**CSE 574 Introduction to Machine Learning**

**Project 3 Report**

The task was to implement machine learning methods for the task of classification. We had to

implement an ensemble of four classifiers for a given task. Then the results of the individual classifiers were to be combined to make a final decision.

The classification task was that of recognizing a 28X28 grayscale handwritten digit image and

identify it as a digit among 0, 1, 2, ..., 9.

For the purpose of this task we had 2 different datasets:

1. MNIST Dataset

For both training and testing of our classifiers, we used the MNIST dataset. The MNIST database

is a large database of handwritten digits that is commonly used machine learning. The database contains 60,000 training images and 10,000 testing images.

2. USPS Dataset

We use USPS handwritten digit as another testing data for this project to test whether your models could be generalized to a new population of data. Each digit has 2000 samples available for testing, so a total of 20000 samples are available to test on.

The pipeline for processing and running the 4 classifiers is as follows:

1. Data preprocessing
2. Partitioning the data
3. Train the model parameter
4. Tune the hyperparameters
5. Test on MNIST dataset
6. Test on USPS dataset
7. Perform Majority Voting for 4 classifiers

**1 - Data Processing and Partition**

1. We have 2 datasets namely USPS and MNIST. We read the MNIST dataset and separate the target and the features and divide these two datasets into training set, testing set, and validation set.

The division is 50,000 training, 10,000 validation and 10,000 testing samples for MNIST.

We keep the whole USPS dataset as the testing set of 20,000 samples.

**2 –Logistic Regression Solution**

What is Logistic Regression?

Logistic Regression is basically a binary classifier which estimates or predicts 0 or 1 values. Since we want to do digit recognition for 0-9 digits, we use SoftMax as it is a multiclass classifier. SoftMax regression is a method in machine learning which allows for the classification of an input into discrete classes. Unlike the commonly used logistic regression, which can only perform binary classifications, softmax allows for classification into any number of possible classes. The softmax function is given by [1]:



1. **Features and Targets**

Here are the steps for calculating the weights:

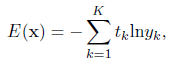
1. Initialize weights

We randomly initialize the weights as a vector of zeroes. We could use randomly initialize W by any value.

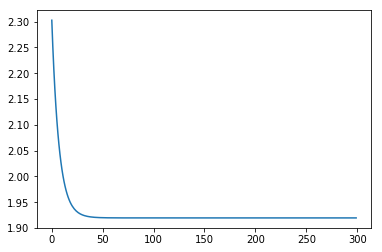
1. Update weights iteratively
2. Calculating error

When the model is trained using the training data and the model is functioning, we have no way of knowing whether the predicted value is correct. So, to test that out we use the error function. The error function or cost function is called as cross entropy.

The cost function is as follows:



We keep the number of epochs as 300 and learning rate as 0.05 and lambda as 1.



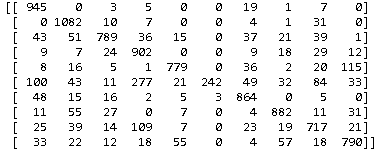
Logistic Regression

Loss vs Number of Epochs

MNIST Training accuracy is: 0.7832

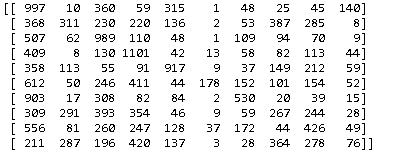
MNIST Testing accuracy is: 0.7992

The confusion matrix for MNIST is as follows:



USPS accuracy is: 0.289

The confusion matrix for USPS is as follows:



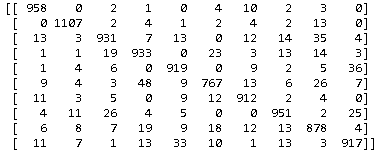
**3 –Neural Network Solution**

A neural network has input and output neurons, which are connected by weighted synapses. The weights affect how much of the forward propagation goes through the neural network. The weights can then be changed during the back propagation — this is the part where the neural network is now learning. This process of forward propagation and backward propagation is conducted iteratively on every piece of data in a training data set. [4] I have implemented neural network using the Keras library which is built on top of TensorFlow library. I have used the SoftMax activation function and have keep the number of epochs as 100 and the neural network consists of 2 layers with batch size of 128.

