PROJECT REPORT 3

Identify Handwritten Digit Image Using Classification

CSE 574 Introduction to Machine Learning

Aditi Asthana
50169677
12/9/15

Introduction:

The goal of this project report is to describe the implementation of classification models which recognizes a 28*28 grayscale handwritten digit image and identify it as a digit among 0, 1, 2, ... , 9.

Objective:

There are three tasks to be completed in this project:

1. Implement logistic regression, train it on the MNIST digit images and tune hyperparameters.
2. Implement single hidden layer neural network, train it on the MNIST digit images and tune hyperparameters such as the number of units in the hidden layer.
3. Use a publicly available convolutional neural network package, train it on the MNIST digit images and tune hyperparameters.

Data Set:

For the training of our classifiers, we will use the MNIST dataset. The MNIST database is a large database of handwritten digits that is commonly used for training various image processing systems. The database is also widely used for training and testing in the field of machine learning. This data can be downloaded from http://yann.lecun.com/exdb/mnist/.

The database contains 60,000 training images and 10,000 testing images.

The original black and white (bi-level) images from MNIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. The images were centred in a 28x28 image by computing the centre of mass of the pixels, and translating the image so as to position this point at the centre of the 28x28 field.

This data set is divided into 3 parts:

a. Training Data Set (80% of 60000 = 48000)

b. Validation Data Set (20% of 60000 = 12000)

c. Test Data Set (10000)

Classification Models:

There are number of classification models that can be used to identify handwritten digits. Some of them that have been implemented in this system are:

1. Logistic Regression:

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution. 1-of-K coding scheme for the multiclass classification task can be represented as:

$$t = [t_1, t_2, \ldots, t_K]$$

The multiclass logistic regression model could be represented in the form:

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

where, the activation function a_k is given by $a_k = w^T_k x + b_k$

The cross-entropy error function for multiclass classification problem seeing a training sample x would be

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

where, $y_k = y_k(x)$.

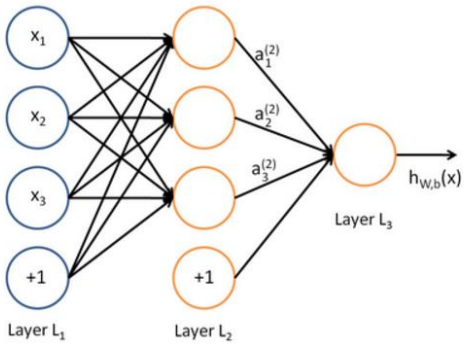
The gradient of the error function would be

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

Thus stochastic gradient descent is used to find the optimum of the error function and find the solution for w_j :

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

2. Single Layer Neural Network:



Neural network models in artificial intelligence are usually referred to as artificial neural networks (ANNs); these are essentially simple mathematical models defining a function $f : X \to Y$ or a distribution over X or both X and Y, but sometimes models are also intimately associated with a particular learning algorithm or learning rule.
In this project neural network having one hidden layer have been implemented. The input layers is denoted by x_i and the output is y_k. The feed forward propagation is as follows:

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

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

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

where z_j are the activation of the hidden layer and h (·) is the activation function for the hidden layer.
In this project logistic sigmoid function has been used as an activation function.

T
he use cross-entropy error function used is:

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

where $y_k = y_k(x)$.

