Project3:
Classification

Manish Reddy Challamala
Department of Computer Science
University at buffalo
UBIT Name: manishre
Person Number : 50289714
manishre@buffalo.edu

November 22, 2018

Abstract

To train the MNIST Dataset with different classifiers and combine the result of individual classifier to make a final decision.

1 Introduction

The goal of this project is to implement different classification methods for MNIST data set(handwritten digit images) and combine the results of individual classifier to make a final decision.
The features are obtained from two different source:

1. MNIST Dataset
2. USPS Dataset

The above feature data sets are trained by using following methods:

1. Logistic Regression using Softmax.
2. Dense Neural networks[DNN].
3. Convoluted Neural netwotk [CNN].
4. Random Forest
5. Support Vector Machine [SVM]

Train the models using MNIST Dataset and test it on both MNIST and USPS Dataset.

2 Theory

3 Logistic Regression Model with softmax function

The logistic regression model is a classification model with linear regression algorithm which tries to predict y for a given x.

1. The logistic regression gives the probability to which output each input belongs.

2. To generate probability for each input logistic regression takes a function that maps the output between the range of 0 and 1 for all values of input x but as our data consists of multiple classes [i.e; from 0 to 9 classes] we convert our binary classification into multiclass classification problem.

3. By using the following function it is possible to map all the classes:

$$y(X^{(i)}) = Wx(i) + b \tag{1}$$

   where
   x^i is the vector consisting of all features of single image.
   W is the Weight matrix [no of samples(images)X no of categories(classes)]
   B is a bias vector and

4. The hypothesis for logistic regression is given below

$$h_\theta = \frac{e^{W_{yj}^T x^{(i)}}}{\sum_{j=1}^{K} e^{W_j^T x^{(i)}}} \tag{2}$$

   where
   θ is weights

5. The above equation is called the logistic function or the sigmoid function.

6. we calculate the loss function by taking the derivative of logistic function.

$$l(\theta) = logL(\theta) = \sum_{i=1}^{N} y^i log(h(x^i)) + (1 - y^i) log(1 - h(x^i)) \tag{3}$$

7. we calculate the maximum likelihood of the data by minimizing the loss function either by increasing or decreasing the weights that best fit for our data.we can achieve this by taking the partial derivative of loss function.Which is nothing but using the SGD and update the weights iteratively.

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{N} (y^i - \theta^T x^i)^2 \tag{4}$$

$$\theta_j := \theta_j + \alpha(y^i - h_\theta(x^i))x_j^i$$

4 Neural Network

The neural network model is a classification model which tries to predict output y for a given input x

1. The neural network model contain two phase:
   1 Learning Phase
   2. Prediction Phase

2. In learning phase, the neural network takes inputs and corresponding outputs.

3. The neural network process the data and calculates the optimal weights to gain the training experience.

4. In prediction process the neural network is presented with the unseen data where it predicts a output by using its past training experience on that given data.

5. A neural network model can be build by using different layers:
   1. Dense Layer : A linear Operation in which every input is connected to every output by weights
   2. Convolution Layer: A linear operation in which input are connected to output by using subsets of weights of dense layer.

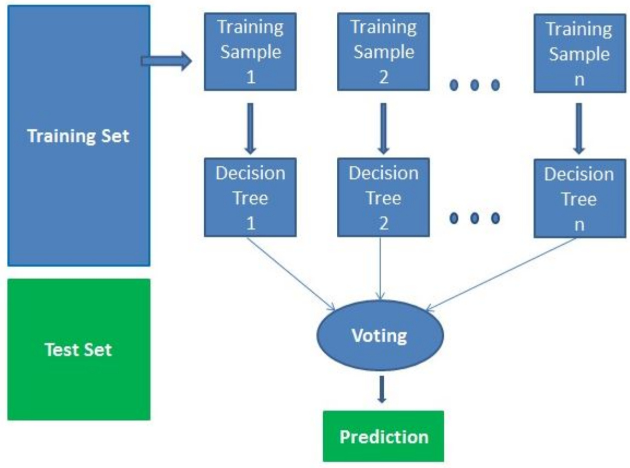
5 Random Forest

1. Random forest is a supervised Algorithm, Which creates the decision tress on randomly selected data samples.

2. From each decision tress the random forest algorithm gets the individual prediction values.

3. The Final prediction values are the values with most votes in individual prediction results.

4. Random Forest is known for its high accuracy and it does not suffer overfitting problem because it takes the average of all predictions.



