Project -2

1 Project Synopsis

Project 3 is to implement an ensemble of four classification methods namely Multiclass Logistic Regression, Neural Network, Random Forest and Support Vector Machines for the MNIST dataset and USPS dataset.

MNIST database of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size image. The USPS handwritten digit as another testing data for this project to test whether your models could be generalizing to a new population of data.

We will be training each model with the MNIST dataset and testing the model on USPS and MNIST dataset. Each prediction of the model will be stored and will be used to ensemble with MAX Voting technique. Each model prediction will support the "No Free Lunch Theorem".

Hyper parameters like learning rate, number of trees, kernel and Gamma in SVM, Neuron and number of layers in Neural Network is varied to get better accuracy.

2 Implementation

The data of MNIST and USPS is preprocessed. The test, train and validation is separated. The classification task will be that of recognizing a 28 x28 grayscale handwritten digit image and identify it as a digit among 0 to 9. The Multiclass Logistic Regression is implemented without any libraries, and for Multilayer Perceptron Neural Network, random forest and SVM I have used given code or from libraries.

All the Classification methods are combined together with the ensemble technique MAX Voting.

2.1 Data Preprocessing

2.1.1 MNIST

The MNIST data is loaded from the file with pickle.load. The loaded data is split into training data, testing data and validation data. Each training set (x_train, y_train) will have 50000

thousand image data and target value. Each testing set will have 10000 x 10000 size n – dimensional array. The training set will be used to train each of the model.

2.1.2 USPS

The USPS data is extracted from the file and is stored in a two lists USPSMat and USPSTar as one dimensional array each containing 19999 values. This will be used to test only.

The USPS data USPSMat is converted into an Array and USPSTar is converted to category of 10 classes with input function before passing the data to neural network model for evaluation.

2.2 CLASSIFICATION ALGORITHM

2.2.1 Multiclass Logistic Regression

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables

This logistic regression can be made multiclass by making the output layer to be a discrete probability distribution over the k classes. This is achieved by the softmax function.

The multiclass logistic regression is implemented with softmax algorithm without any external libraries with backpropagation. The gradient descent algorithm is used for backpropagation.

The methods followed to implement is as given in the Appendix 1.

- The target values is converted into 'one-hot vector representation' represented by 1-to10 scheme of class C_k . Class C_k is a 10-dim vector [t1,..,t10] $^{T.}$.

$$\boxed{\sum_{k=1}^{10} p(C_k) = \sum_{k=1}^{10} t_k = 1}$$

- The model for Multiclass regression is represented as below

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

- The activation function is given as $a_k = W^T x + b_k$
- The cross entropy error for multiclass classification problem

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

- The gradient of the error function would be

$$\nabla_{w_j} E(\mathrm{x}) = (y_j - t_j)\mathrm{x}$$

- We will use stochastic gradient descent to update the weights

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

- This finds the optimum of the error function and gets the solution for w_j.
- The hyper parameters which will be varied is the learning rate and the bias.
- Accuracy is found by

$$Acc = \frac{N_{correct}}{N}$$

2.2.2 Neural Network

2.2.2.1 Multilayer perceptron Neural Network

A multilayer perceptron (MLP) is a class of feedforward artificial neural network. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

- For multilayer perceptron, the input data is reshaped to a 784-pixel value input size.
- For output variables is converted into one-hot class encoding with keras.utils.to_categorical () or np_utils.to_categorical ()
- I have defined two hidden layers, each is passed with different number of neurons and activation function
- Each hidden layer will have 'n' number of nodes, which gives us 'n' biases.
- A softmax activation function is used on the output layer to turn the outputs into probability-like values and allow one class of the 10 to be selected as the model's output prediction Categorical_crossentropy is used as the loss function and the efficient ADAM gradient descent algorithm is used to learn the weights.
- The model is fit over 600 epochs with updates every 6000 images.
- The Hyper parameters are varied to get a better result.

