International Journal of Research in Advent Technology, Special Issue, March 2019
E-ISSN: 2321-9637
Available online at www.ijrat.org

Digits Recognition using Neural Networks

Harshit Gupta[1], Nishant Shrivatsava[2]
**Department of Computer Science and Engineering,*
Jaypee University Anupshahr
gharshit1237@gmail.com[1]

Abstract – Handwritten digits recognition being a challenging problem is area of research in recent years. There are many concern areas of this problem including identifying cheque numbers and recognizing number plates of vehicles. The present study design a multi - layer neural networks to identify hand written images from 0 to 9. The digits are identified and extracted using image processing techniques like image enhancement and object recognition. Recognized digit converted in to binary form act as the input to the network. The network is then trained using gradient descent and backpropagation training algorithm to identify the digits. The experimental results have shown that the proposed system can predicts the digits with 96.5% accuracy, once properly trained

Keywords – Neural Networks, Neurons, Gradient Descent, Back Propagation ,Forward Propagation

1. INTRODUCTION

A neural network is a series of algorithm that attempts to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. Neural networks can adapt to changing input so the network generates the best possible result without needing a redesign the output criteria. Neural networks was widely used in 80's and early 90's but diminished in late 90's. Recently it is widely used and one of the most famous algorithm.

In this study ,system is designed for predicting handwritten digits. Image should be of certain size and format. The image is then edited then features are extracted to work upon multi level neural network. This is already used in automatic processing of bank cheques , postal addresses, number plates of vehicles. A wide range of researches has been performed on the MNIST database to explore the strength and weakness of the best recommended methodology. The best approach till date gives the training accuracy of 99.81% by Convolution Neural Network for feature extraction and used RBF network model for prediction of the handwritten digits. Research was extended at Concordia University for predicting handwritten digits. 99.17% accuracy was found on data set by preprocessing of data by Mexican hat wavelet transformation technique. This accuracy is lesser from the earlier one where the data was not preprocessed and the architecture was same.

2. RELATED WORK

Calin et. al [9] proposed a neural computing method for recognizing handwritten digits. A framework was presented by them to classify handwritten digits, and the classification was performed using Convolutional Neural network. With original NIST dataset, it provided 96.74% accuracy, and with the images that are without background it provided 96.56% accuracy.
Dewi et. al [8] proposed a method for recognizing handwritten Latin characters using Freeman chain code Representation and Feed forward Neural network classifier. During the preprocessing stage, they used thinning method for obtaining the skeleton of a character, which also removes any redundant information, as well as it maintains the features of the image.

A wide range of researches has been performed on the MNIST database to explore the strength and weakness of the best recommended methodology. The best approach till date gives the training accuracy of 99.81% by Convolution Neural Network for feature extraction and used RBF network model for prediction of the handwritten digits [10]. Research was extended at Concordia University for predicting handwritten digits.
99.17% accuracy was found on data set by preprocessing of data by Mexican hat wavelet transformation technique. This accuracy is lesser from the earlier one where the data was not preprocessed and the architecture was same [11].

3. PROPOSED METHODOLOGY

Our neural network architecture consist of 3 layers. One input layer with 400 units, hidden layer with 25 units and output layer with 10 units each for one digits from 0 to 9 as shown in Fig 1. Layers are connected to each other using weights represented by Wij where i and j represents the two layers which are connected by weights. In this neural network sigmoid function (eq1) is used. This sigmoid function gives the value in the range [0,1].

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

3.1. Image Preprocessing

The digits are identified and extracted using image processing techniques like image enhancement and object recognition. All images needed to be converted into 20 pixels * 20 pixels format to be recognized. Recognized digit converted in to binary form act as the input to the network. The network architecture is similar to as defined in [2]. Back propagation training algorithm is used to update the weights and learn the patterns. The training algorithm concludes in two passes namely forward and reverse pass.