$fig[1]: \ Block \ Diagram \ of \ Random \ Forest$

6 Support Vector Machine

1. SVM is a supervised learning classification algorithm.

2. SVM differs from all other classifications algorithms,Basically Machine learning algorithms tries to find a boundary that divides the data where error is minimized.

3. But the SVM chooses a decision boundary that maximizes the distance from nearest data points from all classes. which is the optimal decision boundary.

7 Confusion Matrix

1. Confusion Matrix is said to be the performance measurement of machine learning classification problem.

2. It is help-full in measuring the precision, accuracy, specidicity.

3. The Confusion Matrix will give 4 possible type of values
   1. True Postive: The predicted value is positive and its true. which are plotted in diagonal elements of matrix.
   2. True Negitive: The predicted negitive and its true.
   for example we predicted that digit 8 is not 0 and its true.
   3.False Positive: The predicted is positive and its false.which are plotted in upper triangular elements of matrix
   for example we predicted digit 5 has 6 but it is 5.
   4.False Negitive: The predicted is negitive and its false.which are plotted on lower triangular elements of matrix.
   for example we predicted digit 5 has 6 but it is 6.

8 Experimental Setup:

The experimental setup consists of three steps:

1. Data Pre-processing
2. Logistic regression using Softmax Activation Function
3. Dense Neural Network
4. Convolutional Neural Network
5. Random Forest
6. Support Vector Machine

8.1 Data Pre-Processing:

1. In data pre-processing, we have 2 data sets: 1. MNIST Dataset and 2. USPS dataset.
2. Firstly, we are splitting the data into training data, validation data and testing data for given two data set.
3. Total we will be having 2 feature sets generated from the 2 given data sets, Where we split the raw data in accordance to our program requirement.
4. In this process, we are reading the raw image data from a 'mnist.pkl.gz' file and splitting the data into training, validation and testing data .
5. For USPS Dataset, wreading the images from the Numerals folder and converting the images into matrix and normalizing the images by iterating on each image present in that folder.
6. Dimensions for each Dataset is given Below:
   where rows = no of samples and columns = no of features.

   (a) MNIST Dataset

      1. Feature Matrix: 70000 X 784
      2. Target Vector : 70000 X 1
      3. Training Feature Matrix: 50000 X 784
      4. Validation Feature Matrix: 10000 X 784
      5. Testing Feature matrix: 10000 X 784

   (b) USPS Dataset
      1. As USPS Dataset is only used for testing we are not splitting the data.
      2. Feature Matrix: 19999 X 784
      3. Target Vector : 19999 X 1

8.2 Logistic Regression using Softmax activation function:

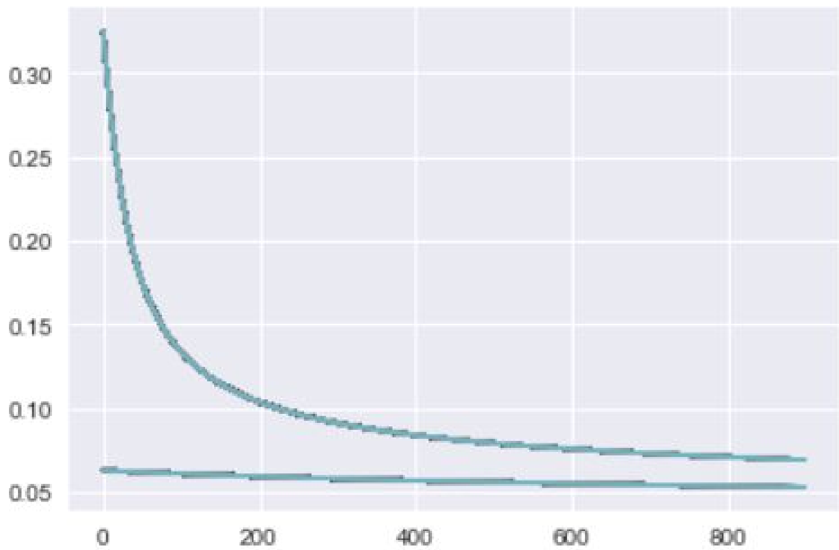
1. In logistic Regression, initialize the random weights.
2. Calculating the optimal weights by plugging the input features and the weight matrix to the softmax function equation[1]
3. By using SGD, we are calculating the loss function at each iteration and updating weights by using equations [4].
4. After, getting the optimized weights we are predicting the outputs for the new unseen data and compare the predicted output with the actual output which gives our accuracy.