The backpropagation is done as follows,

$$\delta_k = y_k - t_k$$

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

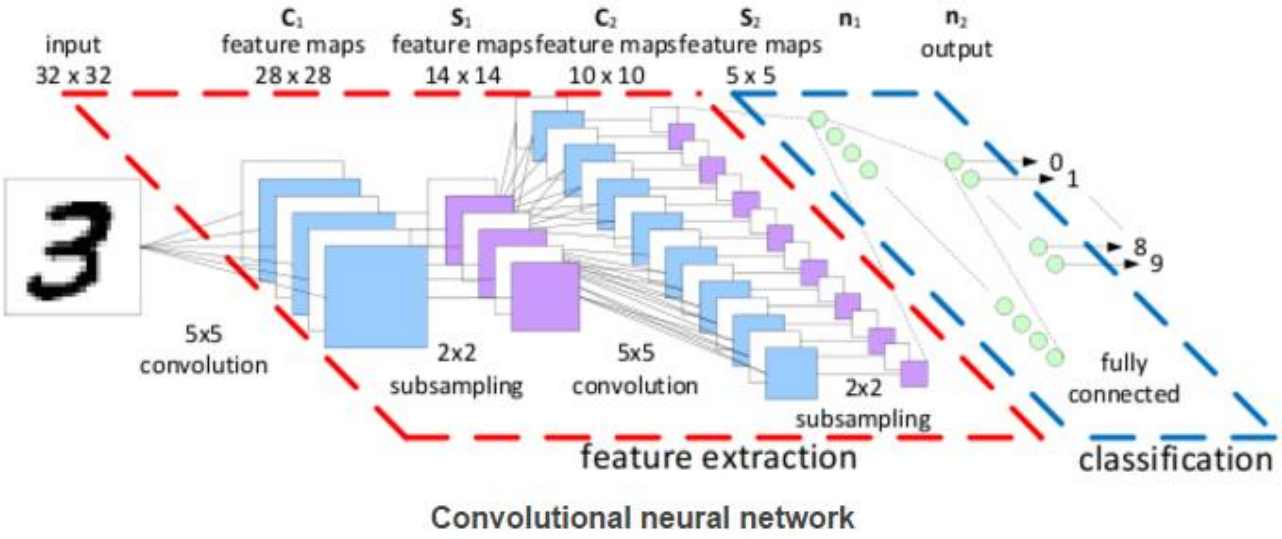
The gradient of the error function would be

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

Thus we can use stochastic gradient descent to train the neural network by updating the weights as:

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

3. Convolution Neural Network:



Convolutional neural network

A Convolutional Neural Network (CNN) is comprised of one or more convolutional layers (often with a subsampling step) and then followed by one or more fully connected layers as in a standard multilayer neural network. The architecture of a CNN is designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal). This is achieved with local connections and tied weights followed by some form of pooling which results in translation invariant features.

Implementation of Classification Models and choosing hyper-parameters:

1. Logistic Regression:

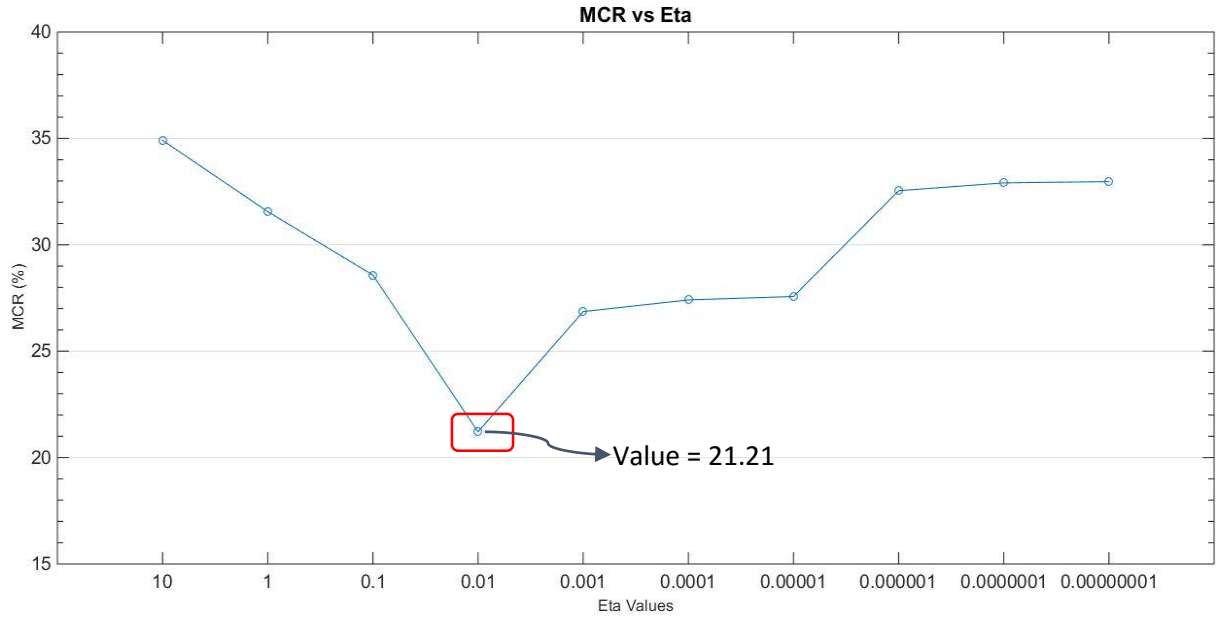
For getting optimized results, hyperparameters need to be selected so that misclassification rate (MCR – in percentage) for training set is minimum. In the case of Logistic Regression eta-η (learning rate) needs to hyper-tuned.

