Project 3

Tanmay Singh
tanmaypr@buffalo.edu

Abstract

The following report is an analysis of the accuracy in recognizing digits on using logistic regression,neural network ,Random Forest and SVM on MNIST and USPS datasets.

1.Introduction

In project 2 we have been provided with two datasets MNIST dataset and USPS datasets. The datasets together provide information on whether the word is written by the same person or not. The human observed dataset consist of 9 features whereas the GSC dataset consist of 512 features.

Confusion Matrix

-confusion matrix is a table that is used to describe how successful was the model in classifying the input values.
-All the values that are along the diagonal from top left to bottom right are correctly predicted.
-All the values that are in the lower part of the diagonal have actual value 1 but have been wrongly predicted as 0
-All the values that are in the upper part of the diagonal have actual value 0 but have been wrongly predicted as 1.
The more number of values on the diagonal better is the model.

2.Neural Network

a)Hyperparameters

Neural Network model has the following hyperparameters
-optimizers
-number of hidden layers
-number of nodes in hidden layers
-dropout rate
-loss

Neural Networks

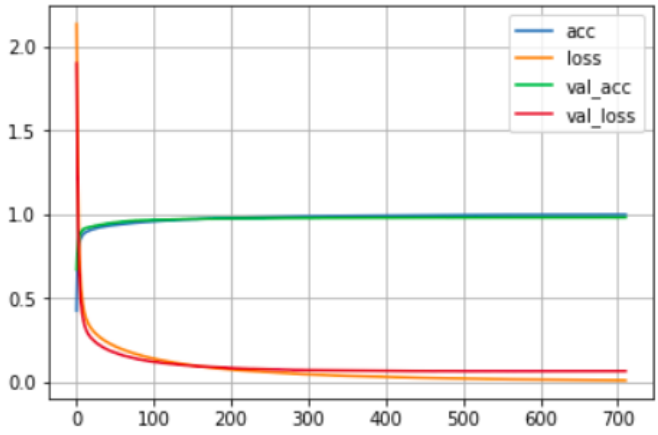
-The loss is calculated using categorical cross-entropy since there are 10 classes we get the predicted values between 0 and 1. Each class is given a probability based upon which we can classify.
-The number of nodes in the output layer is 10 since there are 10 classes from 0 to 9.
- We have to use softmax activation in the last layer since we need to classify the outputs in one of the 10 classes and the addition of all the probabilities should be 1.
-We are performing one hot encoding of the input target matrix of MNIST and USPS dataset so that the target vector can be represented using binary values of 0 and 1. If the target is of the same class the value is 1 otherwise the value is 0.

-We are getting the best results when the hyperparameters are as follow:-
Number of Epooch-1000
Number of nodes in first hidden layer-128
Number of nodes in second hidden layer-128
Batch size-32

we get the following accuracy values:

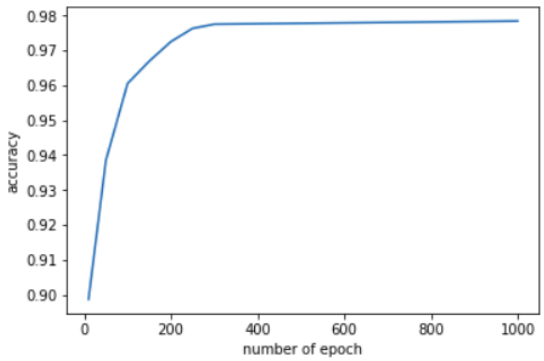
MNIST
Training Accuracy:-0.999
Validation Accuracy:-0.9825
Testing Accuracy:-0.9784

USPS
Testing Accuracy:-0.5238

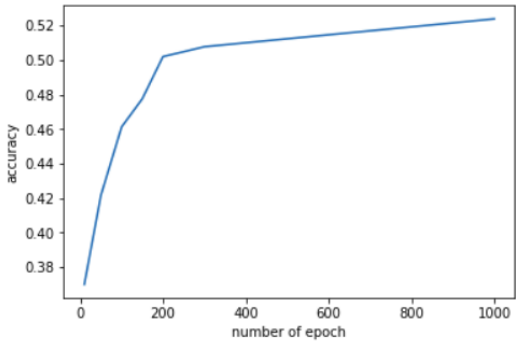


The testing accuracy plot and validation with different number of epoch,number of hidden layers are as follows

