Handwritten Digit Classification
MNIST and USPS

Anunay Rao
anunayra@buffalo.edu

1 Introduction

The project is to apply the different machine learning methods for the task of classification namely, Logistic Regression, Neural Network, Support Vector Machine and Random Forest. Further. We have to create an ensemble of these four classifiers i.e combine all the models and then by majority voting we have to make the final decision. Here, we will train our model on MNIST dataset and then test it on MNIST test set and as well as on USPS test set.

2 Pre-processing of the Dataset

Initially we have been provided with two datasets namely, MNIST Dataset and USPS Dataset. As these datasets are composed of images we have to process them to get the features.

2.1 MNIST Dataset

This dataset consists of grayscale images of digits from 0-9 of size 28x28. The grayscale image has the pixel value from 0 to 255 where 0 corresponds to the darkest and 255 corresponds to the brightest. Thus, taking the pixel value as features we will have 28x28 = 784 features.

2.1 USPS Dataset

This dataset also contains images of digits 0-9 of different sizes which will give different number of features if we consider the pixel values. Therefore, we have to first resize the image to 28x28 and then take pixel value. We have to normalize the pixel value so that all the values are between 0 and 255.

3 Performance Metric

We will evaluate the performance of these two models by accuracy which is defined as:

$$\boldsymbol{Accuracy = \frac{right}{right + wrong} \times 100}$$

4 Hyper-parameters values and Results:

4.1 Logistic Regression

For Logistic Regression since it is a 10-class problem we need to have one-hot encoding to represent the target class. Therefore we will convert the set of target values to the set of one-hot vectors.

Training Samples = 50000
Validation Samples = 10000
Testing Samples MNIST = 10000
Testing Samples USPS = 20000
learning rate: 0.003

4.1.1 Results and Confusion Matrix on MNIST Dataset

Training Accuracy = 92.28
Validation Accuracy = 92.56
Testing Accuracy = 92.01
Confusion Matrix:

```
[ 954    0    1    3    0    5   10    3    4    0]
[    0 1111    2    2    0    2    4    2   12    0]
[    5   10  909   24    7    5   13   11   40    8]
[    2    0   15  933    0   25    2   11   16    6]
[    0    3    6    2  899    1   12    5    9   45]
[    8    2    2   40    6  778   14    8   27    7]
[    8    3    3    2    7   20  910    2    3    0]
[    1    6   21    9    5    1    0  951    4   30]
[    4    8    4   39    8   42    9   12  841    7]
[    7    7    2   12   23   13    0   26    4  915]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

4.1.2 Results and Confusion Matrix on USPS Dataset

Testing Accuracy = 33.44
Confusion Matrix:

```
[ 469    1  183  129   91  360   68  209  147  343]
[   87  288  276  204  177  172   18  554  201   23]
[ 102   16 1237  163   24  245   65   46   69   32]
[   36    3  202 1085    5  537    5   59   50   18]
[   44   31   55   55  770  175   39  381  283  167]
[   78    9  233  216   19 1254   60   62   51   18]
[ 131    3  588   85   46  437  634   17   14   45]
[ 127  102   99  667   44  133   10  462  292   64]
[ 199   15  117  373   65  712   93   75  291   60]
[   20   60   94  563   74  102   12  562  315  198]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

4.2 Mini-Batch Stochastic Gradient Descent – Logistic Regression

Epochs = 25
Batch size = 50

Results:

Testing Accuracy on MNIST = 90.33
Testing Accuracy on USPS = 35.16

Confusion Matrix on MNIST:

```
[ 956    0    3    2    0    2    9    1    7    0]
[   0 1103    2    4    1    2    4    0   19    0]
[  11    6  889   18   15    0   17   21   45   10]
[   5    0   17  905    1   28    4   15   24   11]
[   1    5    5    1  904    0   11    2    8   45]
[  15    5    6   44   14  729   16   10   44    9]
[  16    3    5    2   12   15  899    1    5    0]
[   3   19   28    4   11    0    0  922    4   37]
[   9    9    9   31    8   24   13   13  844   14]
[  10    8    6   11   44   14    0   27    7  882]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

Confusion Matrix on USPS:

```
[ 595    4  357   59  250  122  101   44  159  309]
[ 228  303  130  354  278   54   40  307  289   17]
[ 209   25 1181  143   65   78   95   90   91   22]
[ 106    3  118 1283   19  233   29   59   98   52]
[  62   81   41   63 1017  123   39  130  297  147]
[ 174   20  211  189   45 1042  126   71   87   35]
[ 364   12  357  112  103  224  698   23   72   35]
[ 197  212  312  464   72   78   35  299  284   47]
[ 226   30  144  213  128  571  118   44  446   80]
[  44  184  161  483  149   88   15  366  342  168]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

4.3 Neural Network:

Training Samples = 50000
Validation Samples = 10000
Testing Samples MNIST = 10000
Testing Samples USPS = 20000
input_size = 784
drop_out = 0.2
first_dense_layer_nodes = 512
second_dense_layer_nodes = 256
third_dense_layer_nodes =10
Activation function first layer = ReLu
Activation function second layer = ReLu
Activation function third layer = softmax
Optimizer = rmsprop
Loss = categorical_crossentropy
model_batch_size = 128
Number of Epochs = 25

4.3.1 Results and Confusion Matrix on MNIST Dataset

Results:

Training Accuracy = 99.87
Validation Accuracy = 98.13
Testing Accuracy = 98.24

Confusion Matrix:

```
[ 972    1    0    1    0    2    2    1    1    0]
[   0 1130    2    0    0    1    2    0    0    0]
[   4    2 1007    4    1    0    2    7    5    0]
[   0    0    3  992    0    7    0    3    2    3]
[   3    0    2    0  951    0    7    1    2   16]
[   2    0    0    5    0  877    3    0    3    2]
[   2    3    0    0    2    7  943    0    0    1]
[   1    8    9    1    1    0    0  998    2    8]
[   0    1    1    5    1    7    2    2  951    4]
[   0    2    0    6    4    6    1    2    3  985]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

4.3.2 Results and Confusion Matrix on USPS Dataset

Testing Accuracy = 42.91

Confusion Matrix:

```
[ 390    3  283   42  159  207  435  179   94  208]
[  49  470  534  110  301   89   37  214  115   81]
[  82    4 1537   33   14  100  146   34   44    5]
[  22    1  482 1039    6  332   38   19   41   20]
[   9   70  102   13 1140  145   73  250  154   44]
[  14    0  367   52    2 1314  183   18   40   10]
[  68   10  369    5   17  102 1205  120   15   89]
[  15  239  334  297   53   56   56  772  169    9]
[  76   16  291  280   59  398  227  145  483   25]
[   2  111  177  219  168   29   29  760  273  232]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

