CSE 574 – Introduction to Machine Learning
Project 3: Classification
Project Report

Submitted by: **Maruthi Mohan Reddy Putha**
UBIT Name: **maruthim**
UBIT Number: **50208681**

1. Abstract

In this project, the aim is to implement various classification algorithms such as logistic regression, single layered neural networks and convolutional neural networks and evaluate their performance as compared to each other. The classification task is to recognize a 28 X 28 grayscale handwritten digit image and identify it as a digit among 0,1,2,3,4,5,6,7,8,9. We have been given MNIST data from a web forum in GITHUB and once model is trained on MNIST data and optimized and it has been tested on the USPS Data. For both training and testing 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 widely used for training and testing in the field of machine learning. The database contains 60000 training images and 10000 testing images. The original black and white images from MNIST were 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 centered in a 28x 28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28x28 field. We have used USPS data for testing the performance of various models we have implemented and USPS data contains a set of images of size 28x28 pixels and that data is used to test the implemented models.

We have implemented logistic regression on MNIST data and the weights are initially arbitrarily taken and then those weights are updated based on the gradient descent traversal along the cross entropy error function. And the optimized weights are taken and that is tested on the USPS data. Similarly, we have implemented the Single Layered Neural Networks on the MNIST data using some set value of nodes on the hidden layer initially. And implemented the CNN on the MNIST data set as well.

2. Logistic Regression

Logistic regression is a regression model where the dependent variable is categorical. The dependent variable can be cast into either yes or no category and if the variable can be categorized into multiple values then it is called as multinomial logistic regression. Here the dimensionality of the input variable is 784 and so the weights are assumed to convert the 784 dimensions into classes of 10 distinct numbers. And we have computed the weights in a step by step manner instead of batch manner. We have updated the weights for each and every input same of 1 x 784 dimension and after each iteration for completed data set the weights are updated and they are taken as initial values for the next iteration. And the weights that have been calculated based on the MNIST data set have been used to calculate the performance on the USPS data set that has been provided. As per our testing we have found out that the optimum values have been for **learning rate** of 0.01 and iteration number to be 15 on the outer level. And it has been shown that the performance is **92.114.** And testing with same parameters on the MNIST data the performance is **91.16.** And also with the same parameters, testing on the USPS data showed that the performance is **31.926.** This clearly follows the **no free-lunch theorem**, stating that one model training on a particular data set cannot be assumed to perform the same way on all the data sets.

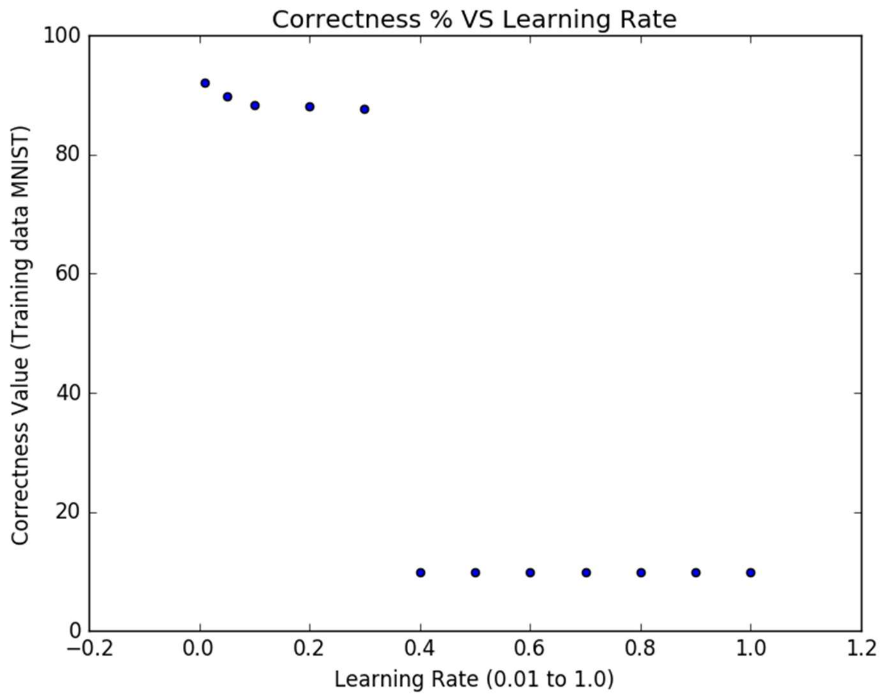

Figure 1 Plot showing the Correctness Percentage with respect to the Learning Rate for Logistic Regression