On varying the value of bias, we are getting the same value for MCR which are as follows:

BIAS VALUE	MCR
0.5	11.38
1	11.38
5	11.38
10	11.38
20	11.38

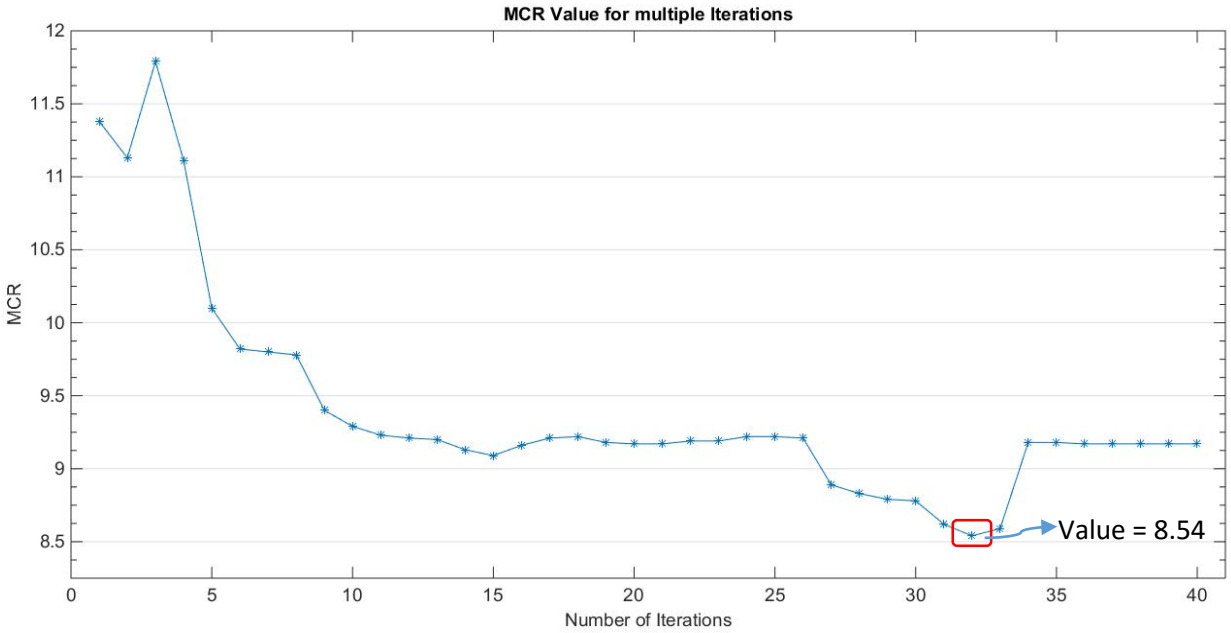
Thus in this model, the value of bias is static and have been set to 1.

