Project 3

Handwritten Digit Recognition using Classification Algorithms

CSE 574

Introduction to Machine Learning

Anirudh Reddy Nalamada

UB No.: 5016-9240

UBIT Name: aniredn

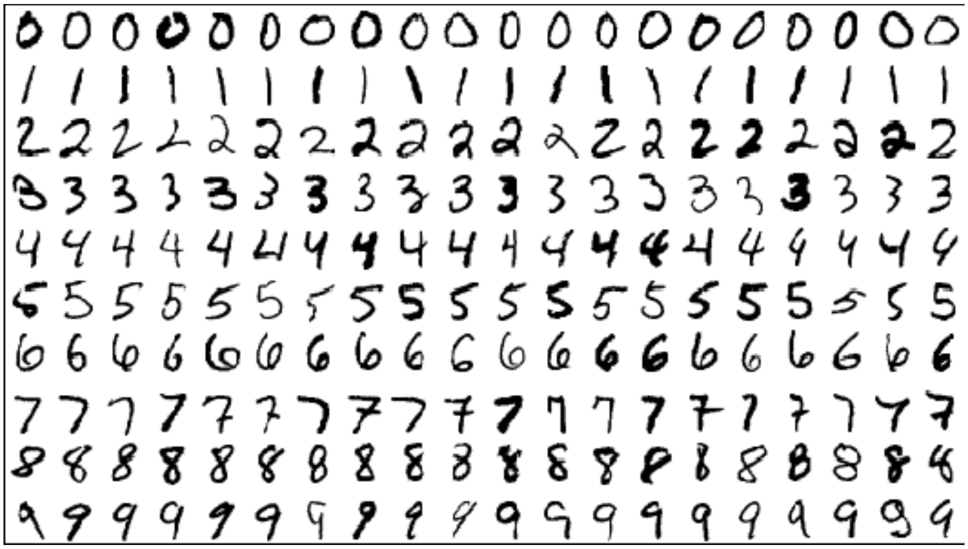
Introduction

The main aim of this project is to implement and evaluate classification algorithms to recognize handwritten digits from 0 to 9. In this project we use three methods to classify digits:

1. Logistic Regression,
2. Single Hidden Layer Neural Network
3. Convolutional Neural Network.

The data provided was the standard MNIST dataset, which consists of a large database of handwritten digits and is commonly used for image processing systems. The dataset consists of 60000 training images and 10000 testing images.

Below is an example of the images found in the MNIST dataset.



Formulae Used

Logistic Regression:

The multiclass regression model can be represented in the form:

$$p\left(\mathcal{C}_k|\mathbf{x}\right) = y_k\left(\mathbf{x}\right) = \frac{\exp\left(a_k\right)}{\sum_j \exp\left(a_j\right)}$$

The activation function is defined as

$$a_k = \mathbf{w}_k^\top \mathbf{x} + b_k.$$

The cross entropy error function is defined as

$$E\left(\mathbf{x}\right) = -\sum_{k=1}^{K} t_k \ln y_k$$

The gradient of the error function is

$$\nabla_{\mathbf{w}_j} E\left(\mathbf{x}\right) = \left(y_j - t_j\right)\mathbf{x}$$

The stochastic gradient descent formula is given by

$$\mathbf{w}_j^{t+1} = \mathbf{w}_j^{t} - \eta \nabla_{\mathbf{w}_j} E\left(\mathbf{x}\right)$$

Single Hidden Layer Neural Network:

The Feed Forward Propagation is as follows:

$$z_j = h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + b_j^{(1)}\right)$$

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + b_k^{(2)}$$

$$y_k = \frac{\exp\left(a_k\right)}{\sum_j \exp\left(a_j\right)}$$

where zj is the activation of the hidden layer and h(.) is the activation function which can be logistic sigmoid, tanh or ReLu.

