**PROJECT 3: CLASSIFICATION**

Monisha Balaji

University at Buffalo

mbalaji@buffalo.edu

**Abstract**

Machine learning and AI has been changing the course of technological advancement for over a few decades now. It has involved the birth of many new techniques and the evolution of many others. Although machine learning algorithms have been substituted in many applications, the problems can be broadly named as either a regression problem or a classification problem. Regression problems, as we know, deal with real values and aim to produce a continuous output. Classification, on the other hand, focuses on predicting the which output class the test data belongs to. In this project, the aim is to implement classifiers to help segregate the MNIST and USPS data sets. The goal is to implement a Deep Neural Network, a Convolutional Neural Network, Random Forest Classifier, Support Vector Machine and Logistic Regression Classifier. This report records the implementation and performance metrics of each of these classifiers.

1. **INTRODUCTION**

Problems that fall under the classification category include many important issues; for instance, predicting the presence of cancer or tumor, this issue, in its simplest form, takes up two classes: positive or negative. This is called a binary classification problem since it involves two classes. So we can use a binary classifier to solve this. Machine Learning provides us with algorithms to implement a Multiclass classifier. One such algorithm is Logistic Regression.

Logistic Regression is a classification algorithm that provides a hypothesis which produces values only between 0 and 1; that is it helps with the implementation of a binary classifier. Since we require the hypothesis to be bounded between 0 and 1, we make a slight modification to the hypothesis from linear regression algorithm. We know that, in Linear Regression, the hypothesis is given by,

**(x) = x**

Where is the parameter matrix

x is the input

In case of Logistic Regression, we perform a function g() to the hypothesis term from linear regression which is given as,

**(x) = )**

**) =**

Where z=x

The function g() that is applied to the hypothesis is called the sigmoid function. The sigmoid function which is also referred to as the logistic function helps bound the target values between 0 and 1. The sigmoid function ranges from 0 to 0.5 at the origin and slowly increases to 1. But in this project we deal with multiple output classes. So a few changes to the original logistic regression algorithm can help implement a Multiclass classifier.

**2.WORKING WITH MNIST AND USPS DATASETS**

MNIST dataset is the standard dataset that is used for the training of image processing systems. The dataset comprises of images of handwritten digits from 0 to 9. It consists of 60,000 training images and 10,000 test images.

USPS dataset also comprises of images of handwritten digits; about 2000 images for each digit. In this project, we use the MINIST dataset for training each of the classifiers and also for testing. USPS data is used only for testing the accuracy of the models. This helps evaluate the performance of the models based on its ability to generalize the data.

**3.IMPLEMENTATION OF VARIOUS CLASSIFIERS**

In this project, we have implemented 5 classifiers:

* Deep Neural Net
* Convolutional Neural Net
* Random Forest Classifier
* Support Vector Machine Classifier
* Logistic Regression with Softmax

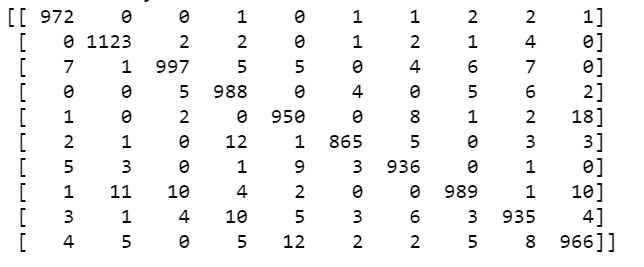
**3.1 DEEP NEURAL NETWORK**

A DNN is an artificial Neural network that consists of multiple hidden layers. Since the layers make the structure intense, it is referred to as a Deep Neural Network. Here, our input is a 28x28 pixels image. Each pixel is taken as an input to the neurons on the input layer. The processing takes place during the flow through the layers during which the images are processed into feature values. Feature extraction is the ideal step of image processing through NNs. These values help the model familiarize with the parameters to help better prediction of the test data.

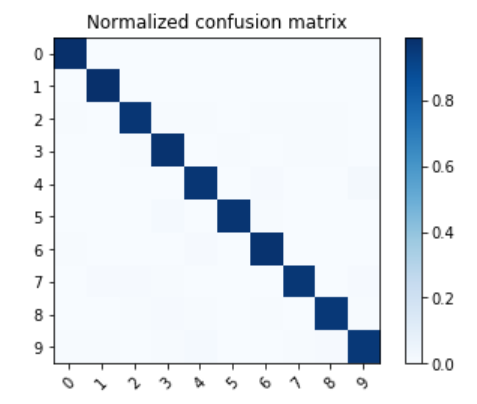
Using Keras, we implement a DNN with the ReLU activation function for the input and hidden layers and Softmax activation for the output layer.