The graphical representation of misclassification rate (MCR – in percentage) obtained for different values of η is as follows:



We can see from the above plot that at 0.01 value of eta the value of misclassification rate (MCR) is minimum (i.e. 21.21).

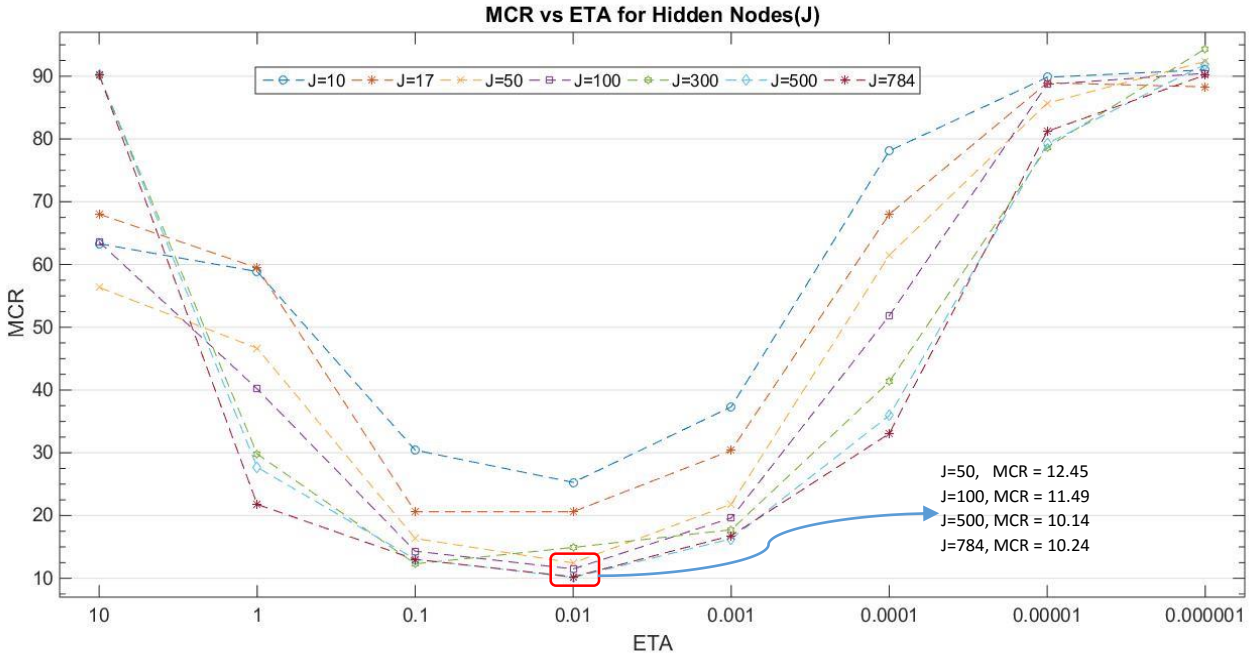
To optimize the error we need to train the data multiple times, thus doing so the below plot is obtained. Thus training the model with the same data multiple times having initial eta value as 0.01 and using adaptive learning technique we get the minimum MCR value to be 8.54 and the same can be determined from the below plot.



