



Handwritten Digit Recognition

Jyoti Shinde, Chaitali Rajput, Prof. Mrunal Shidore, Prof. Milind Rane

Department of Electronics Engineering,
B.R.A.C.TS Vishkarma Institute of Technology Pune, Maharashtra

ABSTRACT

The main objective of this paper is to recognize and predict handwritten digits from 0 to 9 where data set of 5000 examples of MNIST was given as input. As we know as every person has different style of writing digits humans can recognize easily but for computers it is comparatively a difficult task so here we have used neural network approach where in the machine will learn on itself by gaining experiences and the accuracy will increase based upon the experience it gains. The dataset was trained using feed forward neural network algorithm. The overall system accuracy obtained was 95.7%

Keywords: Accuracy, Feed-forward, Handwritten Digit Recognition, Multilayer Neural Network

1. Introduction

As we know human vision is one of the wonders of the world. We carry a super computer in our head where we can sense the world or can see the world through evolution of over hundreds and millions of neurons hence handwritten digit recognition isn't an easy task for machines. Every human has its own way of writing numbers therefore for a machine it becomes difficult to predict digits. Similarly they also have their own tendency of recognizing a digit for example let's take '9' one can remember it as it has a loop and a stroke thus here in case of machine we provide training set. Here we have used Neural Network approach where we have given several hand written digit examples where the system can learn from those examples and predict exactly what the digit is. If we increase the number of training examples accuracy

will be further increased. This proposed system can be used in banking to process cheques and in post offices to recognize address etc

2. Dataset

Here 5000 training examples is feeded to the neural network. Where each training example is a 20 pixel by 20 pixel gray scale image of the digit. Each pixel is represented using a floating point number which indicates the gray scale intensity at that location. The 20 by 20 grid of pixels is unrolled into a 400-dimensional vector. 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.

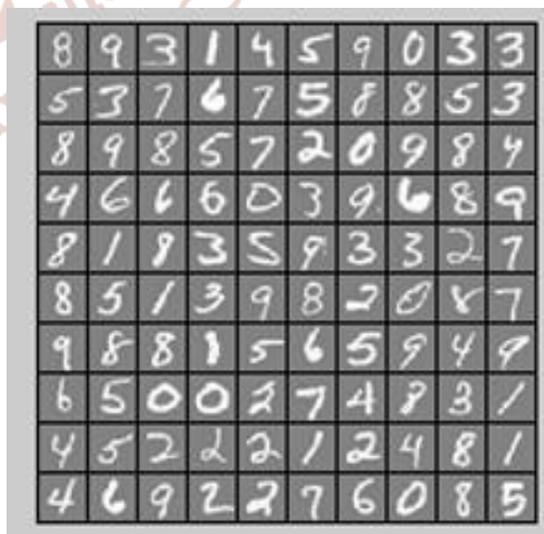


Figure1: Sample hand written digits

3. Proposed Methodology

3.1 Perceptron

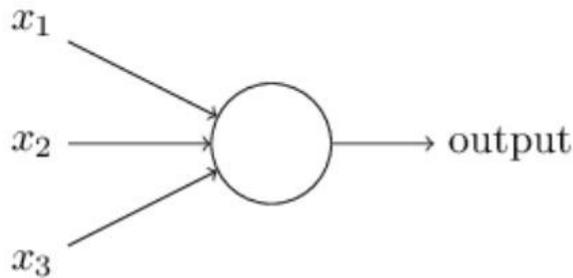


Figure2: Perceptron

A Perceptron takes several binary inputs x_1, x_2, x_3, \dots . And produces a single binary output. They are associated with weights w_1, w_2 , which express the importance of respective inputs to the outputs. The output of the Perceptron is 0 or 1 depending on the value of weighted sum whether it is greater than or less than some threshold value.

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

But the problem with perceptron is that a small change in weights or biases of any perceptron in the network can cause a large change in the output of that perceptron that can flip the output from 0 to 1 or 1 to 0. That flip causes large change in the behavior of the network. To overcome this problem sigmoid neuron was introduced.

3.2 Sigmoid Neuron

Sigmoid neurons are similar to perceptrons, but modified so that small changes in their weights and biases cause only a small change in their output and their values lie between 0 to 1.