3.2. Input Layer

The 400 pixels extracted from each image is arranged as a single row in the input vector X. So the matrix X is of size 5000×400. It consists of the pixel values for the entire 5000 samples. This vector X is then used as the input to the input layer. Considering the above mentioned specifications, the input layer in the proposed 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.

3.3. Hidden Layer

As per the research conducted in the field of neural networks their is no such formulae for defining the number of hidden layers or numbers of units in hidden layers. Many structure is decided through assumption and then best one is picked up by the test of cross validation test graph and error graph. So after going through these process I have decided to choose 1 hidden layer with 25 units.

3.4. Output Layer

The targets for the entire 5000 sample dataset were arranged in a vector Y of size (5000×1). One unit will give the value 1 and the rest will be 0. Therefore 10 units for each digits from 1 to 0.

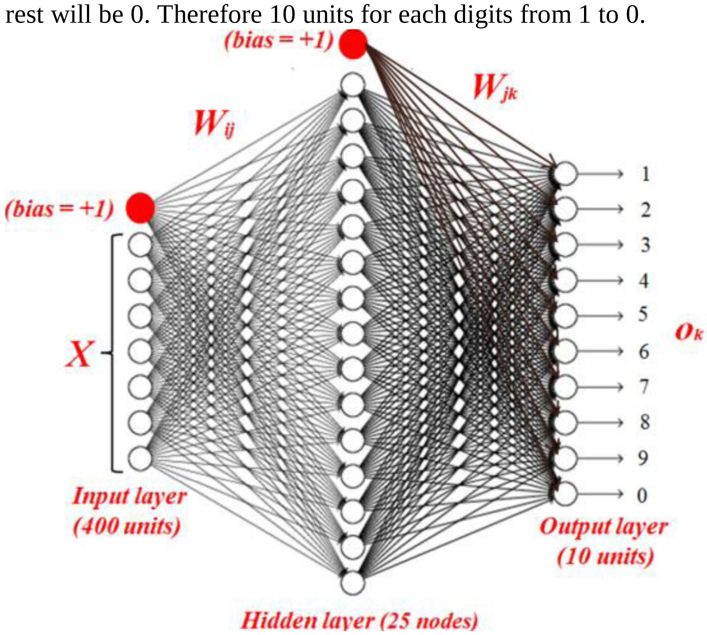

Fig 1:- The neural network architecture [2]

3.5. Forward Pass

The 400 pixels was provided to the input layer which is further multiplied with the weights connecting input and hidden layer which is represented by

$$x_j = W_{ij}x_i \tag{2}$$

This net input is provided to the sigmoid function *eq(1)* and the output is the value of the hidden layer

$$O_j = g(x_j) \tag{3}$$

The values of hidden layer is further multiplied with the weights connecting hidden and output layer which is represented by

$$x_k = W_{jk}O_j \tag{4}$$

This net input is provided to the sigmoid function equation(1) and the output is the value of the output layer

$$O_k = g(x_k) \tag{5}$$

3.6. Reverse Pass

Cost function is given by this equation

$$Cost(h\Theta(x), y) = -y\, log\big(h\Theta(x)\big) - (1-y) log\Theta\big(1 - h\Theta(x)\big) \tag{6}$$

If the output from the system is same as the real output then cost will be 0 otherwise the cost will be ∞. Therefore to reduce this cost function we can use gradient descent but it will be more complex and time taking so we use Backpropagation algorithm.

The gradient value for both output and hidden layers were calculated for updating the weights

$$\delta_k = O_k(1 - O_k)(O_k - Y_k) \tag{7}$$

$$\delta_j = O_j\big(1 - O_j\big) \sum_{k=1}^{k} \delta_k W_{jk} \tag{8}$$

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

$$\Delta W_{jk} = -\eta \delta_k O_j \tag{9}$$

$$\Delta W_{ij} = -\eta \delta_j O_i \tag{10}$$

$$W_{jk_{new}} = W_{jk_{old}} + \Delta W_{jk} \tag{11}$$

$$W_{ij_{new}} = W_{ij_{old}} + \Delta W_{ij} \tag{12}$$

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

Since Backpropagation has some bugs therefore we can use gradient checking to monitor it.