8.2.1 Experimental Results:

```
======MNIST DATASET======
Loss for training set 0.06915787942266446
Loss for validation set 0.052854987152375216
Accuracy is:90.22%

Confusion Matrix for MNIST Dataset:
[[ 962    0    2    2    0    2    8    1    3    0]
 [    0 1100    2    4    1    1    4    1   22    0]
 [  11    6  883   24   16    0   16   22   46    8]
 [   5    0   18  901    1   33    5   18   21    8]
 [   1    1    4    1  907    1   16    1    8   42]
 [  15    3    3   41   14  731   15   10   51    9]
 [  16    3    6    2   10   19  898    1    3    0]
 [   2   16   29    4   12    0    0  920    4   41]
 [  10    8   10   25    8   32   11   18  841   11]
 [  10    7    4   11   49   12    1   28    8  879]]

 ======USPS DATASET======

Accuracy is:35.73678683934197%

Confusion Matrix for USPS Dataset:
[[ 620    4  351   58  250  128   76   46  166  301]
 [ 202  308  143  312  240   56   48  384  286   21]
 [ 209   25 1176  141   59   94   91   84   97   23]
 [ 109    3  121 1275   18  251   24   75   78   46]
 [  52   69   36   62 1019  121   48  149  286  158]
 [ 171   22  193  176   36 1122  100   71   83   26]
 [ 335   13  356   87   97  310  668   29   69   36]
 [ 183  218  291  434   68   92   36  329  302   47]
 [ 221   28  151  206  121  577   99   47  469   81]
 [  37  165  156  458  151   85   12  417  358  161]]
```

Strength and Weakness of Confusion Matrix:

1. If we observe the above confusion matrices, In MNIST confusion Matrix 9,040 images out of 10,000 are classified into their true positive state.
2. In remaining 960,images most declassification's are done between digits (1,8);(2,8);(3,8);(4,9);(5,3);(6,5);(7,9);(8,0);(8,9);(9,4);(9,7).
3. from this we can observe that the above some of following digit image needed to be written in more accurate manner.
4. On the other hand, for USPS dataset, 7167 images are classified in their true state in that majority of 2,3,4,5 digits are classified correctly.
5. As we are training on MNIST Dataset and testing on USPS data set there are many mismatches occurred while classifying.



$fig[1]:$ *Loss of training and Validation data.[loss vs no of ephos] [upper plot is training loss]*

8.3 Neural network:

1. In this project I have implemented both Dense and Convoluted neural network.
2. For Dense neural network ,Creating a model with 3 layers, 1. input layer 2. hidden layer 3. output layer
3. No of nodes for each layer is given below:
   1.No of nodes in input layer = No of features in the data set
   2. No of nodes in hidden layer 800
   3.No of nodes in output layer 10 [because we have 10 digits to classify.]

4. Activation functions used in hidden layer is relu [rectified linear unit] because it introduces the non linearity in the network and softmax function is used on the output layer to predict the target class.

5. For Convolution neural network, Creating a model with 4 layers, 1. input layer 2. 2 hidden layers 3. output layer.

6. No of nodes in each layer are given below:
   1. No of input nodes = No of features
   2. No of nodes in hidden layer 1 is 64 and no of nodes in hidden layer 2 is 32
   3. The output layer is dense layer which has 10 nodes.

