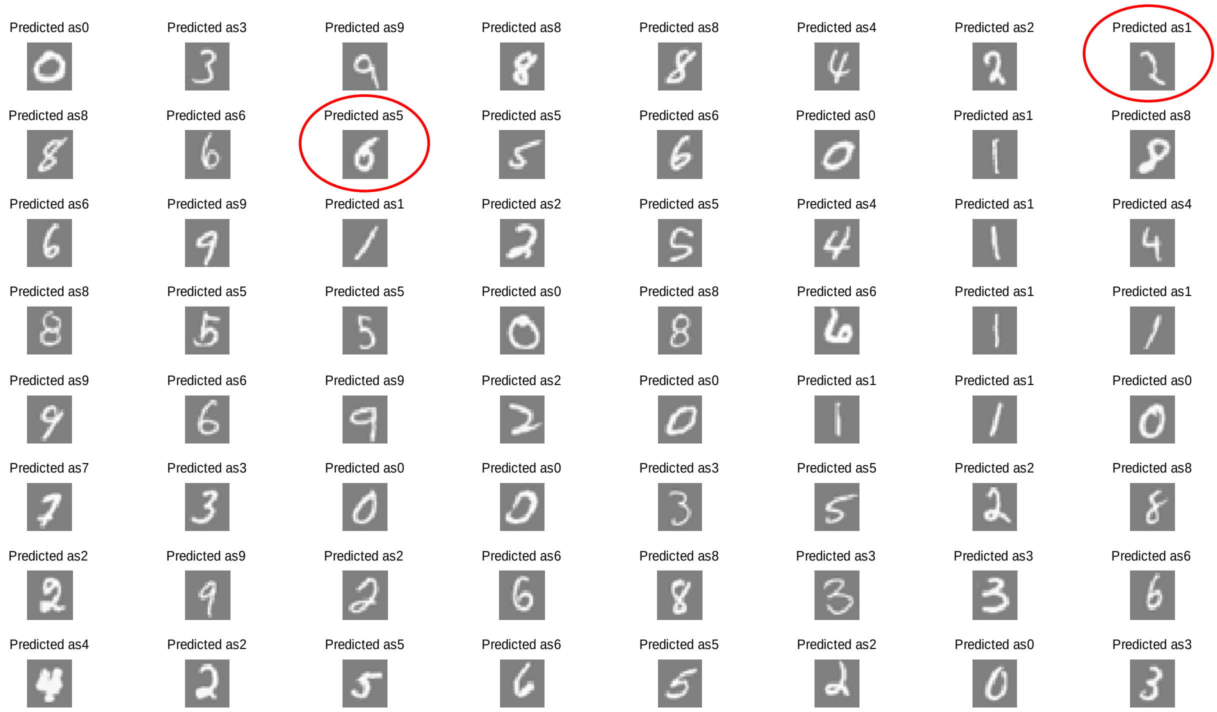

Figure5: Testing of 64 samples with 50 iterations

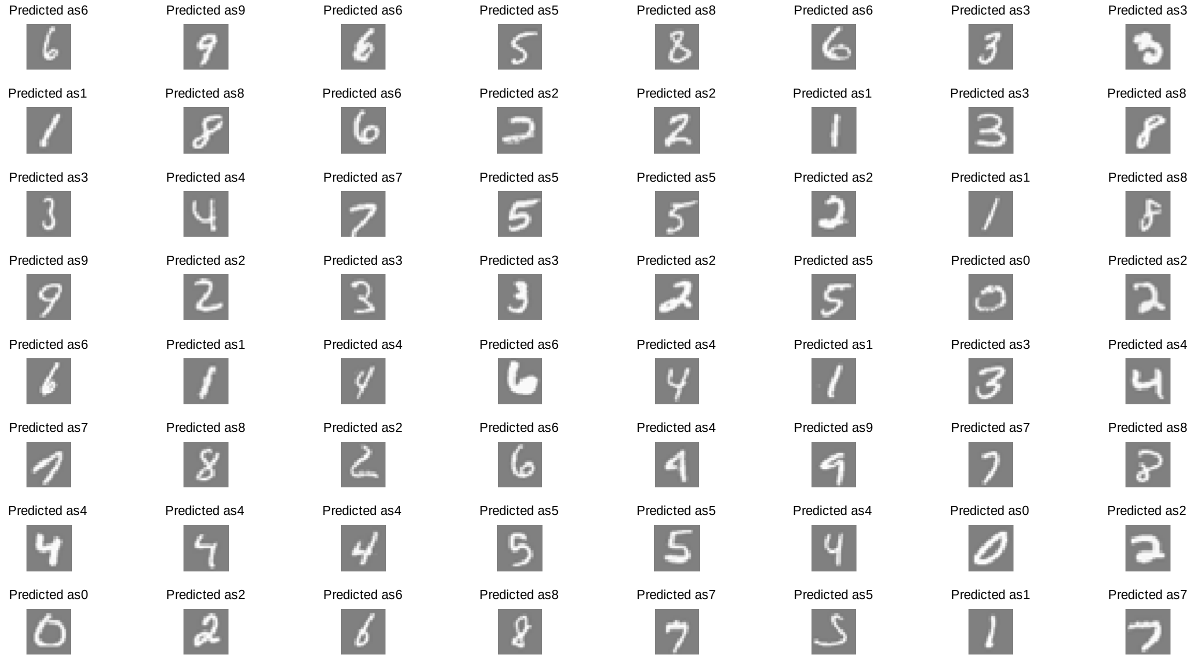

Figure6: Testing of 64 samples with 250 iterations

V. Conclusion

In this paper, a Multilayer Perceptron (MLP) Neural Network was implemented to address the handwritten digit recognition problem. The proposed neural network was trained and tested on a dataset attained from MNIST. The system performance was observed by varying the number of hidden units and the number of iterations. A neural network architecture with hidden neurons 25 and maximum number of iterations 250 were found to provide the optimal parameters to the problem. The proposed system was proved efficient with an overall training accuracy of 99.32% and testing accuracy of 100%.

References

[1] R. Plamondon and S. N. Srihari, On-line and off- line handwritten character recognition: A comprehensive survey, *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, no. 1, 2000, 63-84.

[2] Xiao-Xiao Niu and Ching Y. Suen, A novel hybrid CNN–SVM classifier for recognizing handwritten digits, ELSEVIER, *The Journal of the Pattern Recognition Society, Vol. 45*, 2012, 1318– 1325.

[3] Diego J. Romero, Leticia M. Seijas, Ana M. Ruedin, Directional Continuous Wavelet Transform Applied to Handwritten Numerals Recognition Using Neural Networks, *JCS&T, Vol. 7 No. 1*, 2007.