3.7. Initializing Weights

We can initialize all our weights as 0 but it will help in logistic regression not here. When we back propagate all our nodes will update to all same value repeatedly. Instead of this we can randomly initialize our weights .

3.8. Error Analysis

Start with a simple algorithm and then test with the cross validation data. We can plot the learning curve to decide if our model is high variance or high bias. So we can take the appropriate move and make the changes in out algorithm.

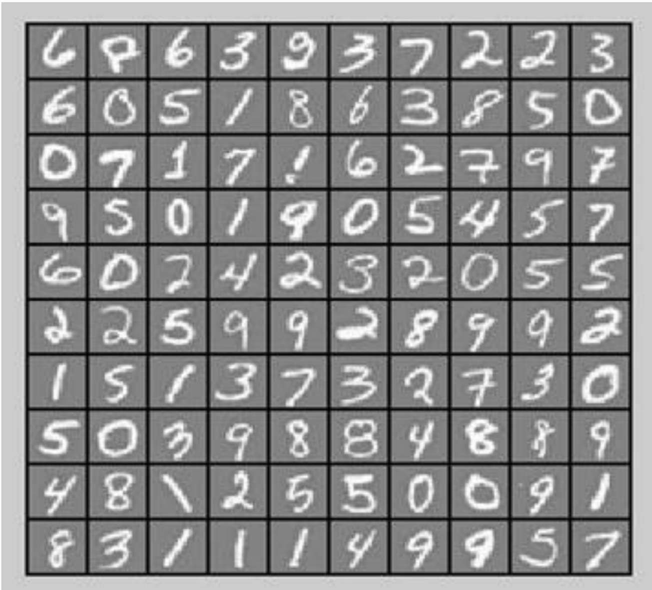

Fig 2:-The data set images

4. DATA SET

Each training images of digits are grayscale (fig 2) and of size 20 pixels * 20 pixels. Each pixel is represented by a floating point number indicating the grayscale intensity at that location. The 20 by 20 grid of pixels is "unrolled" into a vector of size 400*1. Each of these training examples becomes a single row in our data matrix X. This gives us a 5000 by 400 matrix X where every row is a training example for a handwritten digit image. The second part of the training set is a 5000-dimensional vector Y that contains labels for the training set images.

5. RESULT

The performance of the above neural network architecture was tested on 1000 data set of 20*20 gray scale images. It was tested in octave under windows 8 environment on Intel i3 processor with 4GB RAM. The accuracy is the criteria for assessment of the performance. The accuracy rate is given by

$$\frac{Number\ of\ \ samples\ recognized\ corretly}{Total\ Number\ to\ test\ samples} * 100$$

Training and testing was continued at the step size of 50 iteration until we got the optimal accuracy as shown in Fig 3.

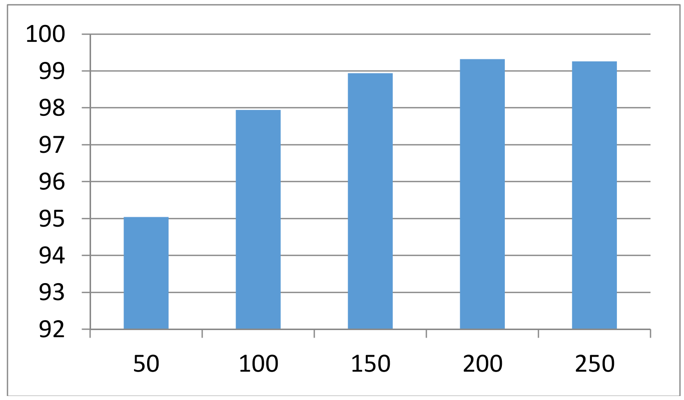

Fig 3: No of iteration vs. accuracy

The Accuracy is above 96.5% and highest is 99.32% on the test data in 200 iterations as shown in Fig 4. The Accuracy rate is 0.49% less than the best accuracy rate offered till date which is 99.81%.