2.2.2.2 Convoluted Neural Network

Convolutional Neural Networks are MLPs with a special structure. CNNs have repetitive blocks of neurons that are applied across space (for ex : images etc). For images, these blocks of neurons can be interpreted as 2D convolutional kernels, repeatedly applied over each patch of the image. At training time, the weights for these repeated blocks are 'shared', i.e. the weight gradients learned over various image patches are averaged.

- The input data is reshaped in the layers used for two-dimensional convolutions expect pixel values with the dimensions [pixels][width][height]. In the case of MNIST where the pixel values are gray scale, the pixel dimension is set to 1.

- The input data is normalized by dividing with 255 to make the value between 0 and 1.
- For output variables is converted into one-hot class encoding with keras.utils.to_categorical () or np_utils.to_categorical ().
- The first hidden(Conv2D) the layer has 32 feature maps, which with the size of 5×5 and a rectifier activation function.
- Next we define a pooling layer that takes the max called MaxPooling2D. It is configured with a pool size of 2×2.
- The next layer is a regularization layer using dropout it is configured to randomly exclude 20% of neurons in the layer in order to reduce overfitting.
- The Flatten layer allows the output to be processed by standard fully connected layers.
- Next a fully connected layer with 128 neurons and rectifier activation function.
- Finally, the output layer has 10 neurons for the 10 classes and a softmax activation function to output probability-like predictions for each class.
- The hyper parameters are tuned for better accuracy.

2.2.3 Random Forest

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.

The random forest is used from SKlearn library. It is tuned by varying the number of trees.

2.2.4 Support Vector Machines

Support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

Support Vector Machines(SVM) tool is used from the sklearn library.

2.3 ENSEMBLE

Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.

The ensemble technique used here is MAX Voting, multiple models are used to make predictions for each data point. The predictions by each model are considered as a 'vote'. The predictions which we get from the majority of the models are used as the final prediction.

I have taken the prediction from each classifier and find the mode of them which gives the max vote. We find the accuracy of them to get the final accuracy of the ensemble.

Accuracy is found by:

