CSE 574 - Project 3: Classification

Baskar Adyar Krishnan
Person 50291475
UBIT: BASKARAD

Abstract

This project is to implement Neural Networks, Logistic Regression, SVM and Random Forest methods for the task of classification.

1 Dataset

We are going to work with two different datasets: 1. MNIST digit images dataset and 2. United States Postal Service(USPS) dataset.

For both training and testing of our classifiers, we will use the MNIST dataset. The MNIST database is a large database of handwritten digits that is commonly used for training various image processing systems.

We use USPS handwritten digit as another testing data for this project to test whether the models could be generalize to a new population of data.

2 Logistic Regression

2.1 Number of iterations:

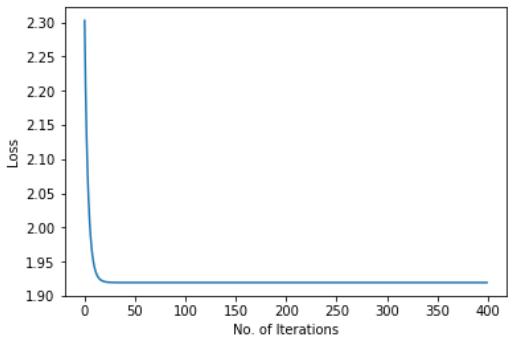

Figure 1: 400 iterations

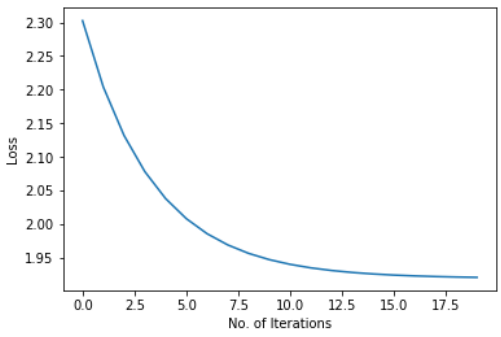

Figure 2: 20 iterations

Initially the model was trained for 10,000 iterations. On see the graph, we could notice that there was not enough decrease in the loss or increase in accuracy after certain number of iterations.

Thus the number of epochs is reduced to 1000 and later to 400 as there is no improvement in the model for more number of epochs. On testing the model was tested with 10,000, 1000, 400, 100 and 20 iterations, we can say for this dataset the model attains saturation after 20 epochs.

There is no change is test accuracy accuracy or loss irrespective of the number of epochs after 20.

2.2 Lamda:

Model is tested with lambda in range 0.01 to 10 ([0.01,0.05,0.1,0.2,0.5,1,5,10]). On analyzing the loss and accuracy we can say that when lambda = 0.05, model has the lowest loss. While the accuracy changes in the small scale with respect to the loss.

