Project 3: Classification

Sahil Suhas Pathak
Department of Computer Science and Engineering
University at Buffalo
Buffalo, NY, 14214
sahilsuh@buffalo.edu
UBITName: sahilsuh, Person Number: 50289739

Abstract

This project is to implement machine learning methods for the task of classification and also an ensemble of four classifiers for the given task. Training is done using the following four classifiers on MNIST digit images. Testing is performed on USPS Dataset.

1. Softmax Logistic regression.
2. Multilayer Perceptron Neural Network using Keras.
3. Support Vector Machine
4. Random Forest

1 Softmax Logistic Regression

Also called as Multinomial Logistic, is used when we have to categorize the samples into more than two classes, k-classes.

1.1 Steps for Calculation

1. We set the number of epochs to iterate over.

2. Calculate net_in = $W^T X$

3. Calculate z = softmax(net_in) where softmax is defined by $y_k(x) = \frac{\exp(a_k)}{\sum \exp(a_k)}$

4. Calculate the cross-entropy error function, $Cost = -\sum t_k ln y_k$

5. Calculate the gradient of the error, $grad = (y_i - t_i)x$

6. Update weights, $w_j^{\tau+1} = w_j - \eta^*\text{grad}$

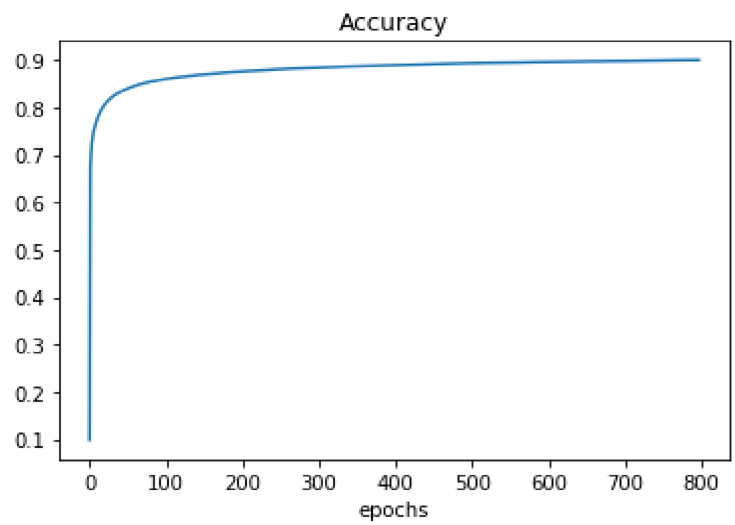
7. Evaluate Training Accuracy, Testing Accuracy on MNIST and USPS Dataset

1.2 Observations

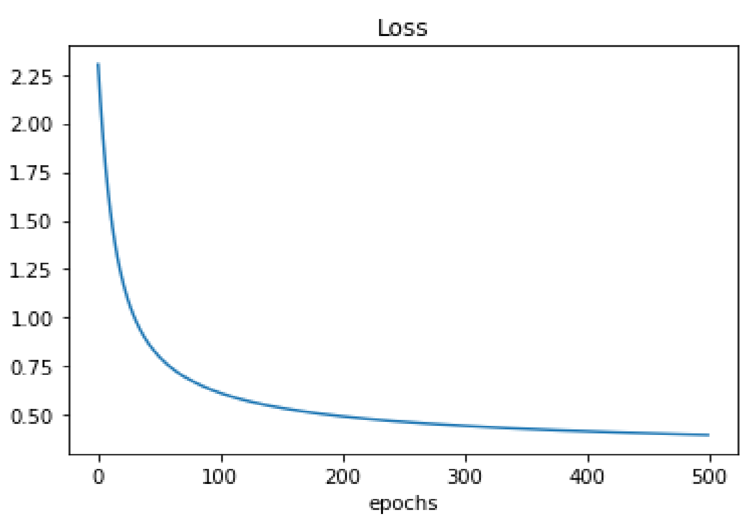
Epochs	Training Accuracy(MNIST)	Testing Accuracy(MNIST)	Testing Accuracy(USPS)
10	0.7710333333333333	0.7889	0.28216410820541027
50	0.8394166666666667	0.8497	0.28216410820541027
100	0.859916666666666	0.8698	0.3313165658282914
200	0.8757666666666667	0.8845	0.3399169958497925
500	0.8936166666666666	0.8998	0.35061753087654385
800	0.9000833333333333	0.9059	0.3546177308865443