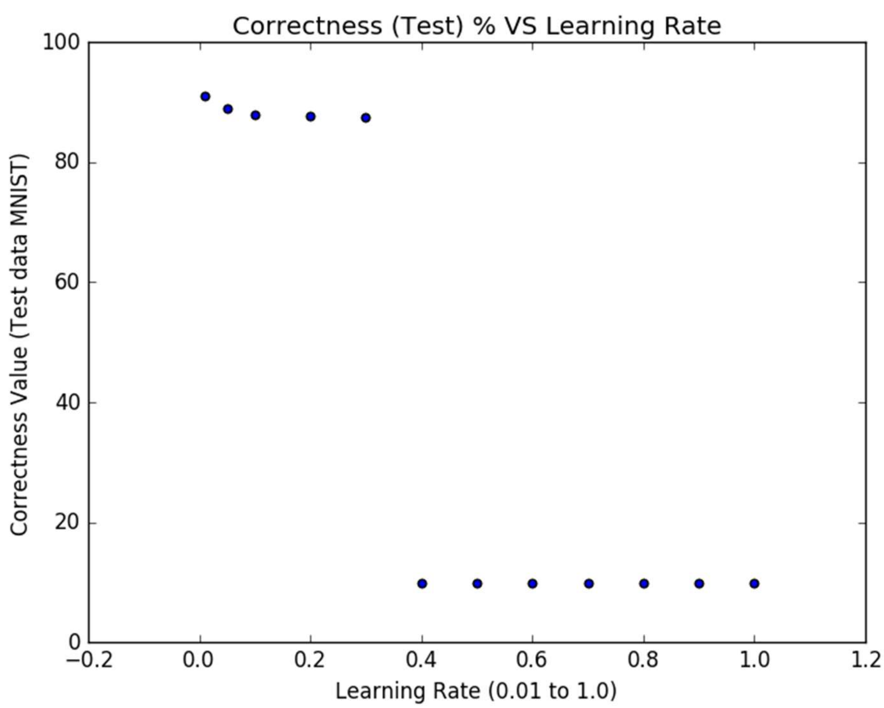

Figure 2 Plot showing the Correctness Percentage for Test Data with respect to the Learning Rate for Logistic Regression

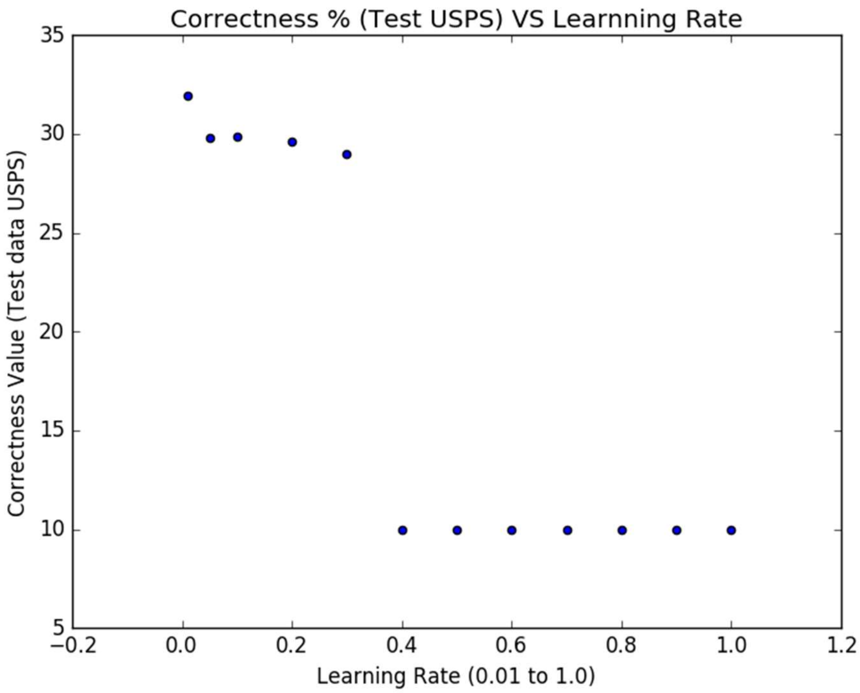

Figure 3 Plot showing Correctness Percentage with respect to the Learning Rate for Logistic Regression