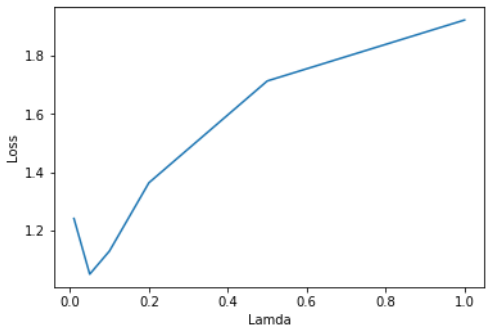

Figure 3: Loss

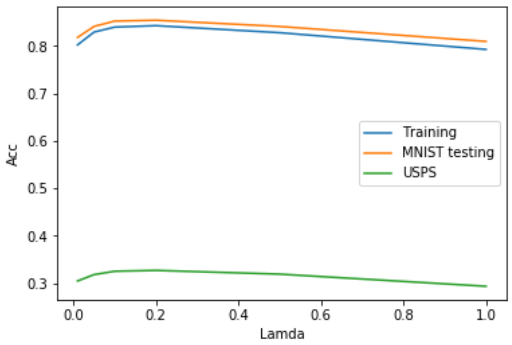

Figure 4: Accuracy

2.3 Testing results for MNIST testing data:

Accuracy for MNIST test data with Logistic Regression: 81.67999999999999

Confusion Matrix for MNIST test data with Logistic Regression:

```
[[ 931    0    5    7    0    0   26    1   10    0]
 [   0 1086    9    5    0    0    4    1   30    0]
 [  28   52  802   35   16    0   35   23   40    1]
 [   8    7   25  892    0    4    9   19   31   15]
 [   6   15    4    0  803    0   33    2   18  101]
 [  46   42   11  225   26  380   45   27   60   30]
 [  30   15   19    2    8   10  869    0    5    0]
 [   9   54   25    0   10    0    3  881   11   35]
 [  19   39   15   98    8    4   21   18  728   24]
 [  25   23   11   14   68    2    3   53   14  796]]
```

Figure 5: For MNIST

Confusion Matrix for USPS data with Logistic Regression:

```
[[ 881    9  392   60  361    6   61   29   50  151]
 [ 336  313  231  211  185    9   52  379  274   10]
 [ 428   64 1054  106   55    2  107   99   74   10]
 [ 310    7  147 1114   55   44   59   88  123   53]
 [ 274  112   47   80 1018   29   32  143  201   64]
 [ 460   46  265  382   53  352  157  100  132   53]
 [ 785   19  343   86  104   16  569   22   39   17]
 [ 281  293  391  345   54   23   61  274  247   31]
 [ 434   78  274  231  178  113  168   48  424   52]
 [ 177  291  204  402  173    9   24  362  276   82]]
```

Figure 6: For USPS

Figure 7: Confusion Matrix

2.4 Testing results for USPS data:

Accuracy for USPS data with Logistic Regression: 30.406520326016302

2.5 Inference:

In order to obtain the max accuracy of the model with logistic regression for hyper-parameters are set as discussed above and we got good accuracy for MNIST dataset and very low (30.40 percentage) for USPS dataset.

3 Support Vector Machine (SVM):

SVM is implemented by importing the library sklearn. Three different settings were done to obtain better results. They are:

3.1 Using linear kernel:

All other parameters are kept default. If gamma is default, sklearn will take ‘auto’ which uses 1 / n_features as gamma value.

```
Confusion Matrix for MNIST test data with SVM:

[[ 967    0    1    0    0    5    4    1    2    0]
 [   0 1120    2    3    0    1    3    1    5    0]
 [   9    1  962    7   10    1   13   11   16    2]
 [   1    1   14  950    1   17    1   10   11    4]
 [   1    1    7    0  937    0    7    2    2   25]
 [   7    4    5   33    7  808   11    2   10    5]
 [  10    3    4    1    5   10  924    0    1    0]
 [   2   13   22    5    7    1    0  954    4   20]
 [   4    6    6   14    8   24   10    8  891    3]
 [  10    6    0   12   33    5    1   14    6  922]]
```

Figure 8: For MNIST

```
Confusion Matrix for USPS data with SVM:

[[ 573    2  428   19  285  248   73   44    6  322]
 [ 110  429  285  137  273  180   46  501   22   17]
 [ 128   18 1402   59   39  198   61   57   23   14]
 [  76    3  186 1123   11  483    5   70   27   16]
 [  18   67   91   14 1167  267   22  194   69   91]
 [ 108   17  257  102   25 1367   60   43   15    6]
 [ 197    7  489   24   98  394  748   13    7   23]
 [  50  225  457  265   57  416   15  452   41   22]
 [  73   25  209  193   87 1006   95   41  244   27]
 [  26  166  228  278  213  165    8  499  214  203]]
```

Figure 9: For USPS

Figure 10: Confusion Matrix for SVM ('rbf' with gamma =default)

Testing Accuracy for MNIST testing Data: 95.62
Testing Accuracy for USPS Data: 32.4248849651