2. Single Layer Neural Network:

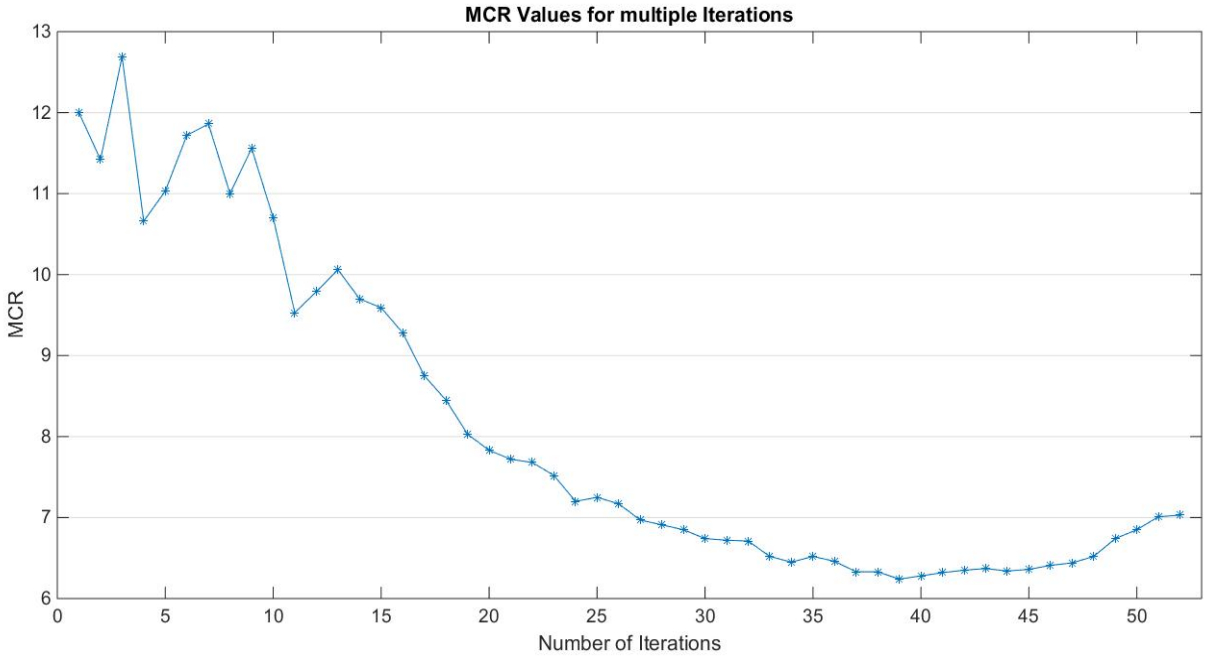
For getting optimized results, hyperparameters need to be selected so that misclassification rate (MCR – in percentage) for training set is minimum. In the case of Single Layer Neural Network eta-η (learning rate) and number of nodes in the hidden layer needs to hyper-tuned.

The graphical representation of misclassification rate (MCR – in percentage) obtained for different values of η and hidden nodes is shown in below plot.



We can see from the above plot that for J=500 and at eta = 0.01 the value of misclassification rate (MCR) is minimum (i.e 10.14).

To optimize the error we need to train the data multiple times, thus doing so the below plot is obtained.



Thus training the model with the same data multiple times having initial eta value as 0.01 and using adaptive learning technique we get the minimum MCR value to be 6.24.