[4] V.N. Manjunath Aradhya, G. Hemantha Kumar and S. Noushath, Robust Unconstrained Handwritten Digit Recognition using Radon Transform, *IEEE-ICSCN*, 2007, 626-629.

[5] Zhu Dan and Chen Xu, The Recognition of Handwritten Digits Based on BP Neural Network and the Implementation on Android, *Fourth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA)*, 2013, 1498-1501.

[6] Adriano Mendes Gil, Cícero Ferreira Fernandes Costa Filho, Marly Guimarães Fernandes Costa, Handwritten Digit Recognition Using SVM Binary Classifiers and Unbalanced Decision Trees, *Image Analysis and Recognition, Springer*, 2014, 246-255.

[7] Al-Omari F., Al-Jarrah O, Handwritten Indian numerals recognition system using probabilistic neural networks, *Adv. Eng. Inform*, 2004, 9–16.

[8] Junchuan Yanga, ,Xiao Yanb and Bo Yaoc, Character Feature Extraction Method based on Integrated Neural Network, A*ASRI Conference on Modelling, Identification and Control, ELSEVIER, AASRI Pro*.

[9] Nazri Mohd Nawi, Walid Hasen Atomi and M. Z. Rehman, The Effect of Data Pre-Processing on Optimized Training of Artificial Neural Networks, *Procedia Technology, ELSEVIER, 11*, 2013, 32 – 39.

[10] Asadi, M.S., Fatehi, A., Hosseini, M. and Sedigh, A.K. , Optimal number of neurons for a two layer neural network model of a process, *Proceedings of SICE Annual Conference (SICE), IEEE*, 2011, 2216 – 2221.