3.2 Using radial basis function with gamma = 1 :

All other parameters are kept default in this setting. The accuracy of this model was very poor as it couldn't even predict the MNIST testing data properly.
Testing Accuracy for MNIST testing Data: 24.14
Testing Accuracy for USPS Data: 10.56514

3.3 Using radial basis function with gamma = default :

sklearn will take 'auto' which uses 1 / n_features as gamma value if it is set to default
Testing Accuracy for MNIST testing Data: 94.35
Testing Accuracy for USPS Data: 38.54192709

3.4 Inference:

Third setting (Using radial basis function with gamma = default) is better than other two setting because it gives better accuracy for USPS data which the model has never seen before. We can say that the model is predicting the output for unseen data with higher accuracy for rbf with gamma= default.

4 Neural Network:

4.1 Testing with different activation functions :

Two different activation functions are tested - sigmoid and relu. Sigmoid gave accuracy of **85.19 percent** for MNIST testing data and **31.8465 percent** for USPS data. Relu gave accuracy of **90.09 percent** for MNIST testing data and **35.8465 percent** for USPS data.

Clearly, Relu is better for this model. One major benefit of Relu is the reduced likelihood of the gradient to vanish. From the function definition of Relu, we know that Relu gives the same output when the function is greater than zero.

softmax is used in the last layers because we are dealing with the multiple class classification problem. softmax gives the probabilities of target being that class.

4.2 By varying number of hidden layers :

Number of hidden layers increased from 1 to 3. As the number of hidden layer increases the accuracy increases. we can settle with the 2 hidden layers as it doesn't change much in accuracy with the 3 hidder layer model.

```
Confusion Matrix for MNIST data with Neural Network:

[[ 959    0    5    2    1    4    6    1    2    0]
 [   0 1096    4    6    1    1    4    2   21    0]
 [  10    3  937   17   15    1   11   13   23    2]
 [   3    1   16  928    0   26    2   14   14    6]
 [   1    1    6    2  917    1   10    2    4   38]
 [  10    3    5   48   11  764   16    7   23    5]
 [  15    3    8    1   12   12  906    0    1    0]
 [   4   11   24    6    8    0    0  947    3   25]
 [   9    3    6   33    8   26   16   11  854    8]
 [  14    4    2   14   48    8    0   24    7  888]]
```

Figure 11: For MNIST

```
Accuracy for USPS data with Neural Network:
[[ 528    4  308   45  246  218   91   73  171  316]
 [ 143  335  172  365  179   90   36  471  189   20]
 [ 159   16 1288  163   43  131   72   58   50   19]
 [  52    4  147 1430    7  219   12   56   46   27]
 [  36   58   45   54 1037  129   38  182  235  186]
 [ 118   23  325  194   39 1045  115   67   51   23]
 [ 266   12  572   96   85  237  645   23   35   29]
 [ 132  231  308  428   46  217   13  394  176   55]
 [ 150   32  230  190  115  671  123   67  355   67]
 [  34  172  165  387  156   89   10  474  304  209]]
```

Figure 12: For USPS

Figure 13: Confusion Matrix

4.3 Inference:

The overall accuracy from Neural network with 2 hidden layer and relu and softmax as activation funtion is:
Accuracy for MNIST data with Neural Network: 91.96
Accuracy for USPS data with Neural Network: 36.33181659082954

5 Random Forest

Random forest is implemented using sklearn library.

5.1 Varying the number of trees:

The model is tested with three different number of trees, 10 , 100 and 1000. By increasing the number of trees the accuracy of the model increases. For MNIST it increased from 94.53 to 97.07000000000001 percentage. And for USPS from 31.096554 to 40.7020351 percentage.

```
Confusion Matrix for MNIST test data with Random Forest:

[[ 970    0    0    0    0    2    3    1    3    1]
 [   0 1122    3    3    0    2    2    0    2    1]
 [   6    0 1001    5    2    0    4    8    6    0]
 [   0    0    9  974    0    6    0    9    8    4]
 [   1    0    1    0  957    0    5    0    2   16]
 [   2    0    0   11    3  860    6    2    5    3]
 [   6    3    0    0    2    3  940    0    4    0]
 [   1    2   18    0    1    0    0  994    2   10]
 [   3    0    6   10    4    5    4    4  927   11]
 [   5    5    2   12   11    3    1    4    4  962]]
```