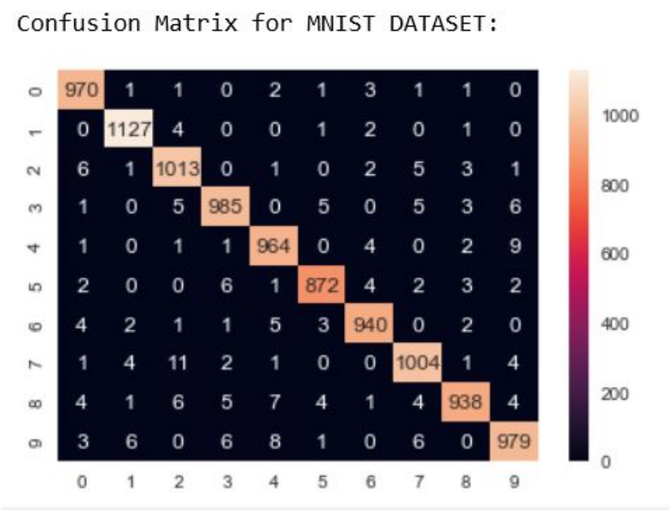
7. By using the above parameters we create 2 models.

8. we run these models by plugging in the appropriate data to it and train the model.

9. After training the model we test the model using the unseen data to predict that which class output belongs with given two datasets.

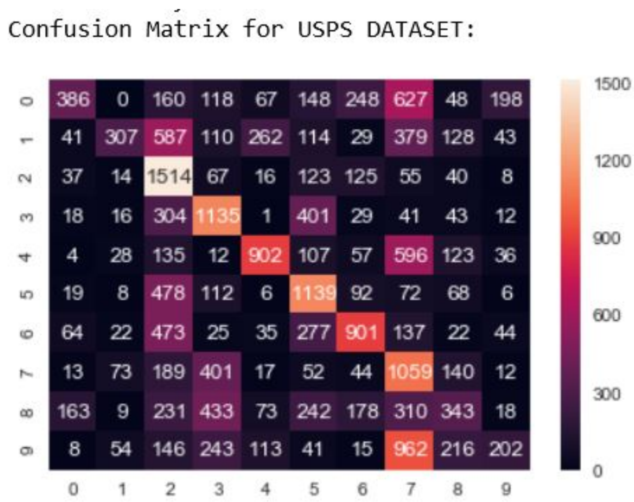
8.3.1 experimental Results:

======Dense Neural Network=======
Test Loss: 0.07109511991865583
Test Accuracy: 97.92