A common measure of performance for classification problems in the confusion matrix. A confusion matrix, also known as the error matrix is a kind of contingency table that is formed from the actual output and predicted output. It is a two-dimensional table that describes the performance of the model. It helps us comprehend the statistics of the predictions.

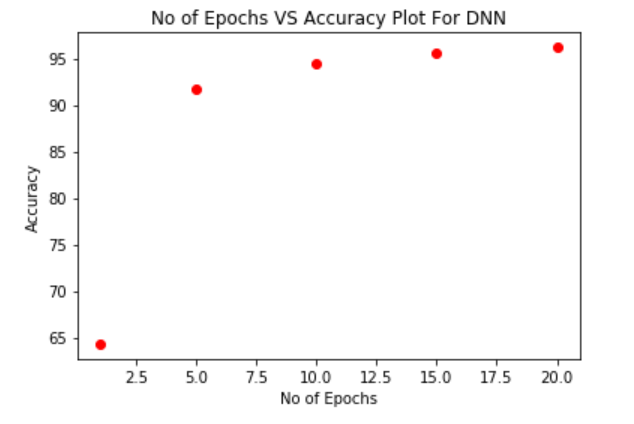
The confusion matrix produced by the DNN for the MNIST data is:



The graph for this confusion matrix is given by:



The correlation between accuracy and the no of epochs is given as:



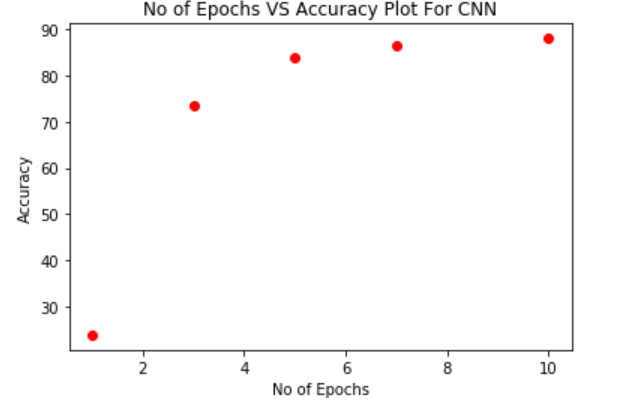
This shows the graph plotted for accuracy against the no of epochs. It shows that as the no of epochs increases, the accuracy also increases.

**3.2 CONVOLUTIONAL NEURAL NETWORK**

Convolutional NN are a variant of the standard NNs. Here again we include an input layer, an output layer and multiple hidden layers. The hidden layers are of different kinds: convolutional layers, pooling layers, fully connected layers and normalization layers. Convolutional layers perform the convolution function of the inputs. Since, in image processing, each image pixel is taken an input to the neurons, large images with lead to a very large number of neurons which in turn leads to process overhead. So convolutional layers focus on processing the inputs only for receptive fields. Next we have the pooling layers. The pooling layers focus on clustering data from all the neurons of the previous layer to produce an input to one neuron in the current layer. The clustering is done based on the pooling function applied. The fully connected layers are the same as the layers from a traditional fully connected NN where each neuron is connected to all the neurons from its previous layer.

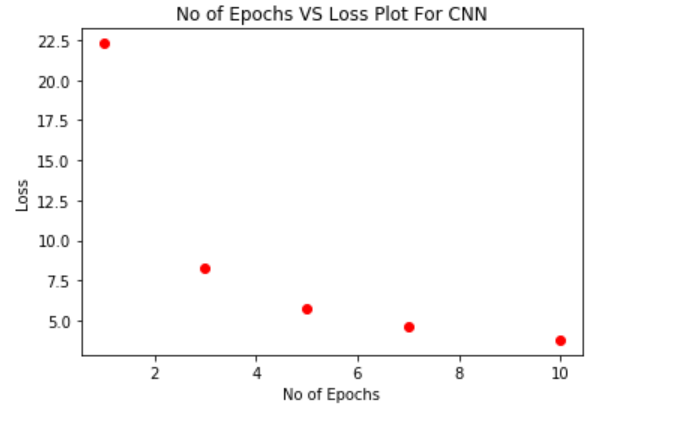
Here again we use the confusion matrix to help assess the performance.