Figure 14: For MNIST

```
Confusion Matrix for USPS data with Random Forest:

[[ 651   13  259   48  455  154   66   93    1  260]
 [  43  563  113  107   58   96   20  983   16    1]
 [  95   24 1282   57   53  192   22  269    4    1]
 [  40    7   89 1294   57  306    3  183    3   18]
 [  13  208   43   20 1091  177   15  389   26   18]
 [ 147   29  133   65   25 1459   23  108    7    4]
 [ 295   45  220   19   87  345  844  132    1   12]
 [  42  328  365  255   41  242   34  682    1   10]
 [  50   40  154  202  110 1102   59   99  169   15]
 [  17  263  234  308  243  124   10  614   82  105]]
```

Figure 15: For USPS

Figure 16: Confusion Matrix for Random Forest

6 Combination of all Classifiers :

All the above implemented classifiers with their maximum possible hyper-parameters, the target is predicted for both MNIST and USPS dataset. All the four predicted target is considered and one final prediction of target vector is done using **majority voting**.

Accuracy for MNIST test data with majority voting of classifiers: 93.91000000000001
Accuracy for USPS data with majority voting of classifiers: 38.666933346667335

7 Result and Inference

1. The accuracy for all the classifier and the dataset is calculated from the confusion matrix by dividing the sum in the diagonal by the sum in the entire matrix.

```
Confusion Matrix for MNIST test data with majority voting of classifiers:

[[ 970    0    0    1    0    3    3    1    2    0]
 [   0 1123    2    2    0    1    3    1    3    0]
 [  13    7  952    8    8    0   14   11   17    2]
 [   3    1   16  961    0    6    1    9   10    3]
 [   1    2    4    0  943    0    7    1    2   22]
 [  10    5    3   48   11  786   11    3   10    5]
 [  16    3    4    1    8   10  916    0    0    0]
 [   5   16   21    3    8    0    0  958    4   13]
 [  10    9    7   22    7   18   13    9  870    9]
 [  14    7    3   13   37    3    0   16    4  912]]
```

Figure 17: For MNIST

```
Confusion Matrix for USPS data with Random Forest:

[[ 651   13  259   48  455  154   66   93    1  260]
 [  43  563  113  107   58   96   20  983   16    1]
 [  95   24 1282   57   53  192   22  269    4    1]
 [  40    7   89 1294   57  306    3  183    3   18]
 [  13  208   43   20 1091  177   15  389   26   18]
 [ 147   29  133   65   25 1459   23  108    7    4]
 [ 295   45  220   19   87  345  844  132    1   12]
 [  42  328  365  255   41  242   34  682    1   10]
 [  50   40  154  202  110 1102   59   99  169   15]
 [  17  263  234  308  243  124   10  614   82  105]]
```

Figure 18: For USPS

Figure 19: Confusion Matrix

2. Do your results support the "No Free Lunch" theorem? Yes. The results supports the No Free lunch theorem which states that there is no clear model/classifier for all the problems/datasets. For the above four models we implemented we can obviously see that none of the model gave better results for both MNIST and the USPS dataset.
3. Which classifier has the overall best performance? The main factor we have to consider on that would be how the classifier performed for the unseen dataset - USPS dataset. SVM, Neural Network and Random Forest all the three on particular setting for this dataset performed almost the same. So, we cannot conclude on one particular classifier as a better performer.
4. Is the overall combined performance better than that of any individual classifier? From the test accuracy we got for both MNIST and USPS datasets, the combined performance of all the four classifiers gave few percentage of accuracy more than the individual classifiers.