Let us consider the Observation where Epochs are 800,

Accuracy V/s Epochs



Loss V/s Epochs



Confusion Matrix for MNIST Testing Dataset:

```
[ 957.    0.    3.    2.    0.    3.   10.    1.    4.    0.]
[   0. 1103.    2.    4.    1.    2.    4.    1.   18.    0.]
[  11.    6.  896.   16.   15.    0.   15.   20.   45.    8.]
[   5.    0.   18.  904.    1.   32.    3.   15.   21.   11.]
[   1.    4.    5.    1.  907.    0.   11.    1.    8.   44.]
[  14.    5.    4.   41.   13.  740.   16.   10.   41.    8.]
[  16.    3.    5.    2.   13.   16.  898.    1.    4.    0.]
[   3.   19.   28.    4.   11.    0.    0.  923.    3.   37.]
[   9.    9.    9.   27.    8.   27.   13.   15.  843.   14.]
[  10.    7.    5.   10.   43.   13.    0.   25.    8.  888.]
```

Confusion Matrix for USPS Testing Dataset:

```
[ 596.    4.  346.   59.  248.  136.   93.   43.  163.  312.]
[ 222.  307.  136.  347.  278.   58.   40.  321.  273.   18.]
[ 206.   25. 1183.  146.   63.   84.   94.   87.   88.   23.]
[ 101.    3.  119. 1296.   18.  244.   25.   62.   87.   45.]
[  56.   77.   39.   62. 1017.  121.   42.  133.  295.  158.]
[ 169.   20.  212.  176.   42. 1083.  116.   68.   84.   30.]
[ 352.   13.  357.  107.  100.  256.  690.   23.   67.   35.]
[ 195.  210.  311.  453.   74.   85.   32.  308.  282.   50.]
[ 220.   30.  149.  209.  125.  592.  109.   47.  438.   81.]
[  40.  178.  160.  477.  147.   89.   12.  384.  339.  174.]
```

2 Deep Neural Network

Observation No. 1:

Model:

Optimizer: sgd

sgd = optimizers.SGD(lr=0.1, decay=1e-6, momentum=0.9, nesterov=True)

Layer (type)	Output Shape	Param #
dense_1 (Dense) (Activation: sigmoid)	(None, 32)	25120
dense_2 (Dense) (Activation: softmax)	(None, 10)	330

Total params: 25,450
Trainable params: 25,450
Non-trainable params: 0

Training:

Batch Size: 128, Epochs: 50, Validation Split: 0.16666 (Total 10000 Validation Samples)

No early stopping implemented
Training Loss: 0.0275, Training Accuracy: 0.9945
Validation Loss: 0.1210, Validation Accuracy: 0.9682

Testing Accuracy on MNIST Dataset: 0.9655
Testing Accuracy on USPS Dataset: 0.415370768530976

Observation No. 2:

Model:

Optimizer: sgd

sgd = optimizers.SGD(lr=0.1, decay=1e-6, momentum=0.9, nesterov=True)

Layer (type)	Output Shape	Param #
dense_57 (Dense) (Activation: relu)	(None, 32)	25120
dense_58 (Dense) (Activation: relu)	(None, 128)	4224
dense_59 (Dense) (Activation: softmax)	(None, 10)	1290

Total params: 30,634

Trainable params: 30,634

Non-trainable params: 0

Training:

Batch Size: 128, Epochs: 50, Validation Split: 0.16666 (Total 10000 Validation Samples)

Early stopping implemented

Early-patience: 10, Monitoring: val_loss

Training Loss: 0.0404, Training Accuracy: 0.9870,

Validation Loss: 0.1469, Validation Accuracy: 0.9692

Epoch 00017: early stopping

Testing Accuracy on MNIST Dataset: 0.9699

Testing Accuracy on USPS Dataset: 0.4563728186290105

Observation No. 3:

Model:

Optimizer: sgd (Default Configuration)

Layer (type)	Output Shape	Param #
dense_72 (Dense) (Activation: relu)	(None, 32)	25120
dense_73 (Dense) (Activation: sigmoid)	(None, 256)	8448
dense_74 (Dense) (Activation: softmax)	(None, 10)	2570

Total params: 36,138

Trainable params: 36,138

Non-trainable params: 0

Training:

Batch Size: 128, Epochs: 50, Validation Split: 0.16666 (Total 10000 Validation Samples)

Early stopping implemented

Early patience: 10, Monitoring: Validation Loss

Testing Accuracy on MNIST Dataset: 0.9451

Testing Accuracy on USPS Dataset: 0.38

Observation No. 4:

Model:

Optimizer: adam (Default Configuration)

Layer (type)	Output Shape	Param #
dense_75 (Dense) (Activation: sigmoid)	(None, 32)	25120
dense_76 (Dense) (Activation: relu)	(None, 256)	8448
dense_77 (Dense) (Activation: softmax)	(None, 10)	2570