The correlation between accuracy and the no of epochs is given as:



This shows the graph plotted for accuracy against the no of epochs. It shows that as the no of epochs increases, the accuracy also increases.

The correlation between loss and the no of epochs is given as:

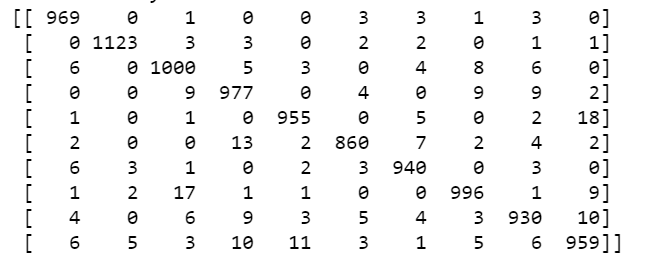
****

This shows the graph plotted for loss against the no of epochs. It shows that as the no of epochs increases, the loss decreases.

**3.3 RANDOM FOREST CLASSIFIER**

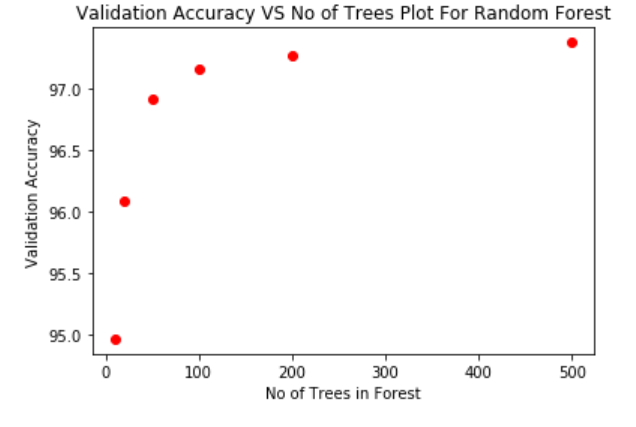
Random Forest classifiers involve a collection of decision trees that are fit on subsets of data and uses averaging to improve the accuracy of the model. This helps to prevent over-fitting. One of the best advantages of this algorithm is that it can be used for both classification and regression. Random forest algorithm is known to add randomness to the models while growing the tress. Rather than choosing the most important feature while splitting the nodes, it picks the best one. Another important aspect of random forest algorithm is that it makes it easy to measure the importance of various features. It concentrates on the distribution of the features across the tress nodes and how many of which produce less incorrect results.

The confusion matrix produced by the random forest is:



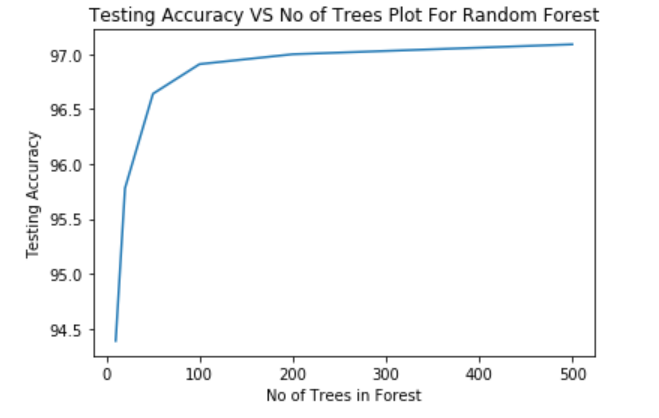
This table helps us understand the predictions how many classes were correctly predicted by the classifier. It gives a clear statistical depiction about the predictions.

The correlation between validation accuracy and the no of trees in the forest is given as:



This shows the graph plotted for accuracy against the no of trees in the forest. It shows that as the no of decision trees increase, the accuracy also increases.