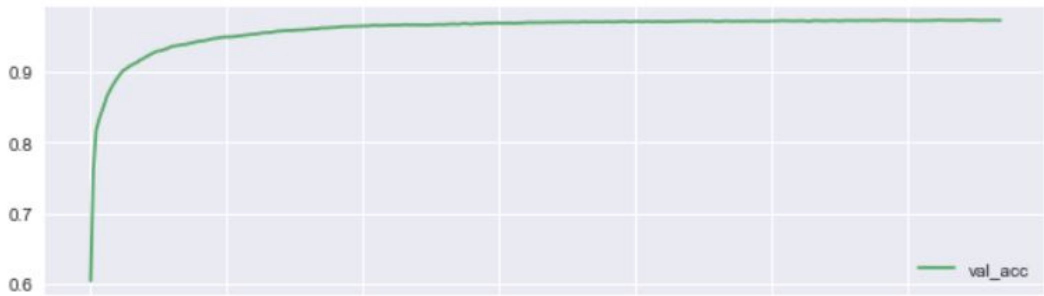


$fig[2]\ Confusion\ Matrix\ for\ MNIST\ Dataset$

Test Loss: 4.985729608269201
Test Accuracy: 39.44197209845591

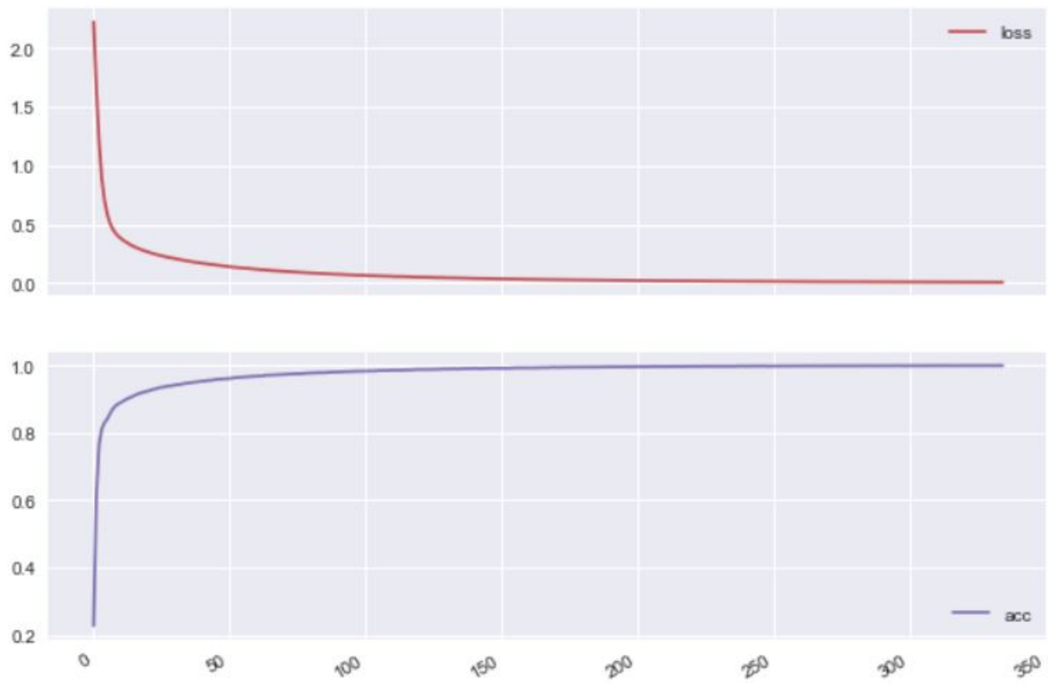


$fig[3]\ confusion\ Matrix\ for\ USPS\ Dataset$



$fig[3]\ validation\ accuracy\ graph$