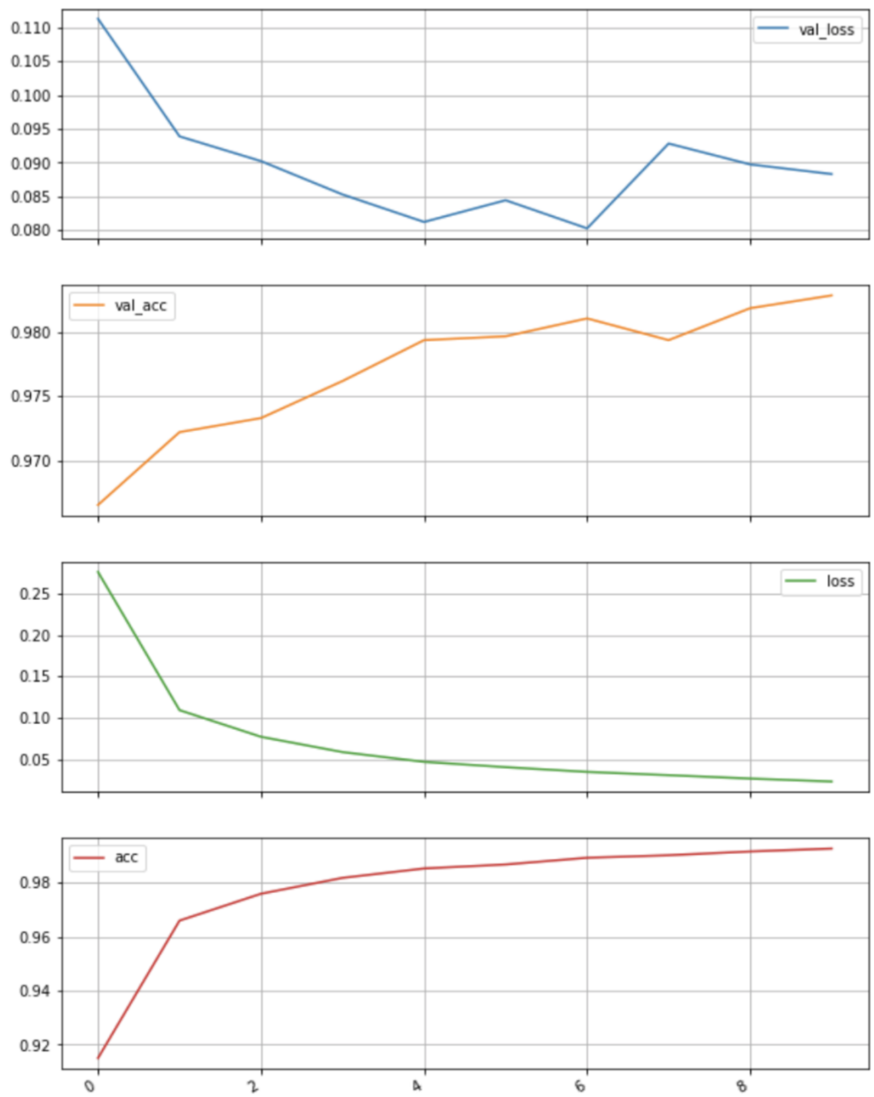

Figure 1:Showing Validation loss, Validation Accuracy, Training Loss and Training Accuracy (top to bottom) against number of epochs

4.4 Support Vector Machine

Training Samples = 60000
Testing Samples MNIST = 10000
Testing Samples USPS = 20000

4.4.1 Support Vector Machine using Linear Kernel

Results:
Testing Accuracy on MNIST = 91.78
Testing Accuracy on USPS = 26.71

Confusion Matrix on MNIST:

```
[  961    0    2    1    1    4    6    3    1    1]
[    0 1112    3    2    0    1    5    1   11    0]
[   11   11  911   18   10    4   13   12   39    3]
[    4    0   19  918    2   22    5   12   20    8]
[    1    4    5    4  913    0    9    3    5   38]
[    9    2    0   39   12  767   18    7   30    8]
[    7    4    7    2    5   21  909    1    2    0]
[    2    8   23    5    7    1    1  948    5   28]
[   11   13    8   20   14   31    8   13  843   13]
[    7    8    2   15   31   12    0   26   12  896]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

Confusion Matrix on USPS:

```
[  381    1  348  233   51  161  111  572   60   82]
[   46  280  658  158  362   96   28  284   67   21]
[   75   56 1243  104   38  202  155   86   20   20]
[   46   34  423  753   19  527   37   89   41   31]
[   64   52  176  120  556  183   67  604  138   40]
[   49   27  752  199   20  716   80  125   24    8]
[   86    8  698  106   51  392  507   85   17   50]
[  149   95  235  447   92  136   28  694   95   29]
[  207   28  155  619  121  371  104  238  117   40]
[   48   56  140  524  101   80   11  768  176   96]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