The correlation between testing accuracy and the no of trees in the forest is given as:

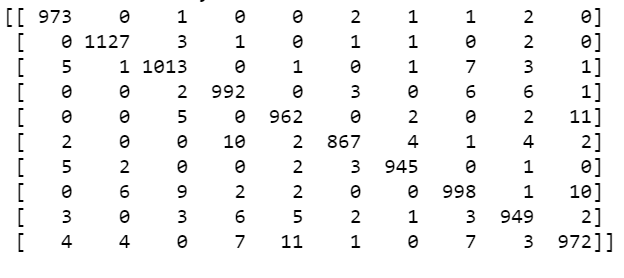
****

This shows the graph plotted for accuracy against the no of trees in the forest. Here again we see that as the no of decision trees increase, the accuracy also increases.

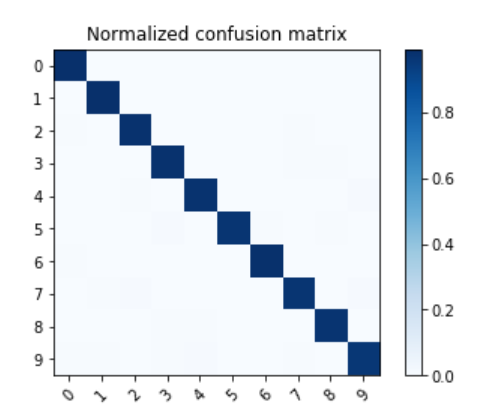
**3.4 SUPPORT VECTOR MACHINE**

SVM is a supervised learning model that helps solve classification problems. SVM is a non-probabilistic model i.e. it predicts such that the data belongs to one particular class amongst the available output classes. The data points are plotted in space in such a way that a clear distinctive gap that separates the different categories. So when new test data is given, it is plotted in the respective region in space.

The confusion matrix for SVM model:



It gives a clear statistical depiction about the predictions of the SVM model. The graph for this confusion matrix is given by:



**3.5 LOGISTIC REGRESSION WITH SOFTMAX**

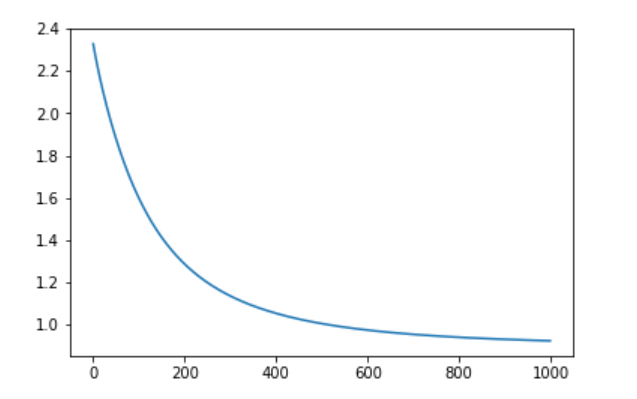
This is a variant of the original logistic regression approach. Instead of the sigmoid function we are going to use the softmax function to the hypothesis term. The hypothesis term is a sum of the product of weights and inputs from the different neurons along with the bias; softmax is applied to this value.

Where is the hypothesis term for the ith input,

is the summation over the feature values

Once the final hypothesis is generated from using softmax, we use optimization algorithms like stochastic gradient descent to reduce the cost function and thus update the weights through the number of iterations. Finally, the updated weights help achieve a better accuracy.

The correlation between the loss and the no of epochs:



**Loss**

**No of epochs**

This shows the graph plotted for loss against the no of epochs. It shows that as the no of epochs increases, the loss decreases.

**4. CONCLUSION**

This project has helped get a concise understanding of the four classifiers: Support Vector Machine, Random Forest, Deep Neural Net, Convolutional Neural Net and also Logistic Regression. We implemented all four of these classifiers. We trained the models on the MNIST data and tested them on both the MNIST and USPS datasets. We also constructed the confusion matrices for each of them which helped evaluate the performances of the models. For our project, we cannot declare one classifier to be the best. The performance of the Classifier depends on the type of data it is handling along with other factors. So a model that works well for one dataset might not necessarily work well for another. Thus, this project gives everyone a precise comprehension of the classification problem and these algorithms.

**REFERENCES**

[1] Logistic Regression- https://medium.com/@awjuliani/simple-softmax-in-python-tutorial-d6b4c4ed5c16

[2] Neural Network - https://keras.io/

[3] Random Forest - https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[4] Softmax Regression - https://www.kdnuggets.com/2016/07/softmax-regression-related-logistic-regression.html

[5] Keras- https://gist.github.com/RyanAkilos/3808c17f79e77c4117de35aa68447045