The Back Propagation is done as follows:

$$\delta_k = y_k - t_k$$

$$\delta_j = h'\left(z_j\right) \sum_{k=1}^{K} w_{kj} \delta_k$$

The gradient of the error function is:

$$\frac{\partial E}{\partial w_{ji}^{(1)}} = \delta_j x_i, \qquad \frac{\partial E}{\partial w_{kj}^{(2)}} = \delta_k z_j$$

After finding the gradients we can use the stochastic gradient descent formula to update the weights.

$$\mathbf{w}_j^{t+1} = \mathbf{w}_j^t - \eta \nabla_{\mathbf{w}_j} E\left(\mathbf{x}\right)$$

Work Done

1. Logistic Regression Method:

In Logistic Regression Method, we find the weights and biases for which the model gives optimum results. The weight and bias terms were updated using gradient descent and mini batch gradient descent methods.

Misclassification rate for Mini batch Gradient Descent Method was approximately 10%. Misclassification rate for Gradient Descent Method was approximately 8%.
The gradient descent method yielded better results than the latter.

Eta	0.00002	0.0002	0.002
Approx. Misclassification Rate	10%	8%	12%

In the Gradient Descent Method, the number of iterations was fixed as 500. The value of eta was fixed as 0.0002. The values for weights and biases were initialized to a random real number.

Mini batch Stochastic Gradient Descent method was also implemented with the batch size as 100 samples. However this method resulted in a higher misclassification rate (~10%) for the same hyper-parameter values. Hence, more focus was applied to Gradient Descent Method.

2. Single Hidden Layer Neural Network:

In Single Hidden Layer Neural Network, we find the weights and biases for the hidden and output layer for which the model gives optimum results. In this approach the initial weights were initialized to a random real number and the bias values were initialized to 1.

The activation function values were first calculated using feed forward propagation. Then the gradient error was calculated using back propagation approach.
The weights were updated using the stochastic gradient descent method. The outer loop was iterated 200 times to achieve an optimum result.

3. Convolutional Neural Network:

To classify the data using Convolutional Neural Networks method, the Deep Learning Toolbox by Rasmus Berg Palm was used.

The input for the program was the images along with image size and the respective labels. The output of the program is the weights and biases for the output layer.

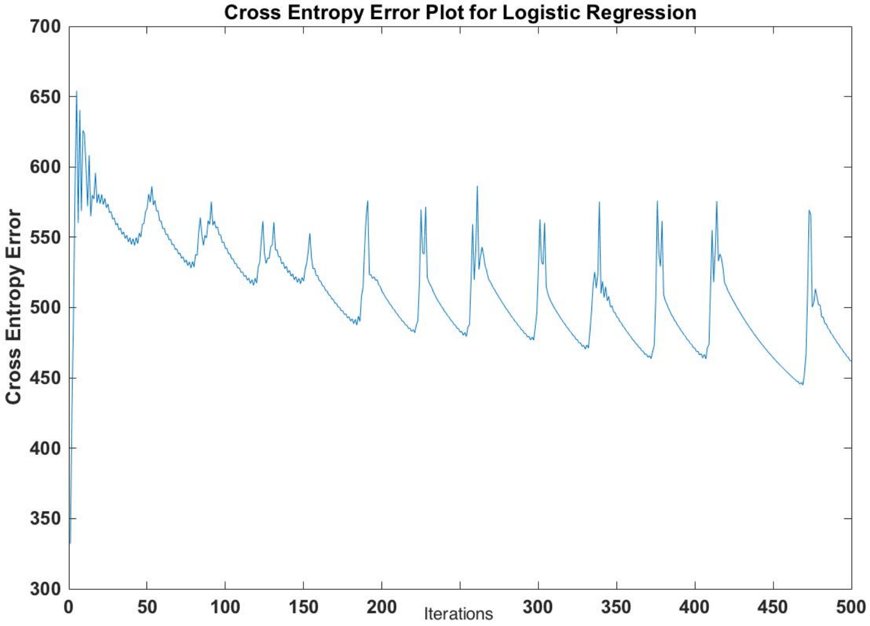
The number of output neurons is the number of output labels. In this case, it is 10, since we have 10 digits from 0 to 9
The convolution neural network was run with number of epochs fixed as 20 and 100.