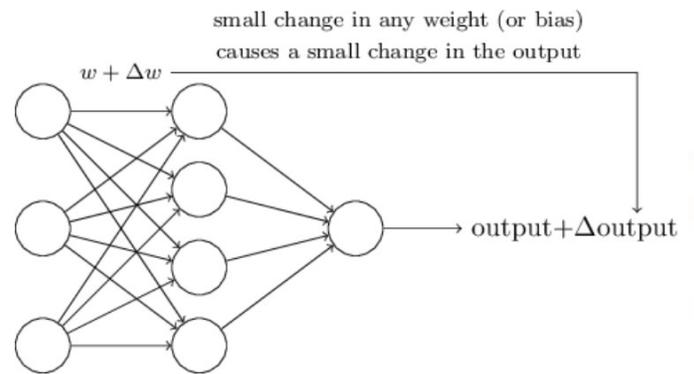


Figure3: Neural Network

3.3 Feed Forward Neural Network

The leftmost layer in the network is called input layer, the input layer of the neuron contains the value of the input pixel and the neurons within the layer are called input neurons. The rightmost or output layer contains the output neurons. The output layer of the network contains 10 neurons. If the first neurons fires, i.e. has an output = 1 then that will indicate that the network thinks digit is 0. Similarly if the second neuron fires then that will indicate that the neuron thinks the digit is 1 and so on. The middle layer is called hidden layer. Here the hidden layer contains 25 neurons. Since the neurons in this layer are neither inputs nor outputs. User can select the number of hidden layers depending upon the applications. The design of the input and output layers in the network is often straight-forward. For example suppose we are trying to determine whether a hand written image depicts a '9' or not. Here the image taken is a 20 by 20 gray scale image then we would have 400 input neurons with the intensities scaled appropriately between 0 and 1. The output layer will contain just a single neuron with output values of less than 0.5 indicating input image is not a '9', and a value greater than 0.5 indicating input image is a '9'.

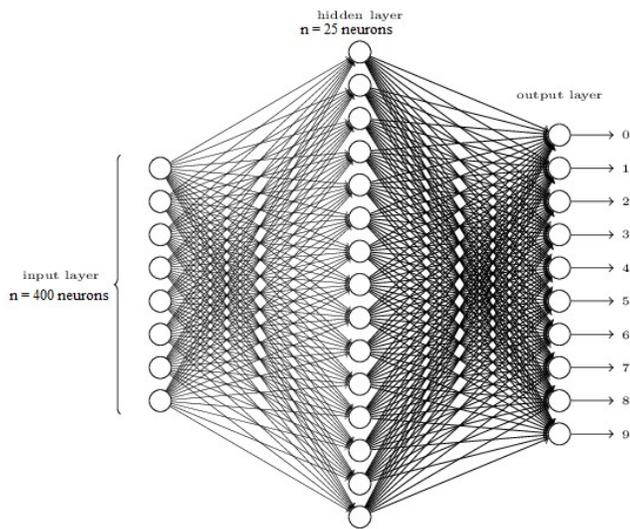


Figure 4: Feed forward neural network

Here in this neural network the output from one layer is used as input to the next layer such networks are called feed forward neural networks. Information is always feed forward never fed back. Each of those perceptron is making a decision by weighing up the results from the first layer of decision making. In this way a Perceptron in second layer can make a decision at a more complex and more abstract level than perceptrons in the first layer. And even more complex decisions can be made by the perceptron in the third layer. In this way a many layer network of perceptrons can engage in sophisticated decision making

3.4 Cost Function

Here we are using a algorithm to find weights and biases so that the output from the network approximates $y(x)$ for all training inputs x . Such algorithm is called cost function.

$$C(w, b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2.$$

Here 'w' denotes the collection of all weights in the network, 'b' denotes all the biases, 'n' is the total no of training inputs, 'a' is the vector of outputs from the network when x is input and the sum is over all training inputs x . Furthermore the cost $C(w, b)$ becomes small, i.e. $C(w, b) = 0$ precisely when $y(x)$ is approximately equal to the output a for all training inputs x . If $C(w, b) = 0$ it implies that our neural network has done a good job. We want to find a set of

weights and biases which makes the cost as small as possible this is done by the algorithm known as gradient descent.

3.5 Gradient descent

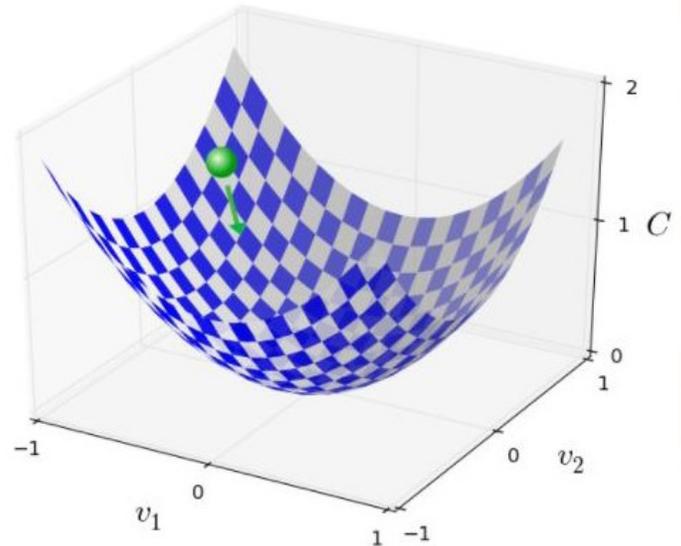


Figure 5: Minimization of cost function

Here $C(v)$ is a function of just two variables v_1 and v_2 . We are interested in finding where C achieves its global minimum. We could compute derivatives and then try using them to find places where C is minimum.