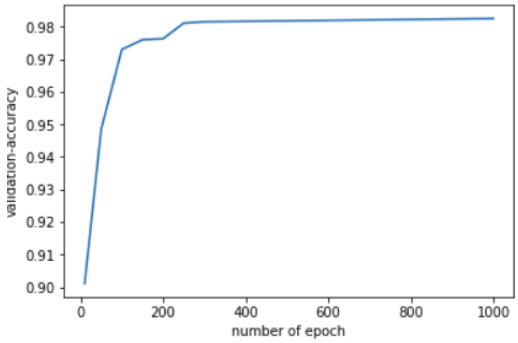
MNIST



USPS



validation accuracy



Confusion Matrix

```
(19999,)
[[ 756    0  114   18  152  108   51   63   77  661]
 [  60  572  186   46  217   51   49  606  148   65]
 [  79   47 1309  129   26  162   76   69   97    5]
 [  27   12  127 1388    4  276    6   38  105   17]
 [  13    9   74   13 1193   92   41  272  199   94]
 [  49    7   33   63   16 1655   14   33  113   17]
 [  78   28  387   29   69  185 1127    7   80   10]
 [  41   68  236  180   29   70    7 1101  227   41]
 [  68   12  192  228   83  335   60  100  883   39]
 [  13   40   72  142  212   41    5  597  386  492]]
('Testing accuracy usps', 0.5238261913095654)
(10000,)
[[ 967    0    1    1    1    1    4    1    3    1]
 [   0 1124    2    2    0    1    3    0    3    0]
 [   4    1 1009    3    1    0    3    7    4    0]
 [   1    0    6  986    0    7    0    4    4    2]
 [   0    0    2    0  964    0    6    2    0    8]
 [   5    1    0    8    2  867    5    0    2    2]
 [   5    3    2    1    3    2  939    0    3    0]
 [   0    6    6    4    0    0    0 1004    2    6]
 [   5    0    2    5    4    2    3    3  946    4]
 [   2    2    0    3   13    5    0    5    1  978]]
('Testing accuracy mnist', 0.9784)
```

The neural network classifier gives the best result with the mnist testing accuracy of 97% as well as the usps testing accuracy of 52%. The diagonal elements of the mnist confusion matrix indicate that majority of the digits were correctly classified.The confusion for other classification has not gone in double digits in any instance.

3.Logistic Regression Model

a) Hyper-parameters

-Learning Rate
-size of mini batch
-number of iterations

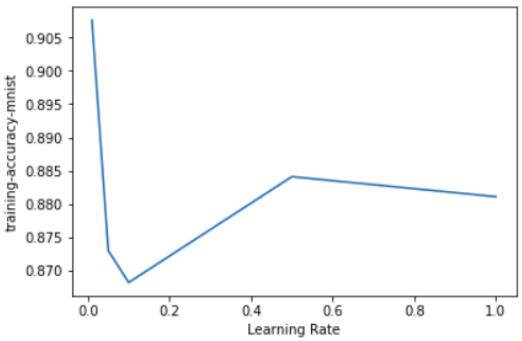
Logistic Regression
- Similar to neural network, logistic regression also uses softmax activation to make prediction for the given datasets.
- mini batch gradient descent is used in the logistic regression as it significantly increases the speed of the model.
- The gradient is being normalized in the model as the value is not the concern rather the direction of the descent is the concern because of this the speed of the model increases considerably.
-The number of iterations significantly affects the performance since if the iterations are very high it might lead to over fitting of data.