$$Acc = \frac{N_{correct}}{N}$$

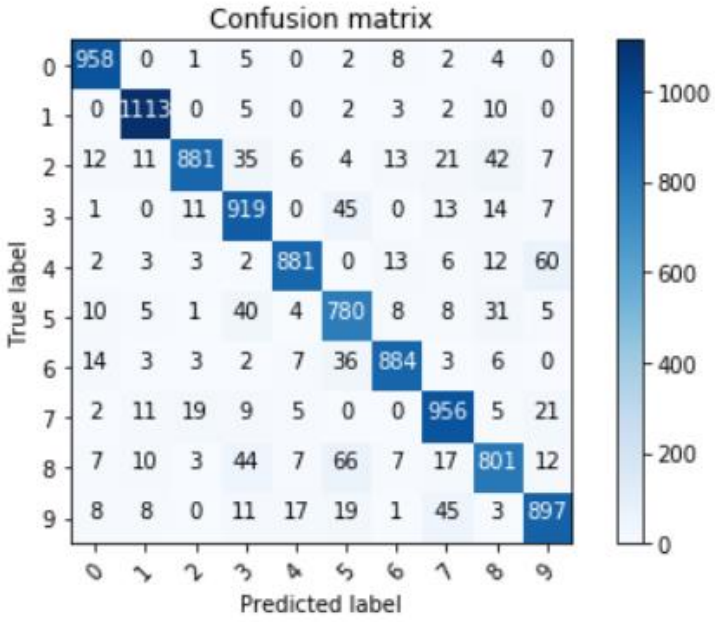
3 OBSERVATIONS

3.1 MULTICLASS LOGISTIC REGRESSION

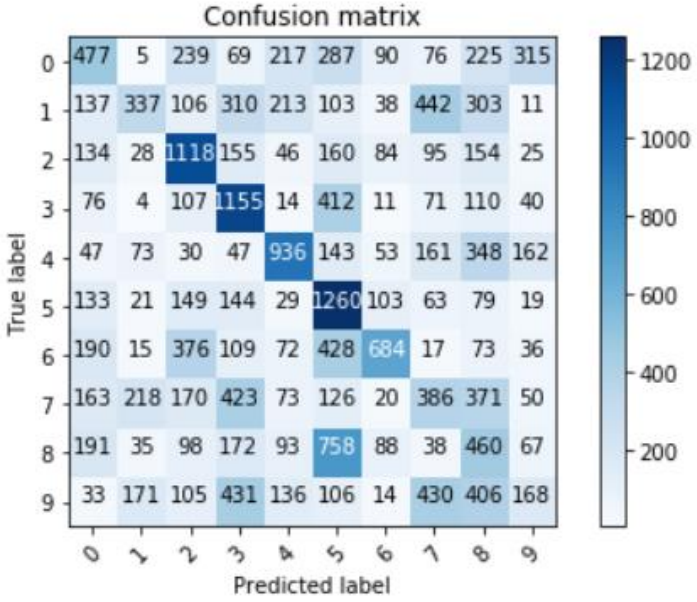
Tuning the hyper parameters – Learning Rate and bias.

Learning Rate	Bias	MNIST_ACC	USPS_ACC
0.001	0.002	89.96	34.3
15	0.05	90.7	34.9

Confusion matrix plot for MNIST



Confusion matrix plot for USPS



3.2 NEURAL NETWORK

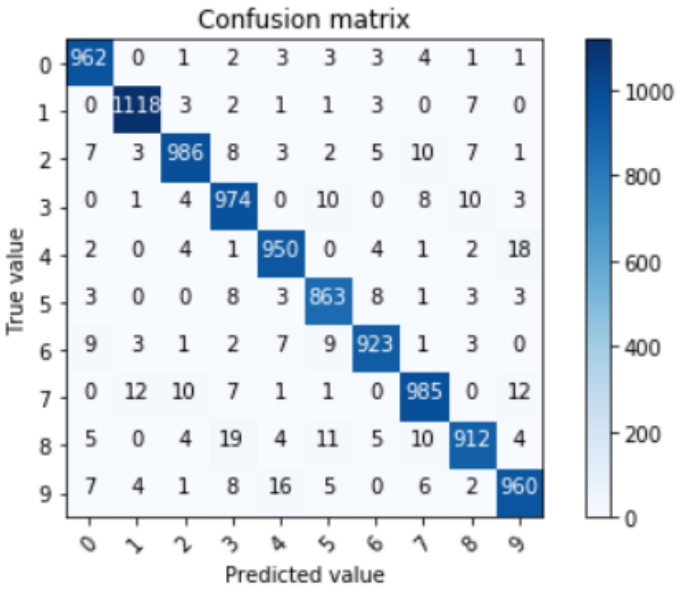
3.2.1 Tune 1

Optimizer = Adam
Batch Size = 6000
Epoch = 30

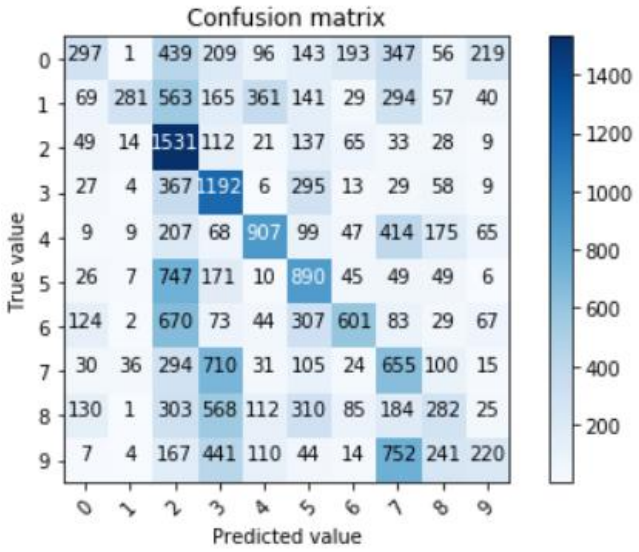
	UNITS	Activation
Layer1	100	relu
Layer2	128	relu
Layer3	10	Softmax