Results and Discussion

1. Logistic Regression:

   The Misclassification rate for the training set is 7.5383%

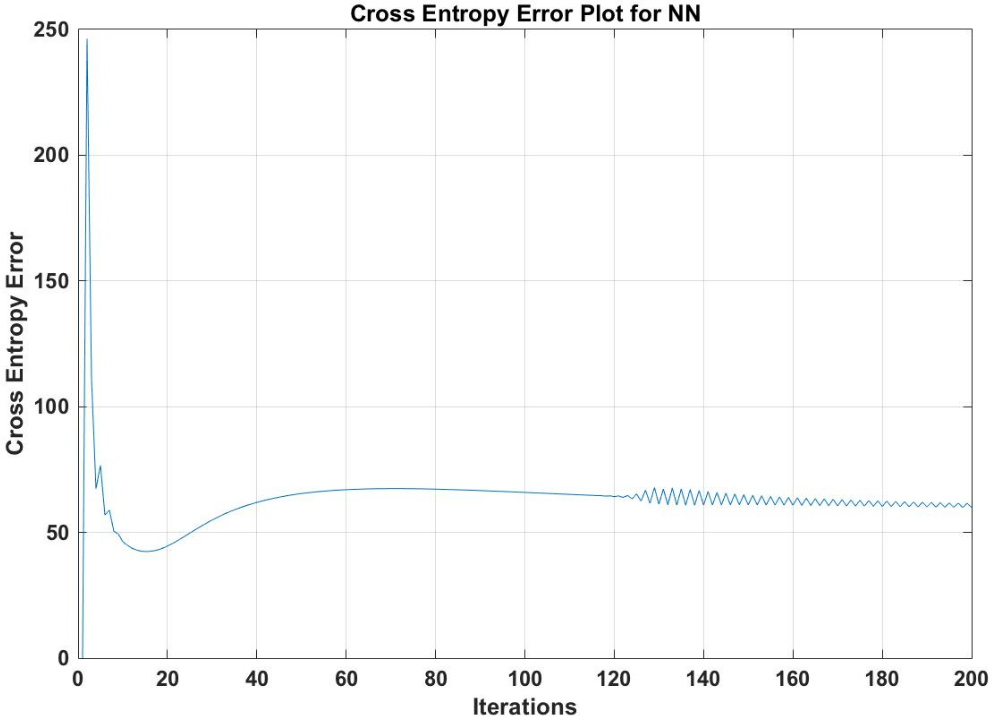
   The Misclassification rate for the testing set is 8.26%



2. Single Hidden Layer Neural Network:

   The Misclassification rate for the training set is 11.25%

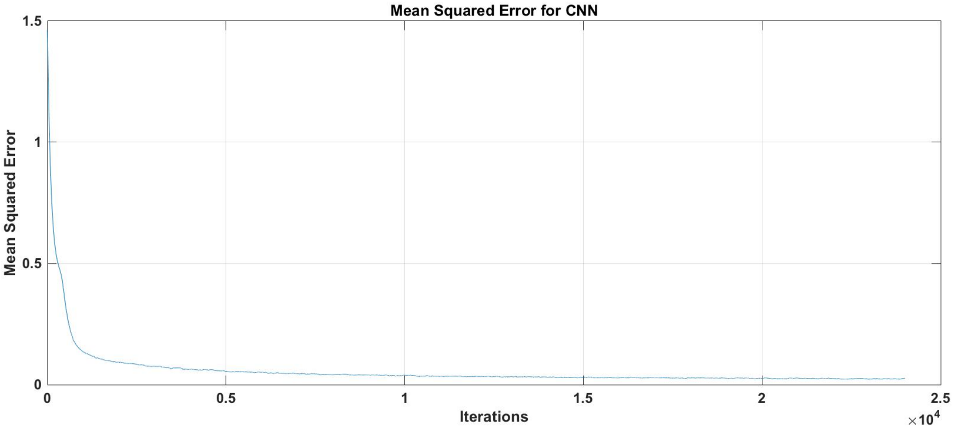
   The Misclassification rate for the testing set is 10.33%