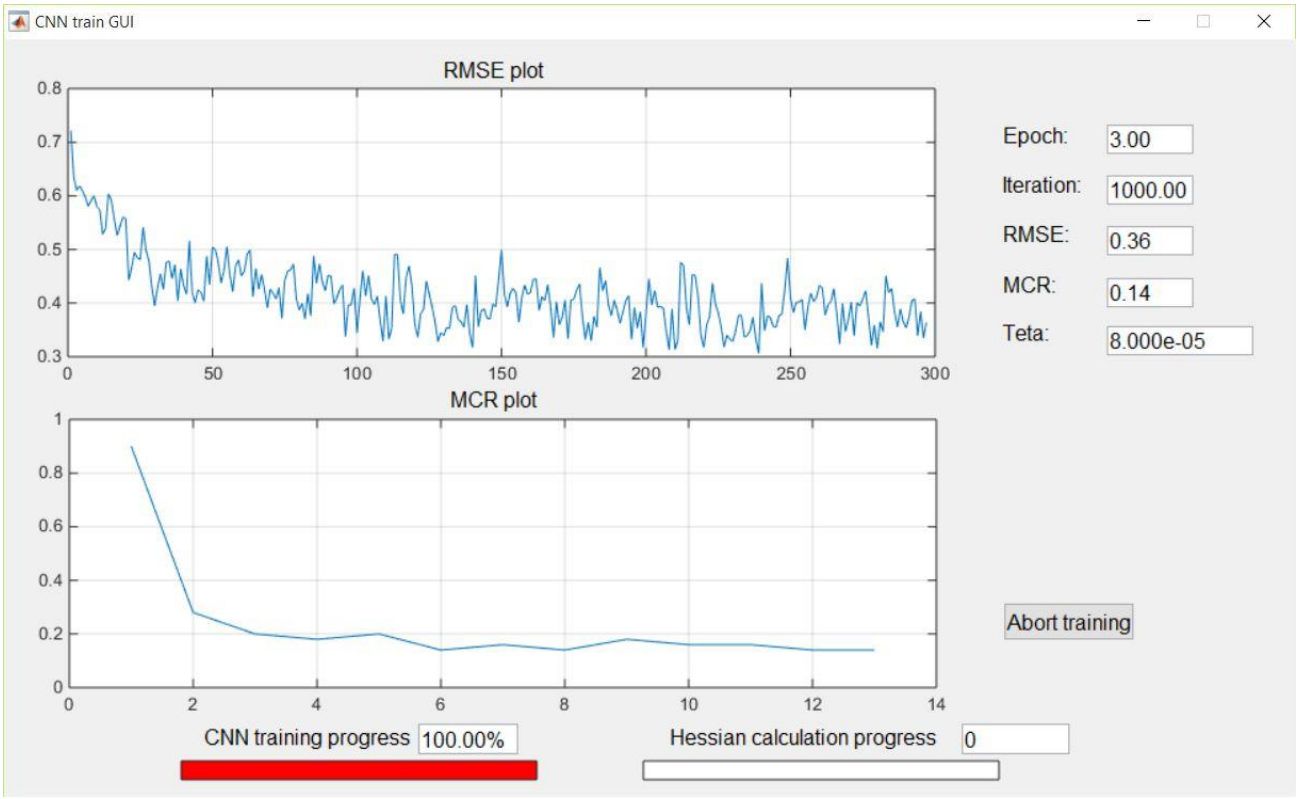
3. Convolution Neural Network:

In this model an online package have been implemented which can be found at the http://www.mathworks.com/matlabcentral/fileexchange/24291-cnn-convolutional-neural-network-class location.

Some of the details of the implementations are as follows:

- In this model a total of 8 layers have been used.
- Due to implementation specifics layers are always in pairs. First must be subsampling and last (before fully connected) is convolutional layer and thus first layer is dummy.
- There are 3 subsampling, 3 convolutional layers and 2 fully connected layers.
- The layers are implemented in the following order:
  Subsampling -> Convolutional -> Subsampling -> Convolutional -> Subsampling -> Convolutional -> Fully Connected -> Fully Connected
- In each convolutional layer a window size of 5*5 pixels is used for further computation in all the three convolutional layers.
- The subsampling rate for each subsampling layers is 1, 2 and 2 respectively.

On running the code with the above configurations, the output generated is as follows:



From the above figure, it is observed that the MCR in this model will be 0.14.

Performance Evaluation:

Different misclassification rate of the three implemented models are:

MODELS	MISCLASSIFICATION RATE
Logistic Regression	8.54
Single Layer Neural Network	6.24
Convolution Neural Network	0.14

Thus from the above data, we can conclude that convolution network performs better than the logistic and neural network.

References:

1. https://en.wikipedia.org/wiki/Logistic_regression
2. http://yann.lecun.com/exdb/mnist/
3. http://ufldl.stanford.edu/wiki/index.php/Neural_Networks
4. http://www.es.ele.tue.nl/~dshe/Education/Dmm2012
5. http://deeplearning.net/tutorial/lenet.html
6. http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/
7. http://www.mathworks.com/matlabcentral/fileexchange/24291-cnn-convolutional-neural-network-class