AGENT BASED INTELLIGENT SYSTEMS PROJECT

REPORT

Topic: Handwritten Digit Recognizer

Vasu Agarwal 14BCE0592

Zeeshan Khan Pathan 14BCE05

Slot: A1

**Personal Research Views**

Summary

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%.

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. 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.

This project 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.

Data Collection

A subset of 5000 samples was conducted from the MNIST database. Each sample was a gray-scale image of size (20×20) pixels. The input dataset was obtained from the database with 500 samples of each digit from 0 to 9.



Methodology Used

*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 *Wij*, where *i* represents the input layer and *j* represents the hidden layer. Similarly, the weights connecting the hidden and output layer are termed as *Wjk* , where, k represents the output layer. A bias of +1 is included in the neural network architecture for efficient tuning of the network parameters.

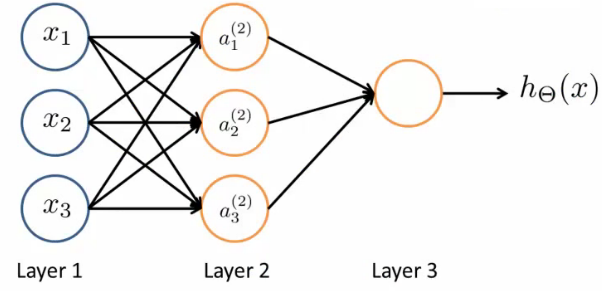
The sigmoid function is defined as shown in equation 1 and it returns a value within a specified range of [0, 1]

Sigmoid Function: f(x) = 1/(1+e^(-x))

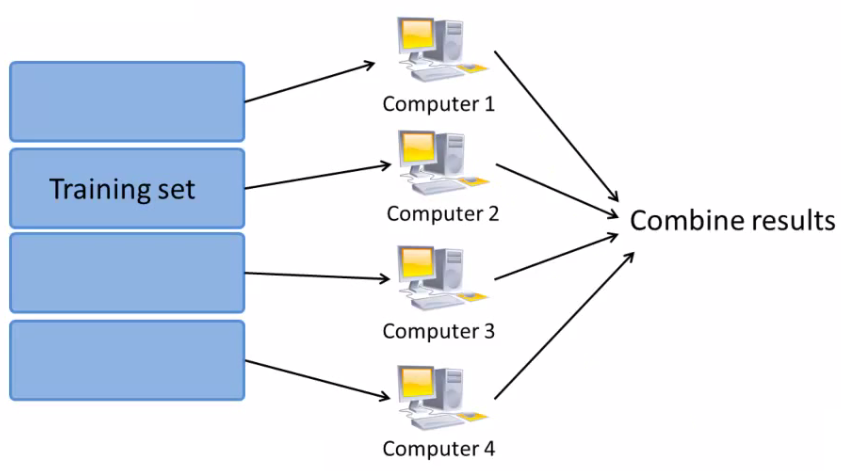


*Input Layer*: The 400 pixels extracted from each image is arranged as a single row in the input vector ***X***. Hence, vector ***X*** is of size (5000×400) consisting of the pixel values for the entire 5000 samples. This vector ***X*** is then given as the input to the input layer. Considering the above-mentioned 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.

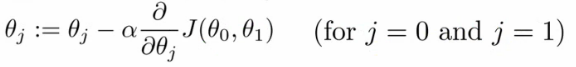
*Hidden Layer*: A geometric mean was initially used to find the possible number of hidden neurons. Thereafter, the cross-validation technique was applied to estimate the optimal number of hidden neurons. Figure 3 shows a comparative study of the neural network training and testing for 5000 samples with respect to various hidden neurons.

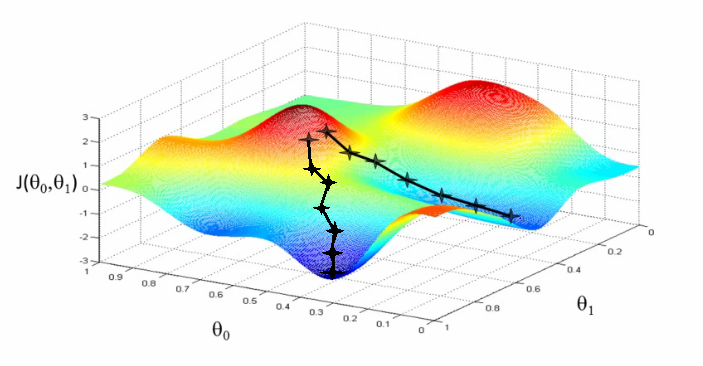


*Output Layer*: The targets for the entire 5000 sample dataset were arranged in a vector Y of size (5000×1). Each digit from 0 to 9 was further represented as *yk* with the neuron giving correct output to be 1 and the remaining as 0. Hence, the output layer consists of 10 neurons representing the 10 digits from 0 to 9. Once the final output was obtained, it was compared with the target representation *yk* to obtain the error to be minimized.