Table 1 : Different approach with accuracy%

S. No.	Methodology	Accuracy%
1.	Y.Le Cun [6]	96.6%
2.	Cruces et. al [7]	99%
3.	Saleh et. al[5]	95%
4.	MNIST Researcher [10]	99.81%
5.	Mexican hat wavelet [11]	99.17%

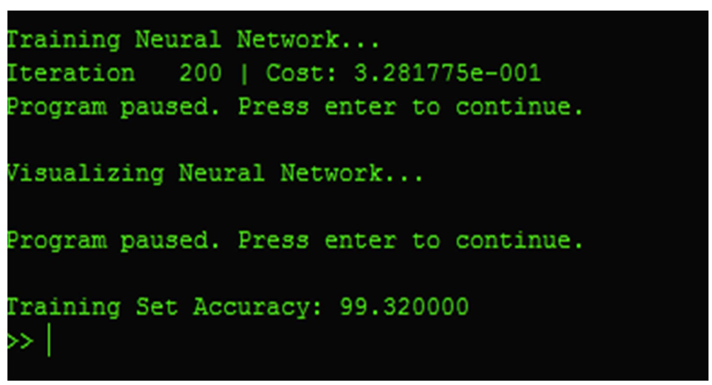

Fig 4: The Output Screen

6. CONCLUSION

A Multilayer Neural Network was implemented to address the handwritten digit recognition problem. The proposed neural network was trained and tested on a dataset. The neural network architecture with hidden neurons 25 and maximum number of iterations 200 were found to provide the optimal parameters to the problem. The proposed system was proved efficient with an overall accuracy of 99.32%. The accuracy is not the highest but it is 7% faster than other methodology.

REFERENCES

[1]Dr.Kusum Gupta" Neural network Handwritten Digit Recognition using various Neural Network Approaches" Sakshica1, IJARCCE Vol. 4, Issue 2, February 2015

[2]Saeed AL-Mansoori "Digit Recognition using Neural Networks" Int. Journal of Engineering Research and Applications ISSN : 2248-9622, Vol. 5, Issue 5, (Part -3), pp.46-51 2015

[3]M. S. Akhtar and H. A. Qureshi, "Handwritten digit recognition through wavelet decomposition and wavelet packet decomposition," Digital Information Management (ICDIM), Eighth International Conference on, Islamabad, 2013, pp. 143-148 2013

[4]https://www.researchgate.net/publication/221522748

[5]Saleh Ali K. Al-Omari, Putra Sumari, Sadik A. Al-Taweel and Anas J.A. Husain Journal of Computer Science 5 (6): 427-434, 2009 ISSN 1549-3636 © 2009 Science Publications

[6]Y.Le Cuu, B.Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard and L.D. Jackel," Handwritten Digit Recognition with a Back-Propagation Network," AT&T Bell Laboratories, Holmdel, N.J.07733 1990.

[7]Daniel Cruces Alvarez, Fernando Martin Rodriguez, Xulio Fernandez Hermida,"Printed and Handwritten Digits Recognition Using Neural Networks", E.T.S.I.T. Ciudad Universitaria S/N.36200 Vigo.SPAIN 1998.

[8]Dewi Nasien, Siti S. Yuhaniz, Habibollah Haron,"Recognition of Isolated Handwritten Latin Characters Using One Continuous Route of Freeman Chain Code Representation and Feed forward Neural Network Classifier", World Academy of Science, Engineering and Technology 67(2), 2010

[9]Calin Enachescu, Cristian-Dumitru Miron,"Handwritten Digits Recognition Using Neural Computing", Scientific Bulletin of the Petru Maior University of Tirgu Mures, Vol.6(XXIII), 2009, ISSN 1841-9267.

[10]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.

[11]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.