3. Single Layered Neural Networks

By connecting multiple neurons, the true computing power of the neural networks comes, though even a single neuron can perform substantial level of computation. The most common structure of connecting neurons into a network is by layers. The simplest form of layered network is comprised of three main layers input layer, hidden layer and output layer. The shaded nodes on the left are in the so-called input layer. The input layer neurons are to only pass and distribute the inputs and perform no computation. Thus, the only true layer of neurons is the one on the right. Each of the inputs x_1, x_2, x_3, ..., x_N is connected to every artificial neuron in the hidden layer nodes k_1, k_2, …., k_N and that same layer is connected to the output layer through the connection weight. Since every value of outputs $ y_1, y_2, y_3, ..., y_N is calculated from the same set of input values, each output is varied based on the connection weights. Although the presented network is fully connected, the true biological neural network may not have all possible connections - the weight value of zero can be represented as ``no connection". Here the model has been trained on the MNIST data with 50000 training samples and the training performance has been found optimal for **learning rate** of 0.01 and the iteration number to be 5 at the outer most level. And the performance has been found to be **91.876.** And also with same parameters the testing on MNIST data set has shown the performance to be **91.48.** And also with the same parameters the testing on USPS data set showed the performance to be **36.242.** This result clearly shows that the model follows the same **no free-lunch theorem**, stating that a model that has been trained on one data set does not perform at the same level on all the data sets.

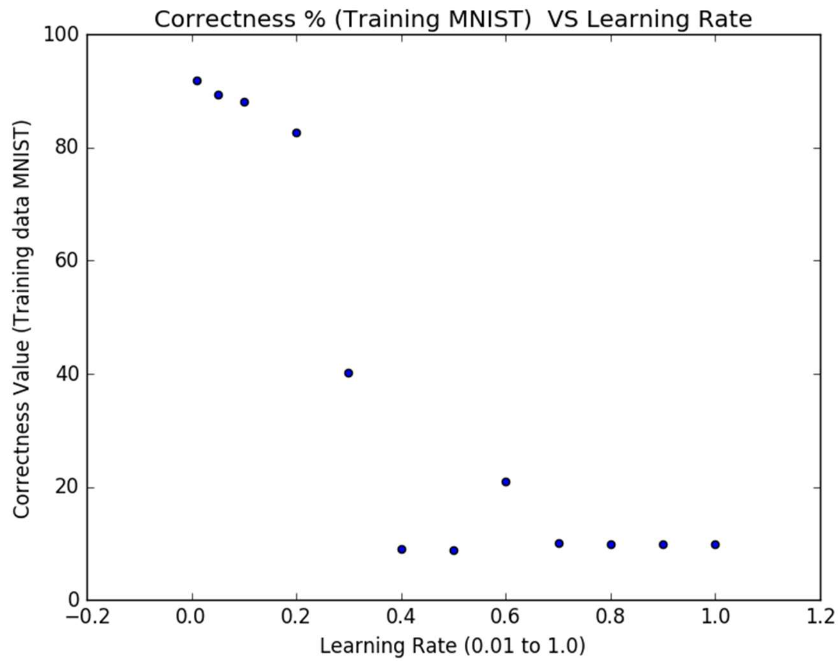

Figure 4 Plot showing the Correctness Percentage with respect to the Learning Rate for Neural Networks

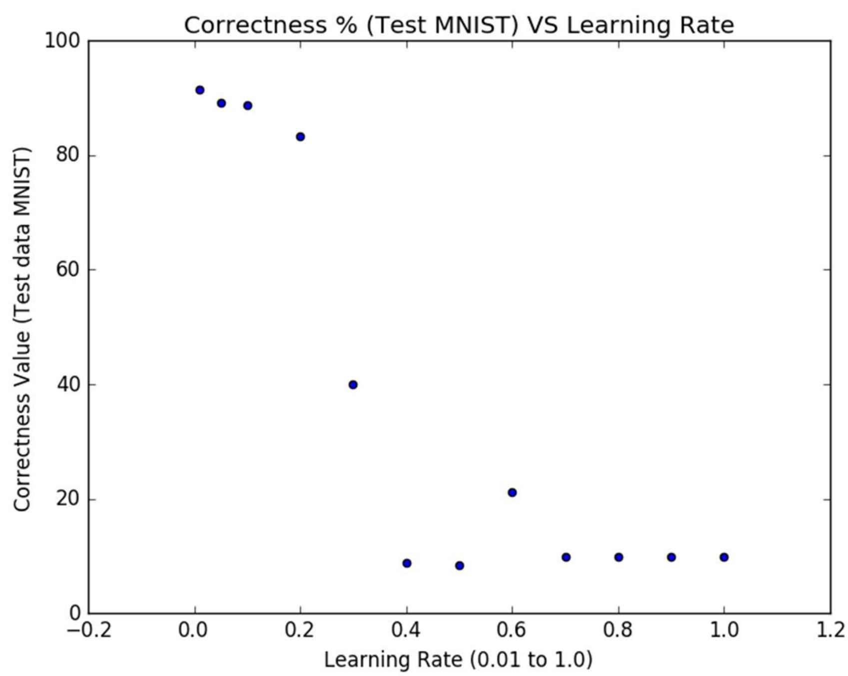

Figure 5 Plot showing Correctness percentage with respect to the Learning Rate for Neural Networks