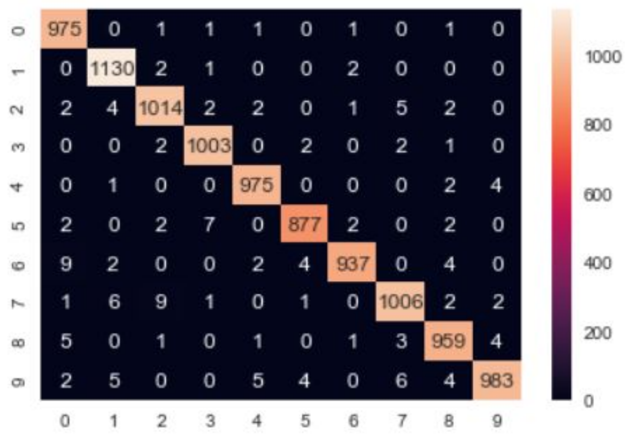
$fig[3]\ validation\ loss\ graph$



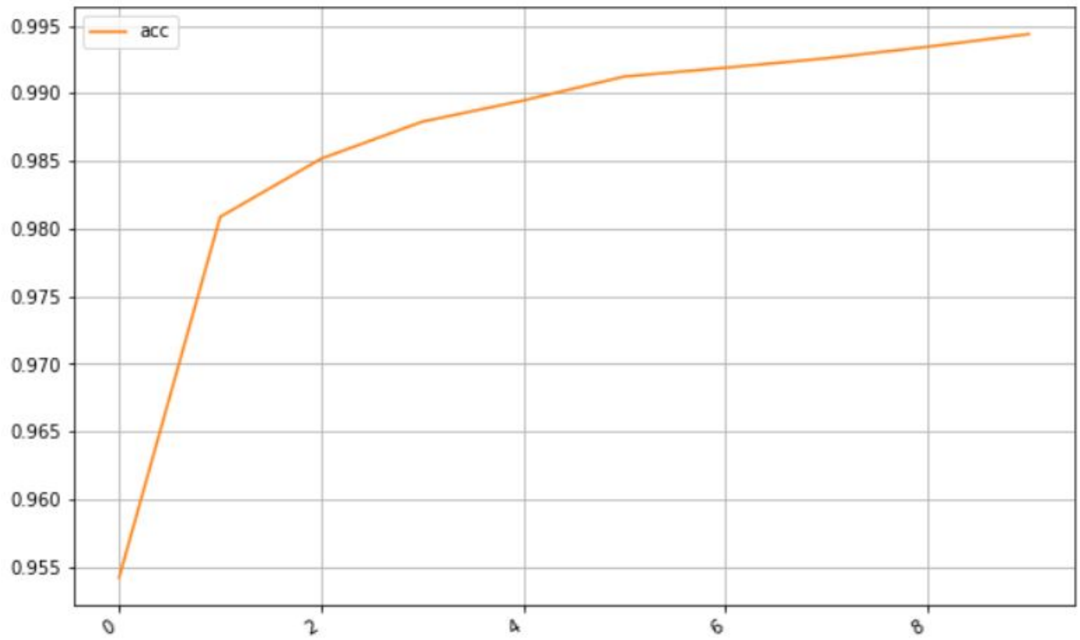
$fig[3]\ accuracyandloss\ graph$

=========Convolution Neural Network =========

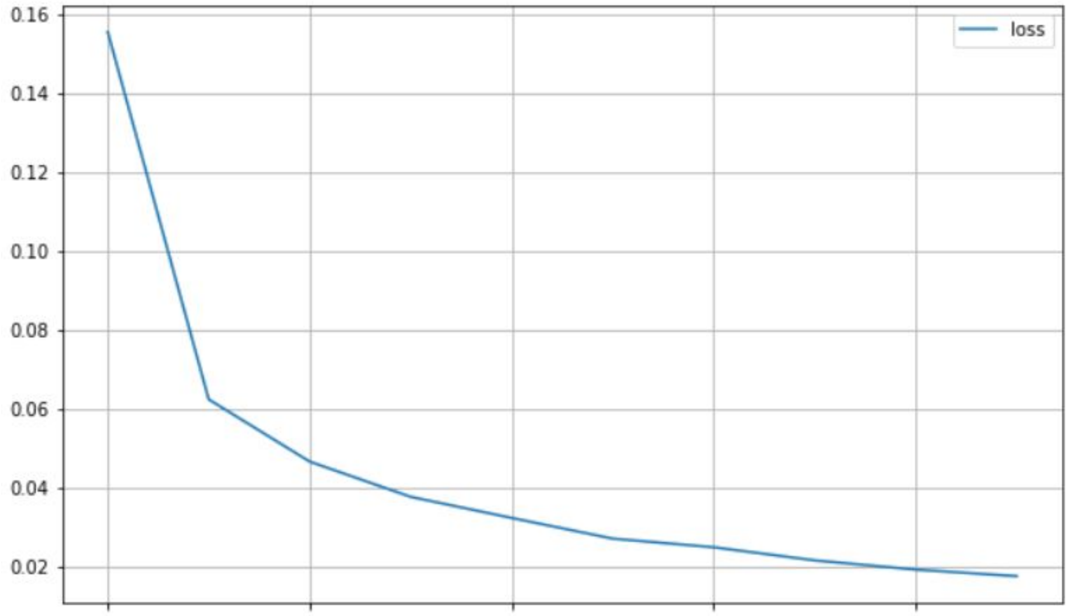
Test Loss: 0.043812535471562296
Test Accuracy: 98.59