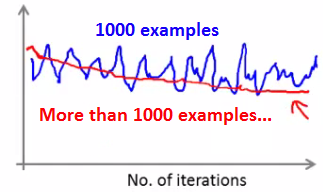


*Calculating gradient*: The gradient value for both output and hidden layers were calculated for updating the weights. The gradient was obtained by evaluating the derivative of the error to be minimized.





*Updating weights*: The weights were obtained as a function of the error using a learning parameter where, *ΔWjk* denotes the weight updates of the weights connecting the hidden and output layer and *ΔWij* represents the weight updates of the weights connecting the input and hidden layer.

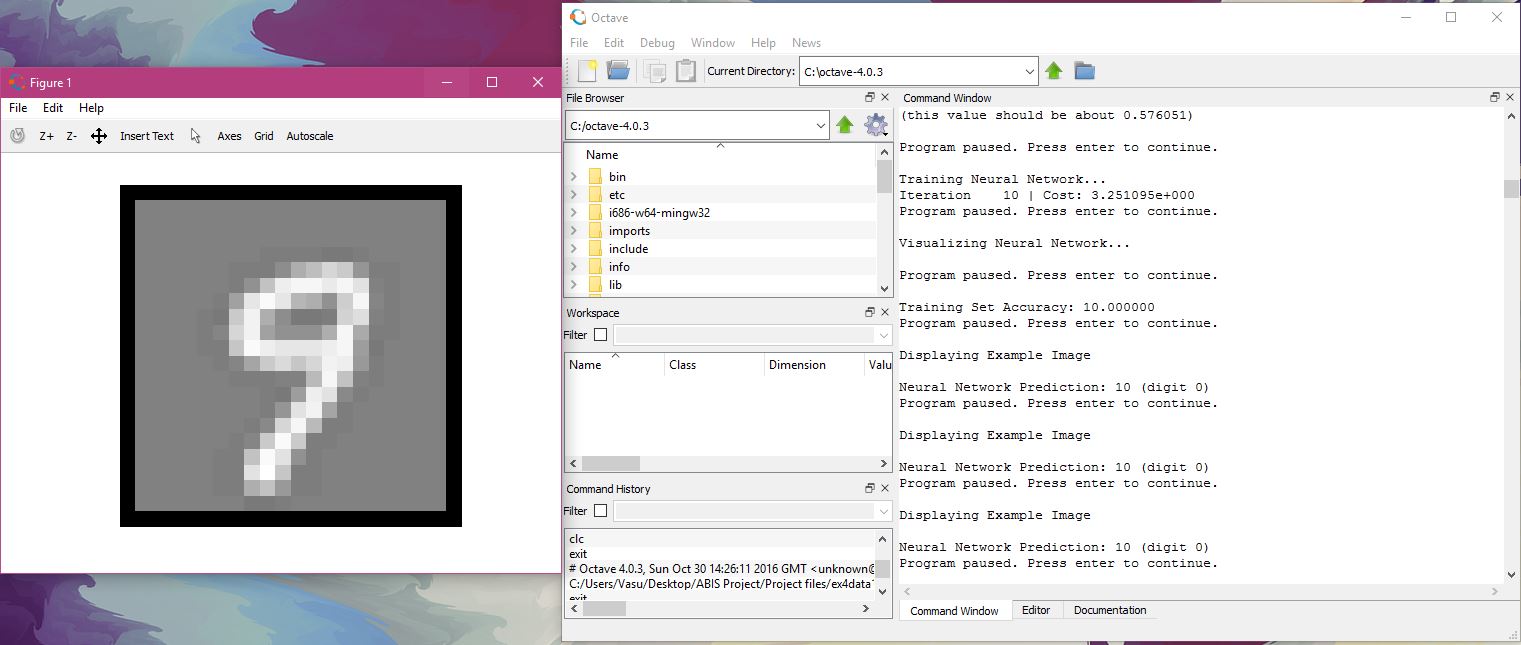


Experimental Results

The program was implemented using the open source software octave.