MNIST ACCURACY = ***96.33***
MNIST VALIDATION = 96.47
USPS ACCURACY = ***34.66***

Confusion matrix plot for MNIST



Confusion matrix plot for USPS



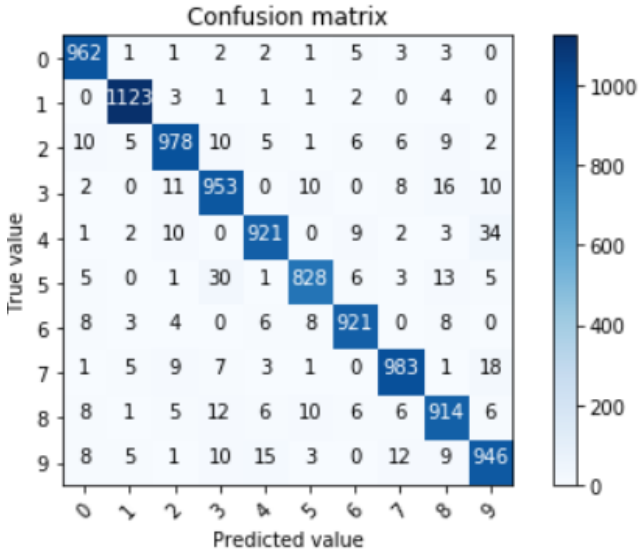
3.2.2 Tune 2

Optimizer = Adam
Batch Size = 200
Epoch = 20

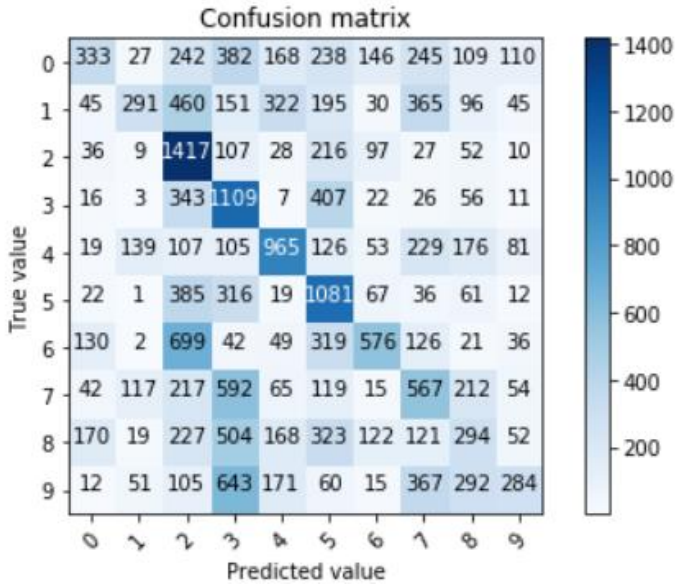
	UNITS	Activation
Layer1	50	softmax
Layer2	100	relu
Layer3	10	Softmax