For MNIST

Accuracy: 0.927 and Loss: 0.253

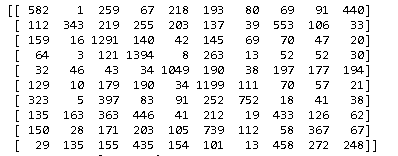
The confusion matrix for MNIST is as follows:



For USPS

Accuracy: 0.382 and Loss: 2.582

The confusion matrix for USPS is as follows:

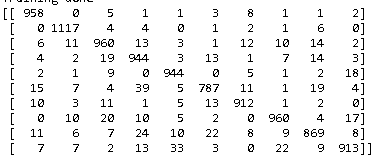


**4 –Support Vector Machine Solution**

Support Vector Machine is a high margin classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. [5] I have executed the SVM model for all the 3 settings which include kernel as linear and radial basis function and values of gamma as 0.05 and 1 with radial basis function.

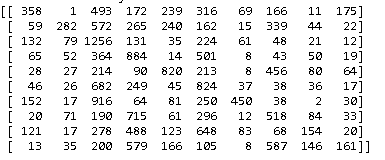
For MNIST Accuracy: 0.9364

The confusion matrix for MNIST is as follows:



For USPS Accuracy: 0.285

The confusion matrix for USPS is as follows:



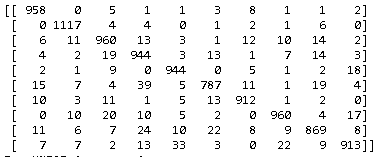
|  |  |  |  |
| --- | --- | --- | --- |
| SVM Settings | kernel=linear gamma=default | kernel=rbf gamma=0.05 | kernel=rbf  gamma=1 |
| MNIST Accuracy | 92.5 % | 93.6 % | 96.2 % |
| USPS Accuracy | 27.5 % | 28.5 % | 30.5 % |

**5 –Random Forest Solution**

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. [2] I have implemented random forest and optimized it to give best accuracy and have taken the number of estimators or the number of trees as 100 which gives best accuracy.

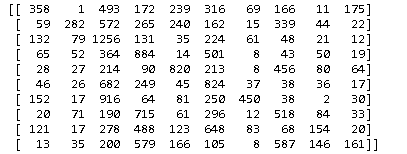
For MNIST Accuracy: 0.9364

The confusion matrix for MNIST is as follows:

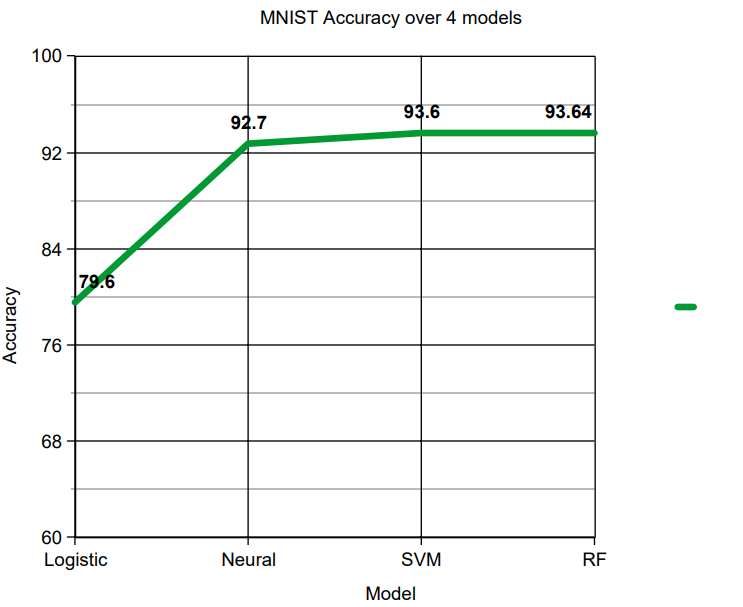


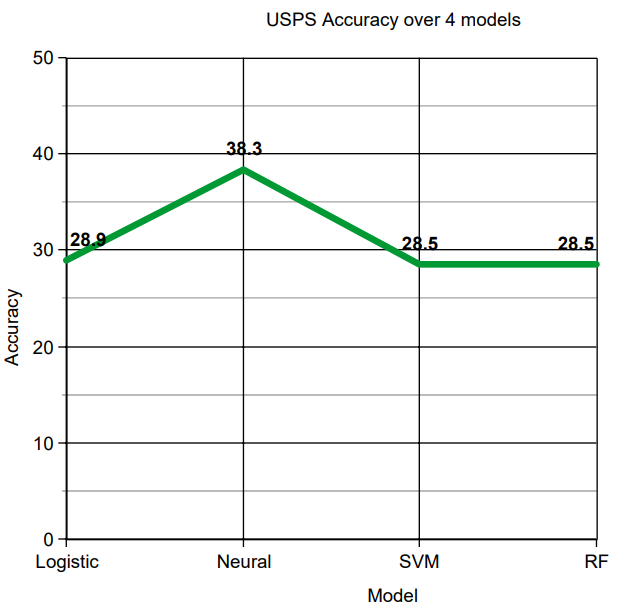
For USPS Accuracy: 0.285

The confusion matrix for USPS is as follows:



|  |  |  |
| --- | --- | --- |
|  | RF Number of Trees=10 | RF Number of Trees=100 |
| Accuracy | 91.6% | 93.64% |





**6 –Majority Voting**

I have implemented hard voting wherein I have checked the frequency of the predictions of Logistic regression, Neural Network, SVM and Random Forest for both MNIST and USPS datasets.

I have selected the predictions that occurs most out of these 4 classifiers and in case of a tie I consider any one of these at random. After running the ensemble classifier, we get a combined accuracy of-

After voting 94.45% MNIST

After voting 37.576% USPS

**7 – Questions to be answered / Observations**

1. We test the MNIST trained models on two different test sets: the test set from MNIST and a

test set from the USPS data set. Do your results support the “No Free Lunch” theorem?

We have trained the 4 models i.e Logistic Regression, Neural Network, Support Vector Machine and Random Forest on 50,000 samples of the MNIST dataset.

Now we test the model on the MNIST dataset which gives us good results.

Now when we test the same model on the USPS dataset, we do not get good results. This is because the model was trained on the MNIST dataset and even though both datasets are used for the same purpose of digit recognition we get varied results. This proves that even by using the same model/algorithm on 2 different datasets, we get different results. This supports the

“No Free Lunch” theorem.

2. Observe the confusion matrix of each classifier and describe the relative strengths/weaknesses of each classifier. Which classifier has the overall best performance?

Out of all the 4classifers we find that the Support Vector Machine works best or gives the best results on both the MNIST and USPS dataset. So, the best overall performance is given by Support Vector Machine.

3. Combine the results of the individual classifiers using a classifier combination method such as

majority voting. Is the overall combined performance better than that of any individual classifier?

I have implemented majority voting for the predictions of Logistic regression, Neural Network, SVM and Random Forest for both MNIST and USPS datasets. I have selected the predictions that occurs most out of these 4 classifiers and in case of a tie I consider any one of these at random.

The overall combined performance is better than that of logistic regression. The combined performance is as follows:

After voting 94.45% MNIST

After voting 37.576% USPS

References:

1. https://medium.com/@awjuliani/simple-softmax-in-python-tutorial-d6b4c4ed5c16
2. https://en.wikipedia.org/wiki/Random\_forest
3. https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d
4. https://medium.com/machinevision/overview-of-neural-networks-b86ce02ea3d1
5. https://en.wikipedia.org/wiki/Support\_vector\_machine