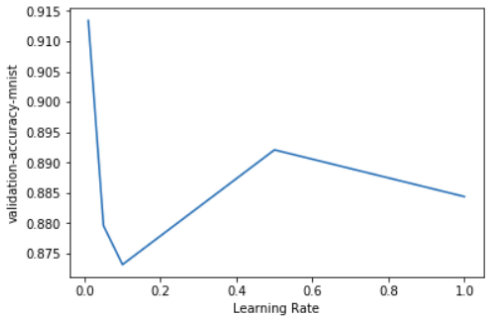
We are getting the best results when the hyperparameters are as follows:

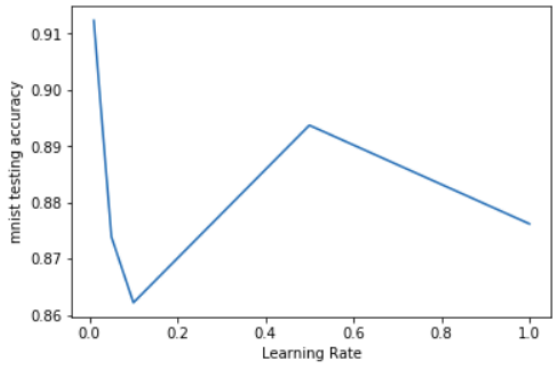
Learning Rate:0.01
Size of mini batch:100

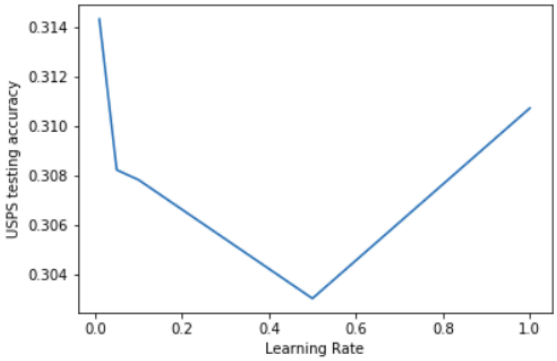
MNIST Training accuracy:-0.9279
MNIST Validation accuracy:-0.9177
MNIST Testing accuracy:-0.9118
USPS Testing accuracy:-0.315515

The testing accuracy plot and validation with different Learning Rate ,Batch size are as follows









Confusion Matrix

```
[[4802    1   19   10    9   22   25    2   34    8]
 [   2 5538   27   19    4   11    4   12   50   11]
 [  20   46 4522   84   40   24   41   51  122   18]
 [  18   14  102 4603    5  148   13   36  121   41]
 [   7   15   30    6 4543    8   36   14   26  174]
 [  39   13   36  156   33 3955   63   10  159   42]
 [  28    7   31    2   24   51 4775    2   28    3]
 [   9   14   52   21   32    8    3 4834   21  181]
 [  24   72   60  130   22  138   37   15 4281   63]
 [  16   15   15   66  109   26    1  141   57 4542]]
('Training Accuracy MNIST', 0.9279)
[[ 955    0    6    1    2    7    8    6    6    0]
 [   0 1044    2    7    1    3    0    1    6    0]
 [   4   11  893   14    9    5    5   14   30    5]
 [   5    1   21  905    1   48    2    7   31    9]
 [   2    9    5    3  917    1   10    7    5   24]
 [  13    3    9   33    8  774   31    5   34    5]
 [   2    2    7    0    6   11  934    0    5    0]
 [   6    4    6    7    5    1    0 1028    2   31]
 [   2   19   15   24    5   34    8    8  878   16]
 [   3    8    3   14   22    7    1   45    9  849]]
('Validation Accuracy MNIST', 0.9177)
```

```
[[ 948    0    1    2    1    5   11    7    5    0]
 [   0 1118    5    2    0    1    3    1    5    0]
 [   8   11  908   19    7    4   10   11   50    4]
 [   4    1   24  903    1   30    3   13   25    6]
 [   1    2    8    1  903    0   12    6   12   37]
 [  12    5    2   33    7  762   20   10   38    3]
 [   8    3    5    3    8   21  904    2    4    0]
 [   0    7   21   10    4    1    0  943    4   38]
 [   5   11    8   26    7   33   14   15  844   11]
 [   7    6    1    9   39   10    0   39   13  885]]
('Testing Accuracy MNIST', 0.9118)
(19999,)
[[ 657    4  101   83  180  245   79  172   87  392]
 [ 103  319   66  166  387   98   61  478  246   76]
 [ 147   74  979  175   45  241  135   65  107   31]
 [  44   17  427  579   22  665   14   72  123   37]
 [  46   11   99   48  796  141   87  392  241  139]
 [ 145    9  113  149   99 1199   47   77  119   43]
 [ 197   11  715  114   68  413  433    6   30   13]
 [ 121   55   90  318  121  169   11  746  279   90]
 [ 206   22  175  398  169  514   72   94  290   60]
 [  52   28   64  333  207   96   13  642  253  312]]
('Accuracy USPS', 0.3155157757887894)
```

The confusion matrix of the training mnist shows that there are some classes for which the confusion of the model is quite high but this is not the case with the confusion matrix of validation and testing datasets of mnist.The confusion matrix of the usps dataset is worse than that of neural network.