Total params: 36,138

Trainable params: 36,138

Non-trainable params: 0

Training:

Batch size: 128, Epochs: 500, Validation Split: 0.16666

Early stopping implemented

Early patience: 10, Monitoring: Validation Loss

Training Loss: 0.0703, Training Accuracy: 0.9801

Validation Loss: 0.1202, Validation Accuracy: 0.9685

Epoch 00381: early stopping

Testing Accuracy on MNIST Dataset: 0.9657

Testing Accuracy on USPS Dataset: 0.39841992099306944

Observation No. 5:

Model:

Optimizer: sgd

sgd = optimizers.SGD(lr=0.1, decay=1e-6, momentum=0.9, nesterov=True)

Layer (type)	Output Shape	Param #
dense_81 (Dense) (Activation: relu)	(None, 32)	25120
dense_82 (Dense) (Activation: relu)	(None, 256)	8448
dense_83 (Dense) (Activation: sigmoid)	(None, 128)	32896
dense_84 (Dense) (Activation: softmax)	(None, 10)	1290

Total params: 67,754

Trainable params: 67,754

Non-trainable params: 0

Training:

Batch Size: 256, Epochs: 500, Validation Split: 0.16666

Early stopping implemented

early_patience: 10, Monitoring: Validation Loss

Training Loss: 0.0152, Training Accuracy: 0.9957

Validation Loss: 0.0977, Validation Accuracy: 0.9747

Epoch 00023: early stopping

Testing Accuracy on MNIST Dataset: 0.9746

Testing Accuracy on USPS Dataset: 0.4565228261338561

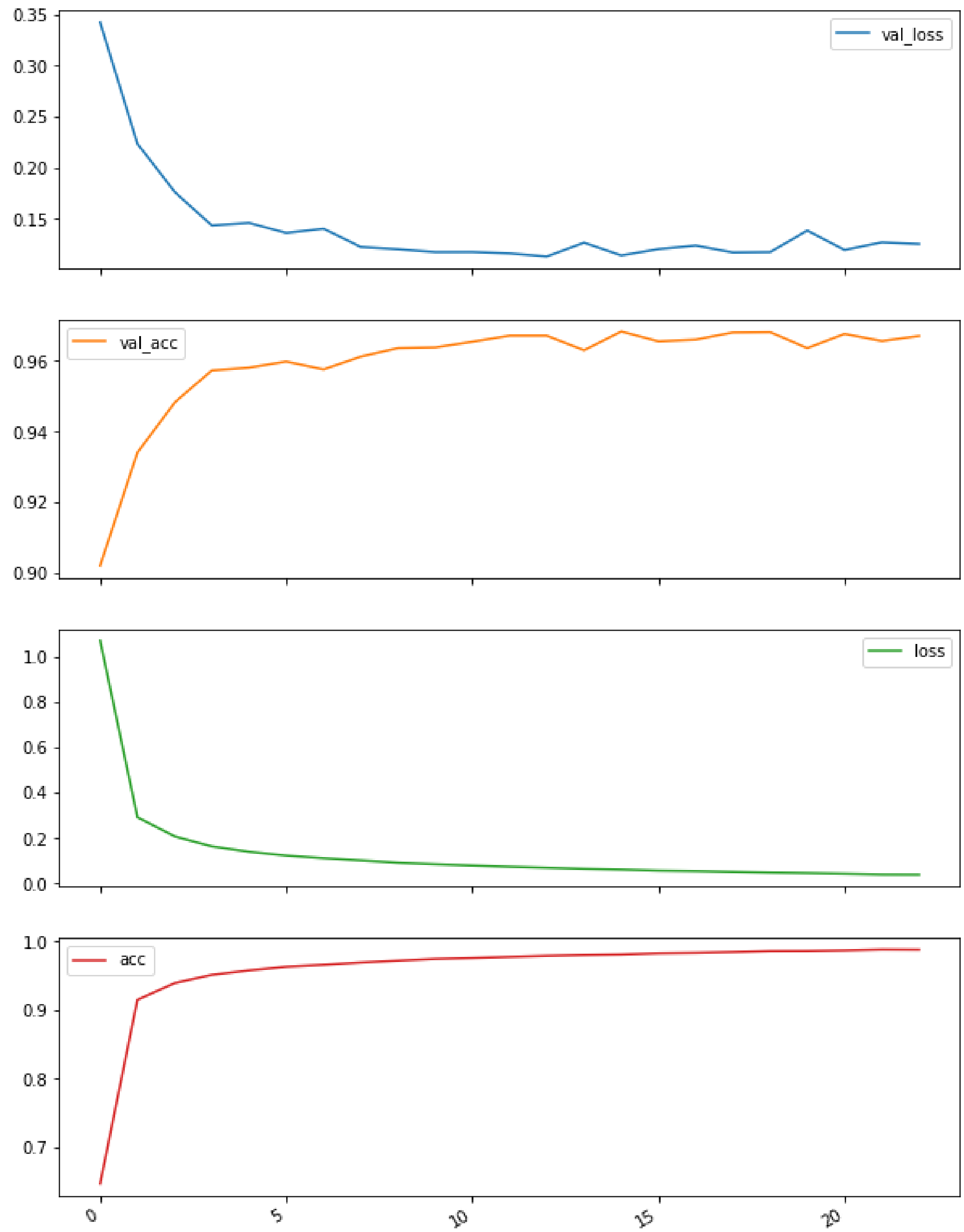


Figure 1. Validation Loss, Validation Accuracy, Training Loss and Training Accuracy for Observation No. 5

3 SVM

1. Using linear kernel (all other parameters are kept default)

SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='auto', kernel='linear', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)

Testing Accuracy on MNIST Dataset: 0.9403

Confusion Matrix for MNIST Testing Dataset:

```
[ 957.    0.    4.    1.    1.    6.    9.    1.    0.    1.]
[   0. 1122.    3.    2.    0.    1.    2.    1.    4.    0.]
[   8.    6.  967.   11.    3.    3.    7.    8.   17.    2.]
[   4.    3.   16.  947.    1.   15.    0.    9.   13.    2.]
[   1.    1.   10.    1.  942.    2.    4.    2.    3.   16.]
[  10.    4.    3.   36.    6.  803.   13.    1.   14.    2.]
[   9.    2.   13.    1.    6.   16.  909.    1.    1.    0.]
[   1.    8.   21.   10.    8.    1.    0.  957.    3.   19.]
[   8.    4.    6.   25.    7.   26.    6.    7.  877.    8.]
[   7.    7.    2.   11.   33.    4.    0.   18.    5.  922.]
```

Testing Accuracy on USPS Dataset: 0.29236461823091153

Confusion Matrix for USPS Testing Dataset:

```