MNIST ACCURACY = ***94.92***
MNIST VALIDATION = 95.2
USPS ACCURACY = ***35.08***

Confusion matrix plot for MNIST



Confusion matrix plot for USPS

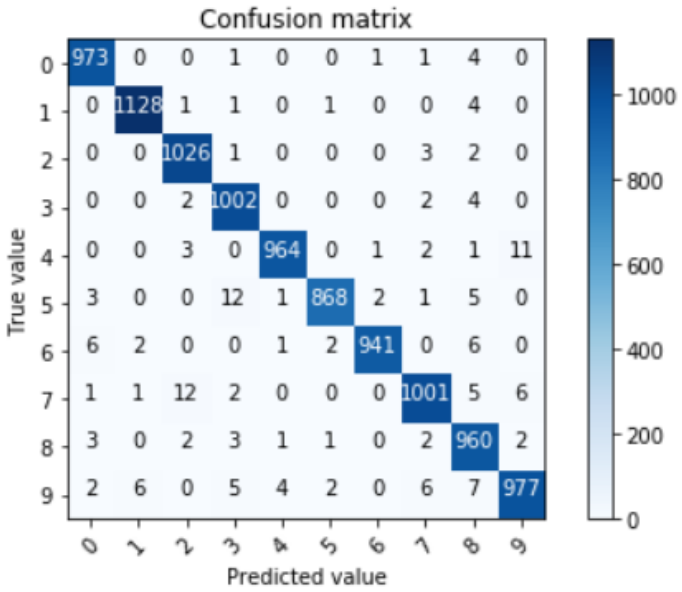


3.3 CNN

I have considered CNN, I have not included the CNN for ensemble only for MNIST with the following hyper parameter settings:
Epoch = 20, batch_size = 200, optimizer = adam , loss = cross_entropy
Layer 1 : neurons = 32, kernel = 5x5 , activation = relu

Layer 2 : neurons = 15, kernel = 3x3 , activation = relu
Layer 3 : neurons = 128, activation = relu
Layer 4 : neurons = 10, activation = softmax