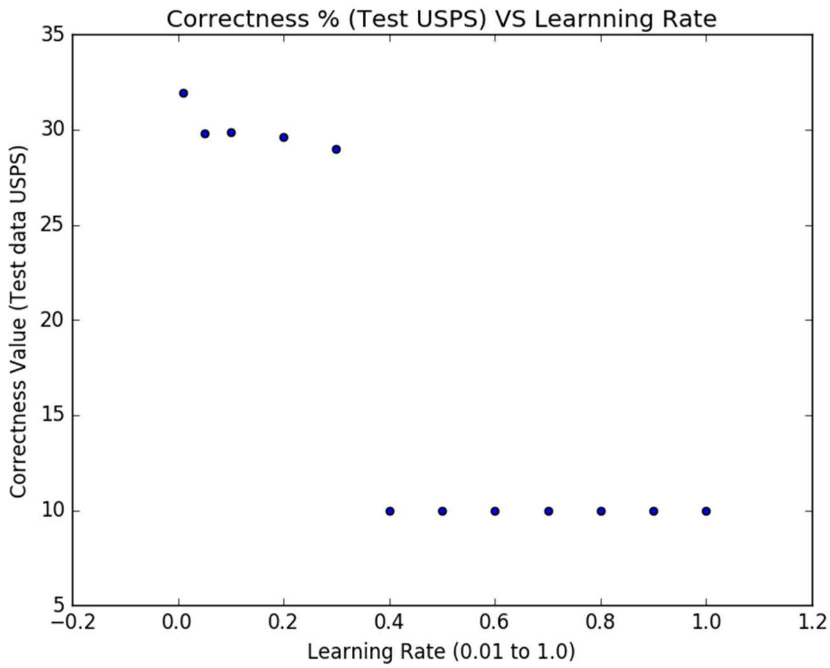

Figure 6 Plot showing the Correctness Percentage with respect to the Learning Rate for Neural Networks for USPS data

We have also computed the results with respect to the change in the number of nodes value. Below table shows the results that have been obtained from this experiment.

Number of Nodes in Hidden Layer	**Iteration Number at Outer level**	**Correctness Percentage for USPS data ($N_{Correct}$ /N_{Total})**	**Correctness Percentage for MNIST data ($N_{Correct}$ /N_{Total})**
100`	5	36.24	91.87
200	5	34.32	91.79
300	5	32.42	89.19

4. Convolutional Neural Networks (CNN)

Convolutional Neural Networks are very similar to ordinary Neural Networks from the previous chapter: they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. And they still have a loss function (e.g. SVM/Softmax) on the last (fully-connected) layer and all the tips/tricks we developed for learning regular Neural Networks still apply. We have taken two convolutional layers with pooling layers to each of that. And the performance of **0.9727** on the MNIST data set has been obtained using CNN. And the same value when computed with respect to the USPS testing data we have found that the testing accuracy to be **0.51**. This clearly states that the CNN model also follows the **no-free lunch theorem**, stating that a model trained on one data set does not work at the same level on other data sets.

5. Results

We have observed from results in general that increasing the learning rate would cause the efficiency to lower and also the increasing in the number of nodes in the hidden layer is causing the efficiency to lower as opposed to the natural behavior. And the results of the program are printed below.

```
Name       : Maruthi Mohan Reddy Putha
UBIT Number: 50208681
Hyper Parameters
Learning Rate: 0.01
-----------------------------------------------
Logistic Regression Evaluation on MNIST data :
-----------------------------------------------
Correctness Percentage for training data :  92.114
Correctness Percentage for validation data :  92.19
Correctness Percentage for testing data :  91.16
------------------------------------------------
         Evaluation on USPS Data
------------------------------------------------
Correctness Percentage for testing data :  31.9265963298
Hyper Parameters
Learning Rate: 0.01
Number of nodes in the Hidden Layer: 100
-----------------------------------------------------------
Single Layered Neural Networks Evaluation on MNIST data :
-----------------------------------------------------------
Correctness Percentage for training data :  92.032
Correctness Percentage for validation data :  92.45
Correctness Percentage for testing data :  92.01
------------------------------------------------
         Evaluation on USPS Data
------------------------------------------------
Correctness Percentage for testing data :  36.1768088404
------------------------------------------------
Convolutional Neural Networks
------------------------------------------------
Testing Accuracy for MNIST Data is 0.9738
Testing Accuracy for USPS Data is 0.5138
```