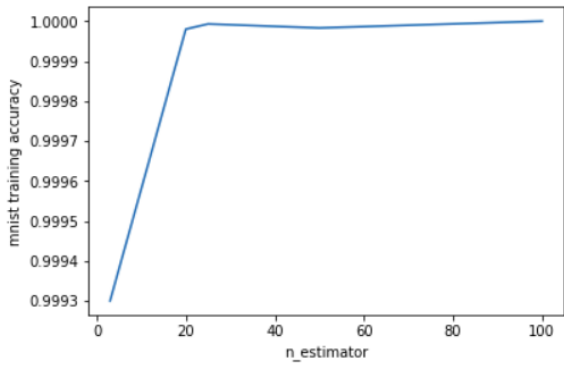
4.Random Forest

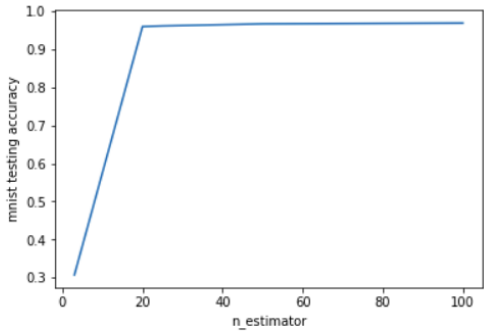
a)Hyperperameters
-no of estimators:- This gives the number of trees which are present in the forest

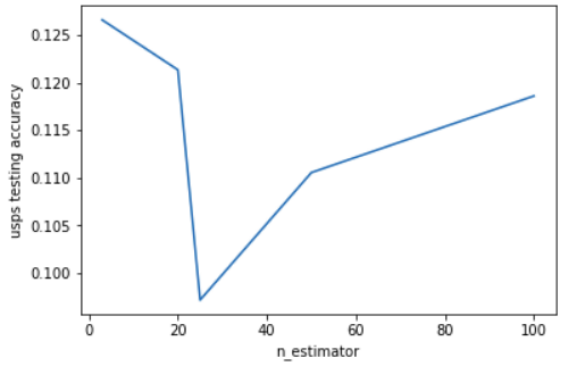
Random Forest Classifier
-Random forest uses the dataset to create number of tress each classifying the input into one of the 10 classes.
-There are n_estimators number of trees present in the forest.

-The final classification is done based on the mode of all the sub trees of the forest very similar to hard majority voting in ensemble.
- If the number of trees are very high it might lead to overfitting of data as well as the time required for computation will increase considerably.

The testing accuracy plot and validation with different n_estimators are as follows:







Confusion Matrix

```
[[5923    0    0    0    0    0    0    0    0]
 [   0 6742    0    0    0    0    0    0    0]
 [   0    0 5958    0    0    0    0    0    0]
 [   0    0    0 6131    0    0    0    0    0]
 [   0    0    0    0 5842    0    0    0    0]
 [   0    0    0    0    0 5421    0    0    0]
 [   0    0    0    0    0    0 5918    0    0]
 [   0    0    0    0    0    0    0 6265    0]
 [   0    0    0    0    0    0    0    0 1800]]
('Testing Accuracy MNIST', 9.67548377418871e-06)
[[   0  166    0    0    0  236    0 1598    0    0]
 [   0  493    0    0    0   27    0 1480    0    0]
 [   0  433    0    0    0   61    0 1505    0    0]
 [   0  203    0    2    0  193    0 1602    0    0]
 [   0  391    0    0    3  103    0 1503    0    0]
 [   0  243    0    0    0  411    0 1346    0    0]
 [   0  294    0    0    0  358    0 1348    0    0]
 [   0  458    0    0    0   32    0 1510    0    0]
 [   0  389    0    0    2  598    0 1011    0    0]
 [   0  332    0    0    0   67    0 1601    0    0]]
('Testing Accuracy usps', 0.12095604780239012)
```

we can clearly see that the model is overfitted for the mnist dataset. We are getting a very high accuracy for mnist testing and very low accuracy for usps testing.