Confusion matrix plot for MNIST



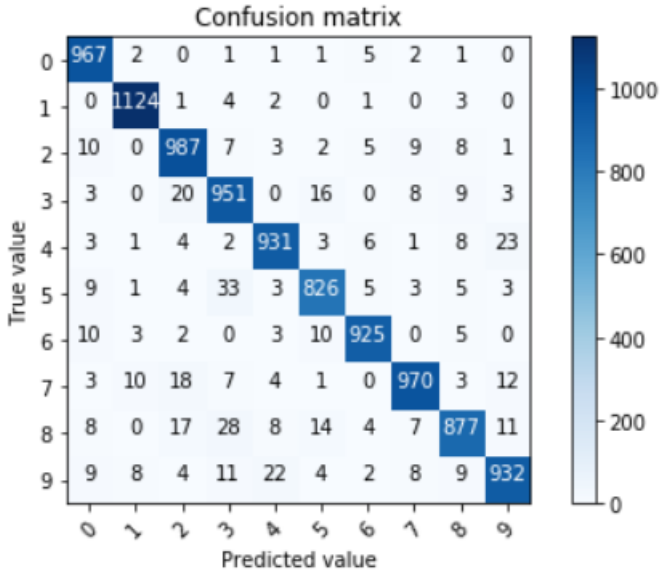
3.4 RANDOM FOREST

3.4.1 Number of Trees : 10

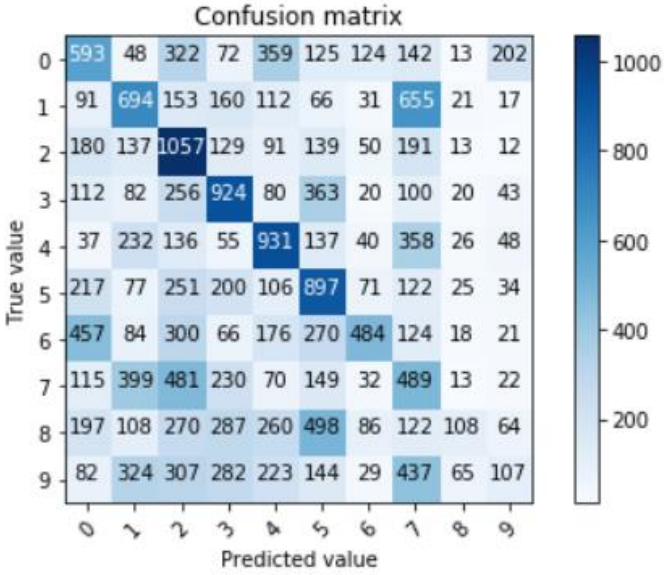
MNIST Accuracy = 94.9

USPS Accuracy = 31.42

Confusion matrix plot for MNIST



Confusion matrix plot for USPS

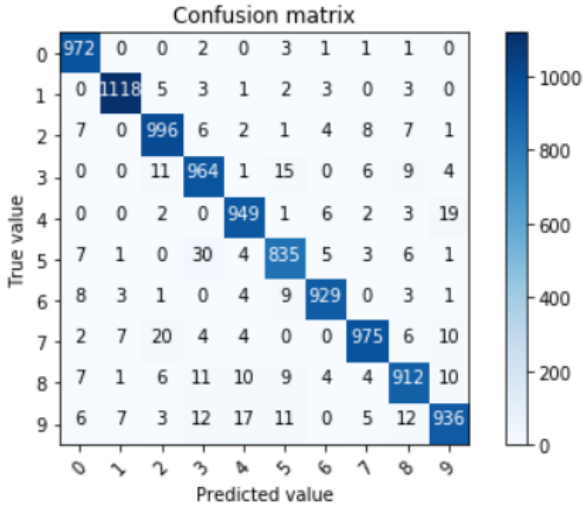


3.4.2 Number of Tress = 20

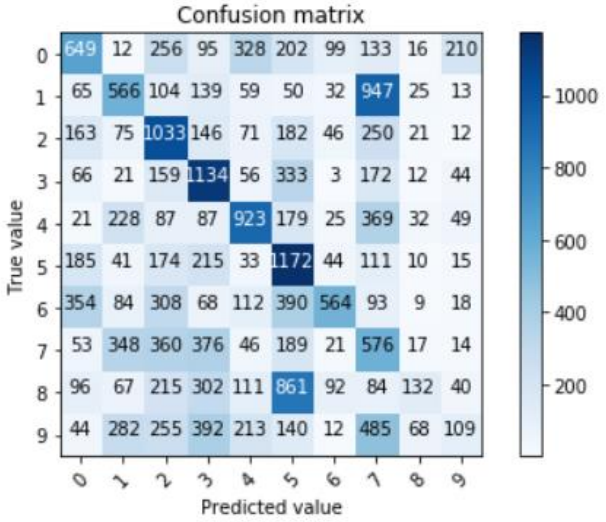
MNIST Accuracy = 95.86

USPS Accuracy = 34.29

Confusion matrix plot for MNIST



Confusion matrix plot for USPS

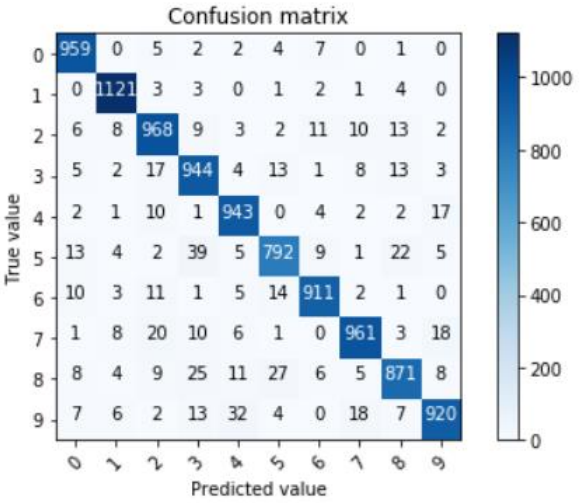


3.5 SVM

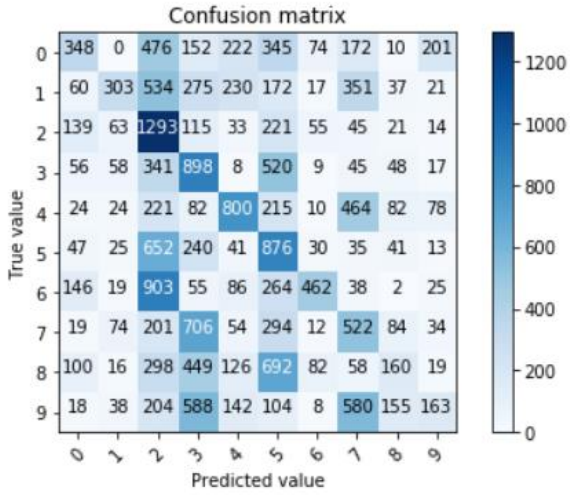
3.5.1 Kernel = Linear

MNIST ACCURACY = 93.9
USPS ACCURACY = 29.12

Confusion matrix plot for MNIST



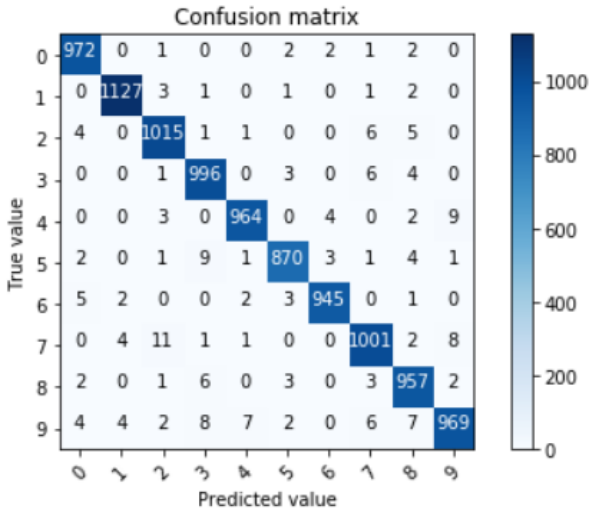
Confusion matrix plot for USPS



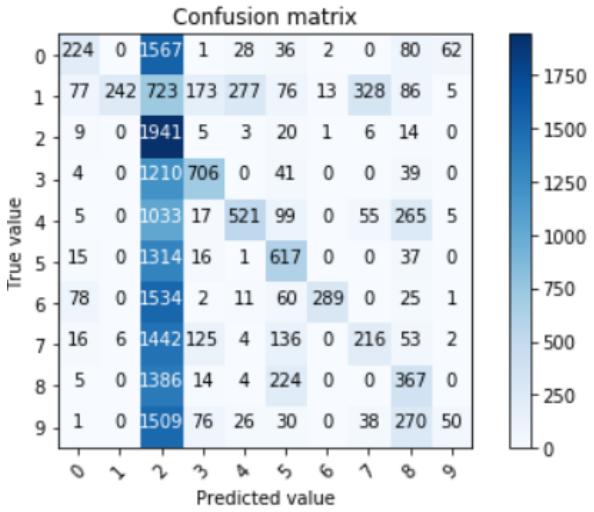
3.5.2 Kernel = rbf , gamma = 1

MNIST ACCURACY = 98.16
USPS ACCURACY = 33.12

Confusion matrix plot for MNIST



Confusion matrix plot for USPS



3.6 ENSEMBLE ACCURACY

For the MAX voting method, I have considered Logistic regression, MLP, Random Forest and SVM. The Accuracy changes every time we run the code, hence

Accuracy for MNIST = ***98.48***

Accuracy for USPS = ***38.7536***

3.7 ANSWERS

3.7.1 No Free Lunch theorem

As the 'free lunch theorem' goes, a model cannot perform well on all the datasets given trained in a single dataset of that instance.

As we can see in the above observations, it is clear that the models get trained on the MNIST dataset performs well on MNIST but not on USPS dataset.

Our results will agree with 'no free lunch theorem'

3.7.2 Classifier Best performance

A confusion matrix is a technique for summarizing the performance of a classification algorithm. Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset. A confusion matrix is a summary of prediction results on a classification problem.

As we can see the confusion matrix plot of each classifier, it is evident that MNIST classifier as better confusion matrix than for USPS dataset.

The confusion matrices of Neural Networks was shown to be the best matrices, among the classifier. Thus, giving an overall best performance.

3.7.3 Overall strength

The overall combined performance accuracy was better than all the models considered (logistic regression, Neural Network, Random forest and SVM) with 97.48 for MNIST dataset and 38.75 for USPS dataset.

4 REFERENCES

1. http://yann.lecun.com/exdb/mnist/
2. https://en.wikipedia.org/wiki/Multilayer_perceptron
3. https://machinelearningmastery.com/logistic-regression-for-machine-learning/
4. https://www.quora.com/How-is-deep-learning-different-from-multilayer-perceptron
5. https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/