4. Neural Network based Handwritten Digit Recognition system

In this paper a neural based offline handwriting recognition system is developed. In this method each character is resized into 20 x20 pixels and after that feed forward neural network is used to train the pixels.

4.1 Input image

The scanned image which is given to the neural network as input

4.2 Pre-processing

In pre-processing stage various operation are performed like on image like binarization, noise removing, and edge detection.

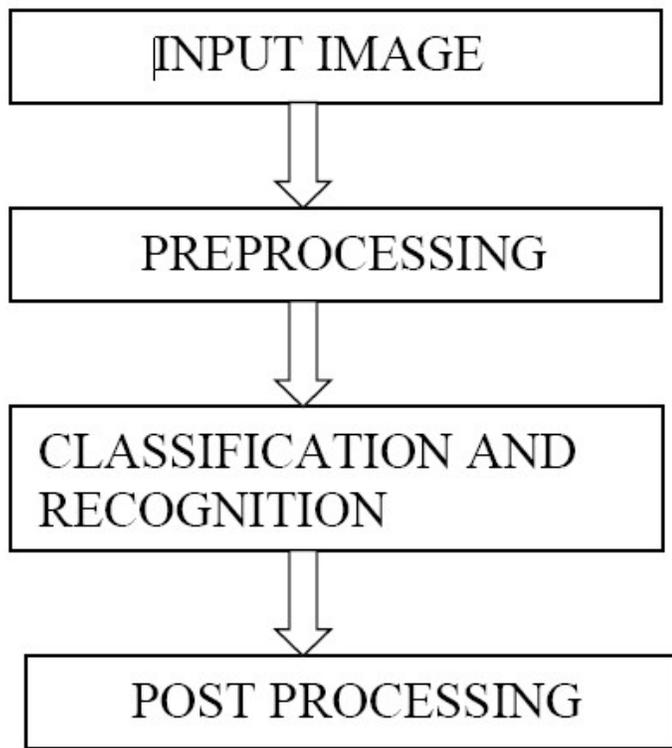


Figure 6: A Diagram of Recognition System

4.3 Classification and Recognition

This stage is the decision making stage of the recognition system. The classifier contains of one hidden layer is present. The hidden layers used log sigmoid activation function to train the data.

4.4 Post-processing

Post processing is last stage of recognition system. It prints the actual output after recognition.

5. Experimental Results

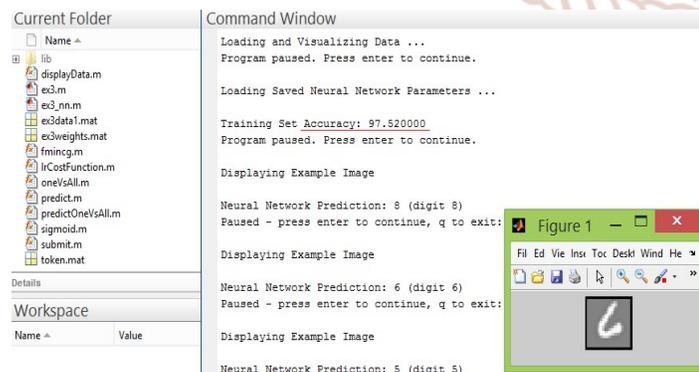


Figure7: Experimental Result

6. CONCLUSION

This review paper represent the technique used for recognize the hand written digits. This review paper also focuses on that in today’s world hand writing reorganization is very difficult but very important. There are many applications where we need hand writing recognition system like bank cheque and form documents.

7. FUTURE SCOPE

We can add fuzzification with Back propagation algorithm to improve the efficiency and correctness of the algorithm. This algorithm can be used to recognize multiple digits at a time and also to recognize characters.

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. 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.
3. 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.
4. V.N. Manjunath Aradhya, G. Hemantha Kumar and S. Nousath, “Robust Unconstrained Handwritten Digit Recognition using Radon Transform”, *IEEE-ICSCN*, 2007, 626-629.