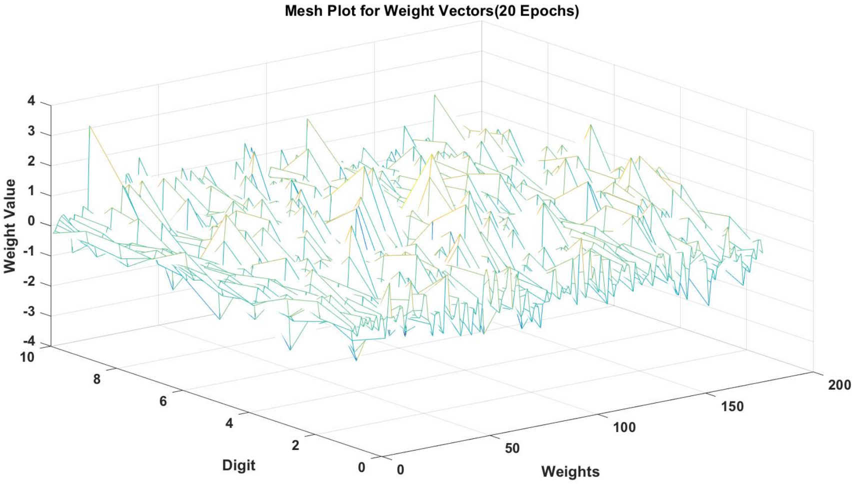


Although, the neural network should give a lower misclassification than the logistic regression, it is not true in this case. Tuning the value of eta and increasing the number of iterations can improve this result.

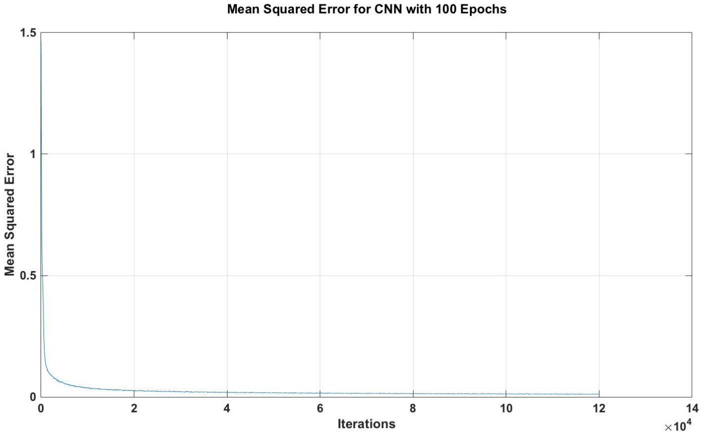
3. Convolutional Neural Network:

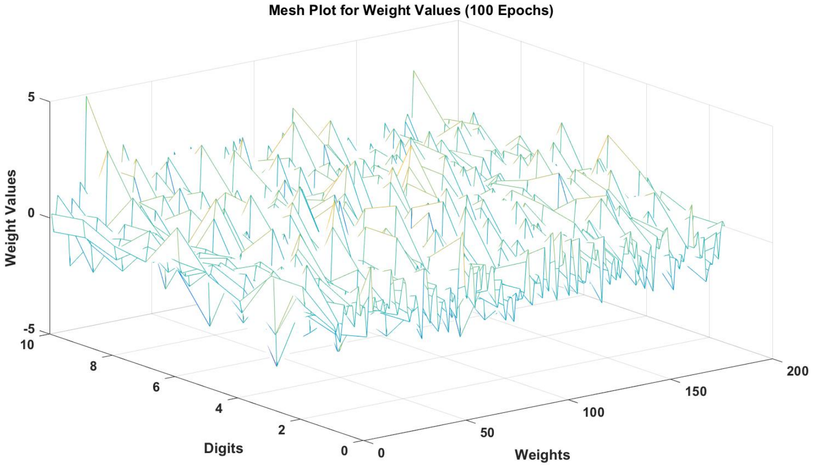
When 20 epochs were used, 178 samples were misclassified out of 10000 samples, which makes the misclassification rate as 1.78%.





When 100 epochs were used, 108 samples were misclassified out of 10000 samples, which makes the misclassification rate as 1.08%.





References:

- Dataset - http://yann.lecun.com/exdb/mnist/
- Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, *Proceedings of the IEEE, 86(11):2278-2324, November1998, \cite{lecun-98}.*
- Data Extraction- http://ufldl.stanford.edu/wiki/index.php/Using_the_MNIST_Dataset
- Deep Learning Toolbox - https://github.com/rasmusbergpalm/DeepLearnToolbox
- https://chrisjmccormick.wordpress.com/2015/01/10/understanding-the-deeplearntoolbox-cnn-example/