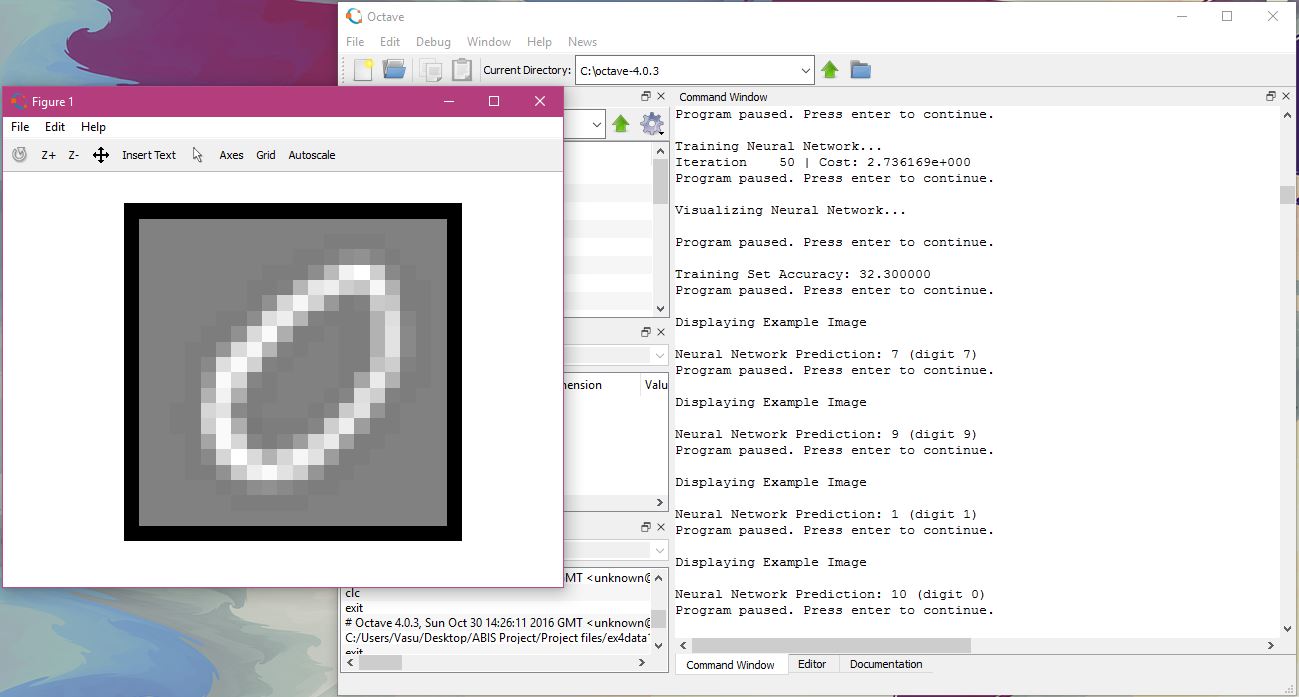
The experiment was performed with various number of learning algorithm iterations: 10,50,100,250,500 and then the corresponding accuracy of results was noted.

10 iterations:



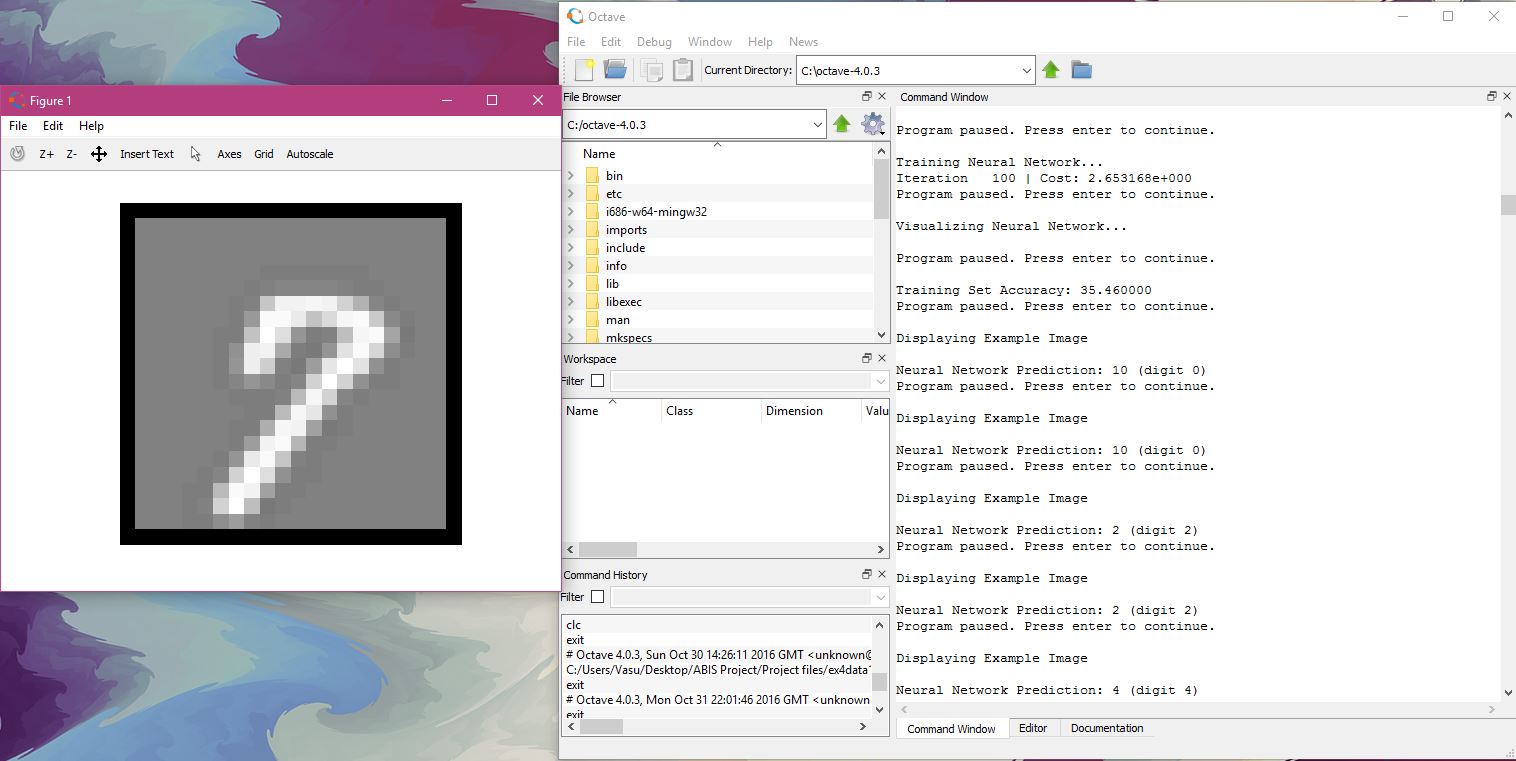
Accuracy: 10.000000

50 iterations:



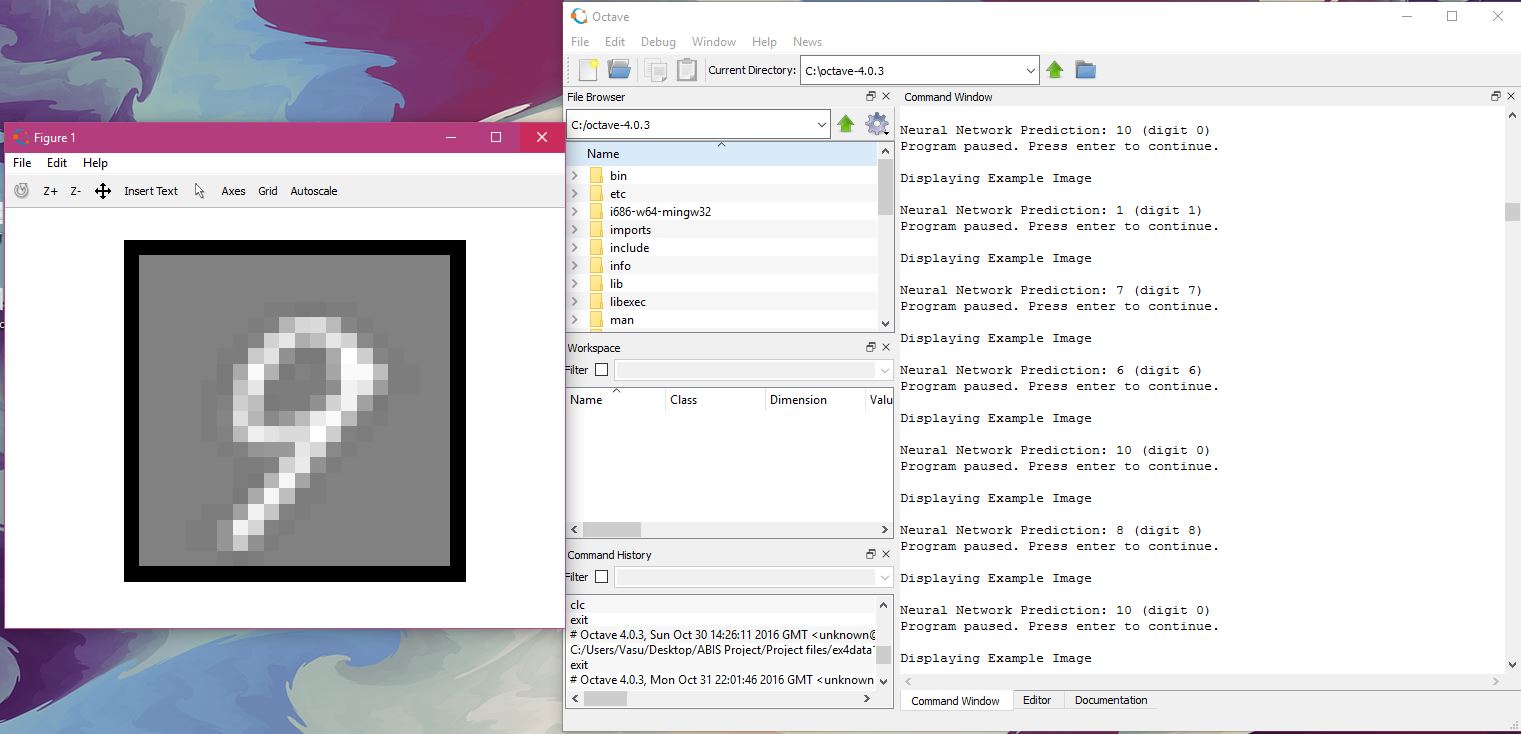
Accuracy: 32.300000

100 iterations:



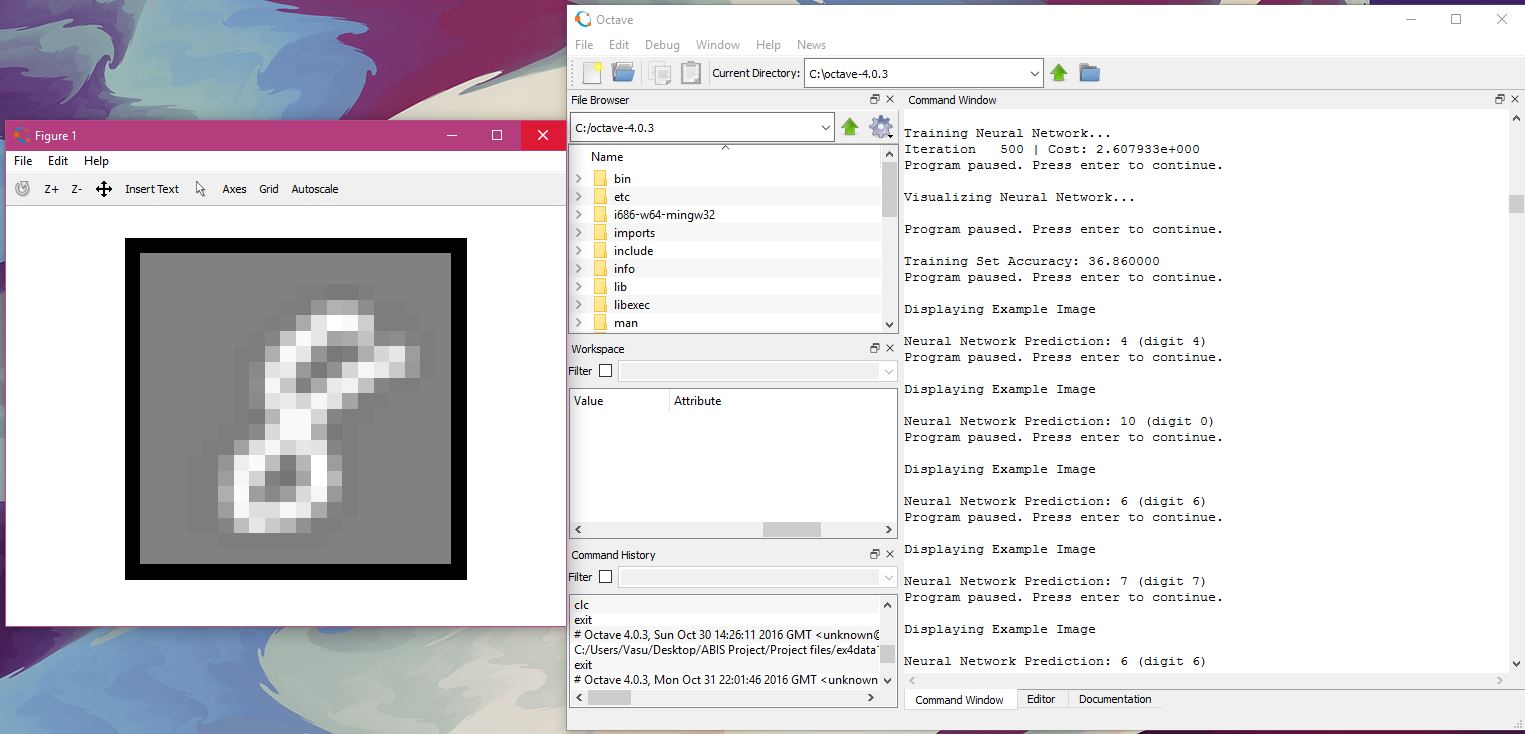
Accuracy: 35.4600000

250 iterations:



Accuracy: 35.8900000

500 iterations:



Accuracy: 36.86000000

Conclusion:

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.

We observed that the amount of accuracy increased drastically at first and then slowly as we increased the number of iterations.

**Literature Survey and References**

Research papers:

**Intelligent Handwritten Digit Recognition using Artificial Neural Network**

*Saeed AL-Mansoori Int. Journal of Engineering Research and Applications*

*ISSN: 2248-9622, Vol. 5, Issue 5, (Part -3) May 2015, pp.46-51*

**Fast Efficient Artificial Neural Network for**

**Handwritten Digit Recognition**

*Viragkumar N. Jagtap et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2), 2014, 2302-2306*

**Comparison of learning algorithms for handwritten digit** recognition

*Y. LeCun, L. Jackel, L. Bottou - Bell Labs*

**Reading Handwritten digits: A zip code recognition system**

*AT&T Bell Laboratories, Holmdel, N.J. 07733*

**Handwritten Character and Digit Recognition Using Artificial Neural Networks**

*K.Venkata Reddy, D.Rajeswara Rao, U.Ankaiah, K.Rajesh*

*Department of computer science engineering,*

*KL University, Guntur, AP, India*

**Other references:**

*Machine Learning course by Andrew Ng (Stanford University) on Coursera*

[www.kaggle.com](http://www.kaggle.com)

**Summary of survey**

Handwritten digit recognition plays important role in the modern world. It can solve more complex problems and makes human’s job easier. There are different techniques that can be used to recognize handwritten digits. We use the multilayer perception artificial neural network to recognize the handwritten digits. Digit reorganization device is one of such smart devices that acquire partial human intelligence with the ability to capture and recognize various digits. In the MLP network is use the back-propagation algorithm to train and test the data.

Back Propagation Algorithm:

Back propagation algorithm consists of two phases. First phase is forward phase. This is the phase where the activations propagate from input layer to the output layer. The second phase is backward phase. This phase where the error between the observed actual value and the requested nominal value in the output layer are propagated backwards so it can modify the weights and bias values.

Feed-Forward Phase:

The GPU program aims for a fully parallelization of the

feed-forward operation using a huge number of threads to

avoid any looping. This means every GPU-thread is

therefore, responsible for the computation of only a single

weight value.

Thus 28\*28=784 threads were launched for a BPNN with

784 inputs and 533 hidden neurons in the feed-forward

phase from the input to the hidden layer.

Each thread has read access to a single input neuron, read

access to a single weight value and write access to a hidden

neuron, resulting in (n + 1) \*h write operations in the

hidden layer.

The number of threads was reduced to 533 threads for the

above mentioned use-case.

Each thread has read access to all input layer neurons, read

access to a column of weights concerning a specific hidden

neuron (edges from each input neuron to this particular

hidden neuron) and write access to the same single hidden

neuron.

Artificial neural network is used to recognize ten different handwritten digits. These digits from 0 to 9.In order to have a learning task that is reasonably workable, a great amount of preprocessing of the digit is carried out using conventional Artificial Intelligence (AI) techniques.