4.SVM

a)Hyperparameters
kernel
gamma

SVM
kernel-linear,radial basis function

gamma- 1,default

-The compilation time of SVM is very high because of the calculation of the distance function
- SVM was compiled on google collab
- The accuracy for MNIST dataset is 0.9827
-The accuracy for USPS dataset is 0.261413

Confusion Matrix

```
confusion matrix
[[ 974    0    1    0    0    1    1    1    2    0]
 [   0 1128    3    1    0    1    0    1    1    0]
 [   4    0 1015    1    1    0    0    6    5    0]
 [   0    0    1  997    0    3    0    5    4    0]
 [   0    1    3    0  964    0    4    0    2    8]
 [   2    0    1    7    1  872    3    1    4    1]
 [   5    2    0    0    2    3  945    0    1    0]
 [   0    3    9    1    1    0    0 1004    2    8]
 [   2    0    1    6    1    2    0    2  958    2]
 [   4    4    2    8    7    2    0    6    6  970]]


[[ 226    0 1564    2   26   35    2    0   79   66]
 [  78  257  713  172  262   77   12  337   88    4]
 [   8    0 1944    6    2   20    1    6   11    1]
 [   4    0 1193  725    0   41    0    0   37    0]
 [   6    0 1045   18  522   96    0   56  252    5]
 [  15    0 1305   16    1  626    0    0   37    0]
 [  78    0 1534    2   10   61  290    0   22    3]
 [  17    6 1435  129    6  134    0  220   52    1]
 [   7    0 1387   14    4  221    0    0  367    0]
 [   1    0 1508   79   26   29    0   39  267   51]]
```

5.Majority Voting

- In majority voting we take the predicted values of all the 4 models and create a matrix where each column is a prediction vector.
-on each row we select that which occurs maximum number of times and we store it in our final ensembled prediction array.

We get the final testing accuracy as follows:-

MNIST:-97.1
USPS:-48.85244

we can clearly see that the ensembled predicted value for usps is better than all other models except the best predicted value of neural network.

Ensemble Confusion Matrix

```
ensembled mnist accuracy: 0.971
[[ 967    0    1    1    1    0    4    1    4    1]
 [   0 1123    2    2    0    1    3    0    4    0]
 [   4    1  985    3    7    0    3    6   23    0]
 [   1    0    6  981    1    7    0    4    8    2]
 [   0    0    2    0  964    0    6    2    2    6]
 [   5    1    0    9    5  849    5    0   16    2]
 [   5    3    2    1    5    2  936    0    4    0]
 [   0    6    6    5    0    0    0 1001    4    6]
 [   5    0    2    5    5    2    3    3  945    4]
 [   2    1    0    4   27    5    0    5    6  959]]
```

```
ensembled USPS accuracy: 0.488524426221
[[ 707    4  108   17  144  156   48  165   69  582]
 [  50  563  159   36  177   52   47  750  131   35]
 [  74   75 1255  120   24  169   73  109   95    5]
 [  26   16  127 1279    4  333    5   96  100   14]
 [  12   21   70   11 1033   89   35  482  181   66]
 [  45   15   29   58   15 1621   13   92   99   13]
 [  71   31  363   29   67  274 1071   11   73   10]
 [  35  101  222  163   29   77    7 1125  213   28]
 [  59   34  169  210   76  481   57  136  750   28]
 [  12   67   65  124  183   50    5  789  339  366]]
```

6.Free Lunch Theorem

-The free lunch theorem generally says that if a model is formed for a particular type of dataset and it gives a very good result on that model it is going to perform worse than random for other datasets.
-our results perfectly support the free lunch theorem since in all the models we are able to get a very high accuracy with the mnist datasets but the model gives very poor accuracy for the usps datasets.
-The accuracy for usps datasets is in the range of 30 to 40 which is worse than selecting randomly,for all the models except neural network where the accuracy is close to 50%.

References

1)https://blog.statsbot.co/ensemble-learning-d1dcd548e936
2)http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice/
3)https://docs.scipy.org/doc