4.4.2 Support Vector Machine using rbf Kernel with Gamma=1 and keeping other parameters as default

Results:
Testing Accuracy on MNIST = 17.59
Testing Accuracy on USPS = 26.13

Confusion Matrix on MNIST:

```
[    0    0    0    0    0    0    0  980    0    0]
[    0  731    0    0    0    0    0  404    0    0]
[    0    0    0    0    0    0    0 1032    0    0]
[    0    0    0    0    0    0    0 1010    0    0]
[    0    0    0    0    0    0    0  982    0    0]
[    0    0    0    0    0    0    0  892    0    0]
[    0    0    0    0    0    0    0  958    0    0]
[    0    0    0    0    0    0    0 1028    0    0]
[    0    0    0    0    0    0    0  974    0    0]
[    0    0    0    0    0    0    0 1009    0    0]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

Confusion Matrix on USPS:

```
[    0    0    0    0    0    0    0 2000    0    0]
[    0    0    0    0    0    0    0 2000    0    0]
[    0    0    0    0    0    0    0 1999    0    0]
[    0    0    0    0    0    0    0 2000    0    0]
[    0    0    0    0    0    0    0 2000    0    0]
[    0    0    0    0    0    0    0 2000    0    0]
[    0    0    0    0    0    0    0 2000    0    0]
[    0    0    0    0    0    0    0 2000    0    0]
[    0    0    0    0    0    0    0 2000    0    0]
[    0    0    0    0    0    0    0 2000    0    0]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

4.4.3 SVM using rbf kernel with Gamma=auto (default)

Results:

Testing Accuracy on MNIST = 94.35
Testing Accuracy on USPS = 38.54

Confusion Matrix on MNIST:

```
[  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]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

Confusion Matrix on USPS:

[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]

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

4.4.4 SVM using rbf kernel with Gamma=0.05 and C=5

Results:

Testing Accuracy on MNIST = 98.28
Testing Accuracy on USPS = 26.13

Confusion Matrix on MNIST:

[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	996	0	4	0	5	4	0]
[0	1	3	0	965	0	4	0	2	7]
[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	6	2	0	6	6	971]

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

Confusion Matrix on USPS:

```
[ 226    0 1564    2   26   35    2    0   79   66]
[  78  257  712  173  264   77   12  335   88    4]
[   8    0 1944    6    3   20    1    6   11    0]
[   4    0 1195  725    0   41    0    0   35    0]
[   6    0 1045   18  521   96    0   57  252    5]
[  15    0 1305   17    1  625    0    0   37    0]
[  78    0 1534    2   10   61  290    0   22    3]
[  17    6 1433  129    6  134    0  222   52    1]
[   7    0 1387   14    4  221    0    0  367    0]
[   1    0 1510   79   26   29    0   39  266   50]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

4.5 Random Forest

n_estimators = 10

Results:

Testing Accuracy on MNIST = 94.60
Testing Accuracy on USPS = 39.67

Confusion Matrix on MNIST:

```
[ 967    0    0    2    0    2    5    2    2    0]
[   0 1121    5    2    0    1    2    0    4    0]
[   8    3  985    8    3    0    2   11   11    1]
[   1    0   18  940    2   16    0   12   18    3]
[   3    1    4    3  933    0    8    4    4   22]
[   8    4    4   36    8  812    5    3    8    4]
[   6    3    2    0    8   10  925    0    4    0]
[   4    8   21   10    6    1    0  963    3   12]
[   7    2   14   20   13   12    7    5  888    6]
[   6    9    7   16   19   10    2    9    5  926]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

Confusion Matrix on USPS:

```
[664  54 304 113 352 125  80 134  14 160]
[ 78 494 168 110 218  95  37 747  20  33]
[259 109 931 116  99 183  80 176  31  15]
[123  48 190 926  91 386  31 121  24  60]
[ 33 216 116  90 951 154  41 305  36  58]
[285  88 168 239  68 913  67 119  21  32]
[418  91 338 107 160 288 460  86  27  25]
[140 410 304 256  76 191  45 541  14  23]
[187 125 287 239 159 637  96  92 133  45]
[ 80 282 313 300 245 141  37 421  68 113]
```

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

4.6 Ensemble Classifier

This is the combination of the above models namely, Logistic Regression, SVM using Linear Kernel, Neural Network and Random Forest using Majority Voting.

Results:

Testing Accuracy on MNIST = 95.30
Testing Accuracy on USPS = 36.60

Confusion Matrix on MNIST:

971	1	1	0	0	2	2	1	2	0
0	1127	3	1	0	1	2	1	0	0
7	7	990	5	2	2	3	8	7	1
2	0	17	974	0	7	0	4	5	1
1	2	3	2	950	0	5	1	3	15
6	1	1	34	8	824	9	0	8	1
8	3	2	2	6	16	921	0	0	0
2	6	22	5	6	1	0	974	2	10
6	7	6	23	10	24	9	10	873	6
7	7	3	14	22	7	0	16	6	927

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

Confusion Matrix on USPS:

610	11	354	127	152	180	93	240	57	176
78	433	515	148	266	110	15	362	65	8
119	28	1463	73	16	151	75	39	25	10
52	6	413	1054	4	374	18	42	23	14
47	114	136	60	1008	135	31	290	137	42
101	25	450	176	4	1132	50	38	18	6
189	26	623	58	44	308	665	40	7	40
162	253	229	450	37	108	21	608	121	11
234	37	242	431	66	554	92	107	208	29
39	144	182	455	90	57	7	672	215	139

where C[i,j] is equal to the number of observations known to be in class i but predicted to be in class j.

5 Questions to be Answered

5.1 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?

Answer: No Free Lunch Theorem states that no single Machine learning classification algorithm can be universally better than any other one on all domains. In simple words, it means that no algorithm is universally best for every problem.

In our results, we are getting higher testing accuracy for MNIST test set but much lower accuracy for USPS data set which means that our model is not performing well on USPS dataset and so the results supports No Free Lunch theorem.

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

Answer: Confusion Matrix on MNIST test set for the Logistic Regression, Neural Network, Random Forest, SVM using linear kernel, rbf kernel with gamma=auto, rbf kernel with gamma=0.05 and C=5 clearly classifies the test set to the Actual class with high accuracy.

Further on MNIST dataset, in case of SVM using rbf kernel with gamma=1 and as the gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. With the first two target values in MNIST training set being 7 and 1 the model classifies 92.69% samples into class7 (digit 7) and remaining into class1 (digit 1). In case of USPS dataset SVM using rbf kernel with gamma=1 classifies all the test samples to class7 (digit 7) which is quite expected as 7 being the first target value in MNIST training set.

Logistic Regression: Output has probabilistic interpretation plus it can be regularized to avoid overfitting. It is very fast and gives a bit lower accuracy than other models. Its weakness is that although it is fast it tends to underperform when compared to other alternatives.

Neural Network: Since here we have a large dataset neural network is a good choice. It gives very high accuracy and takes less time to train when compared to SVM.

Random Forest: It is easy to tune and gives high accuracy a little less than SVM and Neural Network but it is much faster than SVM and Neural Network.

Support Vector Machine: These are trickier to tune due to choosing right kernel. It takes a lot of time to train the model among all the methods. It gives very high accuracy if the model is tuned well. As in this case model with gamma = 0.05 and C =5 gives very high accuracy.

Overall considering the test accuracy Neural Network and SVM performed equally well on test set but taking the training time into account Neural Network performed the best.

5.3 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?

Answer: By combining the results of individual classifier using majority voting the overall combined performance is better than the individual performance of Logistic Regression, SVM using Linear Kernel and Random Forest.

6 Conclusion:

	Test Accuracy MNIST test set	Test Accuracay USPS dataset
Logisitic Regression	92.01	33.44
Neural Network	98.24	35.16
SVM using Linear Kernel	91.78	26.71

SVM using rbf kernel (gamma=1)	17.59	26.13
SVM using rbf kernel (gamma = auto)	94.35	38.54
SVM using rbf kernel (gamma=0.05 and C=5)	98.28	26.13
Random Forest	94.60	39.67
Ensemble Classifier	95.30	36.60

Table 1: Testing Accuracy values for different models with different configuration

References

[1] Towards Data Science. (2018). Machine Learning – Towards Data Science. [online] Available at: https://towardsdatascience.com/machine-learning/home

[2] Brownlee, J. (2018). Machine Learning Mastery. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/