$fig[4]\ confusion\ Matrix\ for\ MNIST\ Dataset$



$fig[3]$ *validation loss graph*



$fig[3]$ *validation loss graph*

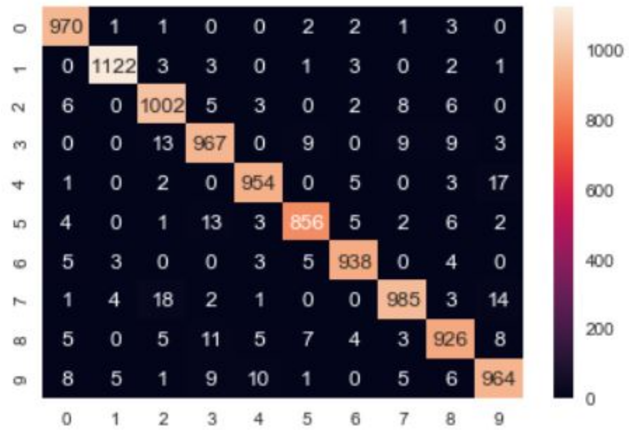
8.4 Random Forest

1. Using the sklearn library and creating the model with nestimators =100, remaining all features are default.

8.4.1 Experimental result:

========Random Forest========

Accuracy: 96.84



fig[5] *confusion Matrix for MNIST Dataset*

Accuracy: 38.791939596979844

fig[6] *confusion Matrix for USPS Dataset*

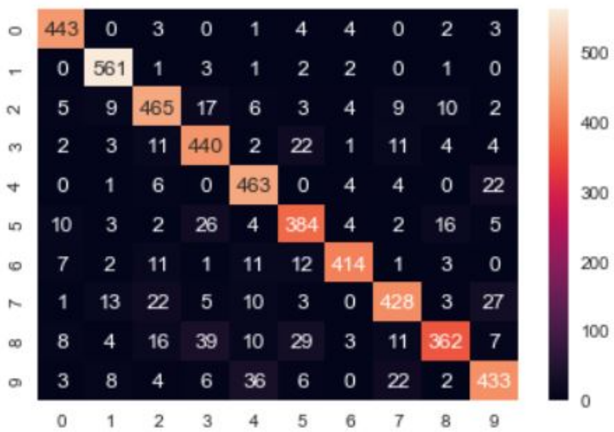
8.5 Support Vector Machine

1. Using the Sklearn library and creating 3 models with following parameters
   1.kernel = linear
   2. kernel ='rbf'(radial function) with gamma =1
   3.kernel='rbf'and remaining default parameters.

8.5.1 Experimental Results:

====== SVM with linear kernel =======

Accuracy: 87.86



fig[7] confusion Matrix for MNIST Dataset

Accuracy: 33.5766788339417

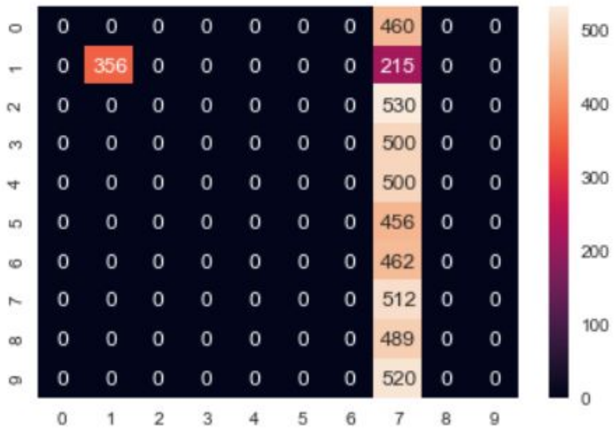
Confusion Matrix for SVM Linear:

$fig[8]\ confusion\ Matrix\ for\ USPS\ Dataset$

======= SVM with radial kernel and gamma =1=======

Accuracy: 35.86

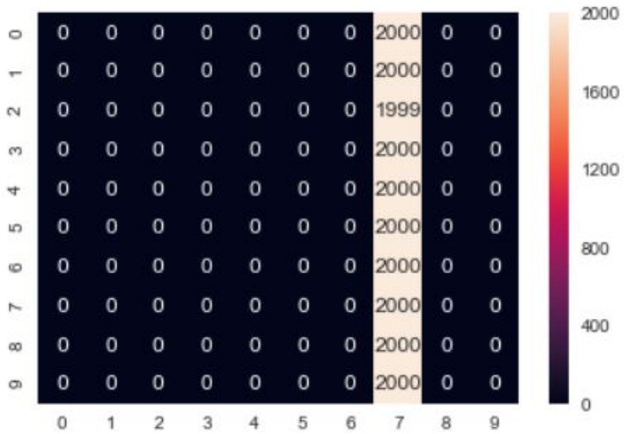
Confusion Matrix for SVM radial:



$fig[9]\ confusion\ Matrix\ for\ MNIST\ Dataset$

Accuracy: 20.54

Confusion Matrix for SVM Linear:

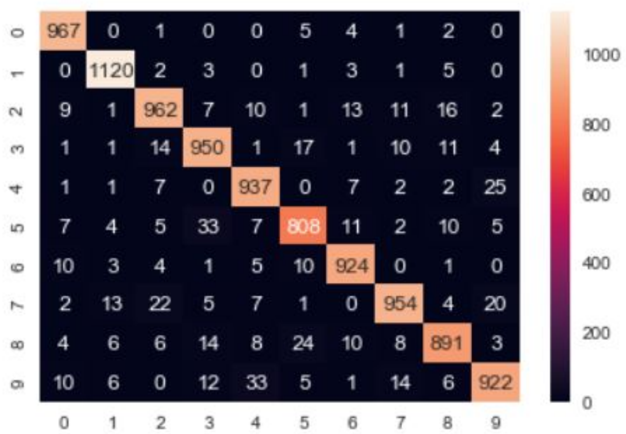


fig[10] confusion Matrix for USPS Dataset

======= SVM with radial kernel and gamma =auto=======

Accuracy: 94.35

Confusion Matrix for SVM radial:



fig[11] confusion Matrix for MNIST Dataset

Accuracy: 38.54192709635482

fig[12] confusion Matrix for USPS Dataset

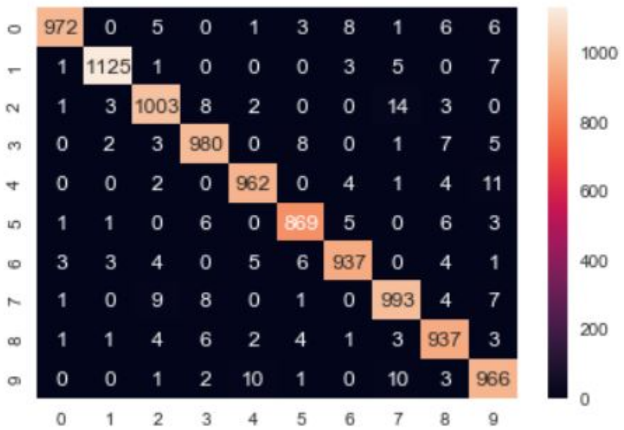
8.6 Ensemble Learning with Hard Voting

1. Make a single model by combining the 4 models.
2. predict the values of the four models, The dimension of the combined predicted values is (10000,1).
3. To find the accuracy and confusion matrix,compare the predicted values with the actual testing target values.

8.6.1 Experimental Results

======= Hard Voting for MNIST dataset=======

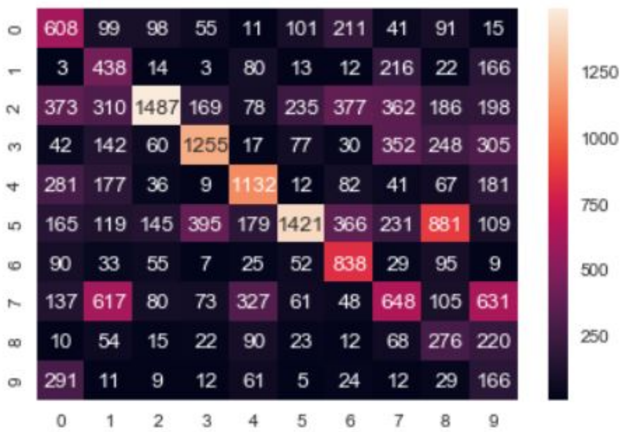
Accuracy: 97.44



$fig[11]$ $confusion\ Matrix\ for\ MNIST\ Dataset$

========Hard Voting For USPS dataset=========

Accuracy: 41.34706735336767



$fig[12]$ $confusion\ Matrix\ for\ USPS\ Dataset$

9 Conclusion:

From the above table[1]: Experimental Results.

1. By comparing the confusion Matrices of all models we can say that convolution neural network has highest precision followed by dense neural network, Random Forest, SVM with radial kernel and gamma =auto, Logistic Regression using Softmax respectively.
2. we can observe that the NO-free lunch theorem holds true for the case.

Reason : By comparing the accuracy of all the models one model is performing better than another and we got highest accuracy by combining the model.

3. By combining the model the overall accuracy of the model increased by 3 percent in usps dataset.

10 References:

1. https://beckernick.github.io/logistic-regression-from-scratch/
2. https://medium.com/@martinpella/logistic-regression-from-scratch-in-python-124c5636b8ac
3. https://github.com/darshanbagul/USPSDigitClassification /blob/master/USPSDigitClassification.ipynb
4. https://www.analyticsvidhya.com/blog/2018/06/ comprehensive-guide-for-ensemble-models/
5. https://www.kaggle.com/ash316/ml-from-scratch-part-2
6. https://stats.stackexchange.com/questions/349540/ hard-voting-soft-voting-in-ensemble-based-methods
7. https://www.toptal.com/machine-learning/ensemble-methods-machine -learning
8. https://forums.fast.ai/t/dense-vs- convolutional-vs-fully-connected-layers/191/3