[ 368.    0.  503.  149.  161.  345.   67.  200.    9.  198.]
[  39.  300.  534.  214.  242.  212.   25.  371.   41.   22.]
[ 113.   62. 1288.   97.   35.  263.   67.   42.   21.   11.]
[  47.   60.  346.  832.   13.  610.    6.   48.   25.   13.]
[  21.   19.  242.   81.  769.  231.   16.  475.   74.   72.]
[  34.   13.  656.  215.   30.  949.   24.   35.   30.   14.]
[ 143.   17.  857.   50.   64.  341.  455.   32.    1.   40.]
[  26.   74.  220.  623.   37.  304.   11.  583.   95.   27.]
[ 110.   10.  344.  434.   84.  708.   68.   60.  152.   30.]
[  11.   34.  218.  576.  130.  115.    5.  617.  143.  151.]
```

2. Using radial basis function with value of gamma setting to 1 (all other parameters are kept default)

SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma=1, kernel='rbf', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)

Note: The following configuration performs poorly on both the Testing sets i.e. on MNIST and USPS Testing Dataset. The Training Accuracy is almost ~100% thus it is a case of overfitting where model tries to learn the training data so accurately that it fails to generalize when new data is fed to it.

Testing Accuracy on MNIST Dataset: 0.1731

Confusion Matrix for MNIST Testing Dataset:

```
[    0.     0.     0.     0.     0.     0.     0.  980.     0.     0.]
[    0.   703.     0.     0.     0.     0.     0.  432.     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.]
```

Testing Accuracy on USPS Dataset: 0.10000500025001251

Confusion Matrix for USPS Testing Dataset:

```
[  368.     0.   503.   149.   161.   345.    67.   200.     9.   198.]
[   39.   300.   534.   214.   242.   212.    25.   371.    41.    22.]
[  113.    62.  1288.    97.    35.   263.    67.    42.    21.    11.]
[   47.    60.   346.   832.    13.   610.     6.    48.    25.    13.]
[   21.    19.   242.    81.   769.   231.    16.   475.    74.    72.]
[   34.    13.   656.   215.    30.   949.    24.    35.    30.    14.]
[  143.    17.   857.    50.    64.   341.   455.    32.     1.    40.]
[   26.    74.   220.   623.    37.   304.    11.   583.    95.    27.]
[  110.    10.   344.   434.    84.   708.    68.    60.   152.    30.]
[   11.    34.   218.   576.   130.   115.     5.   617.   143.   151.]
```

3. Using radial basis function with value of gamma setting to default (all other parameters are kept default)

SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)

Testing accuracy MNIST Dataset: 0.9443

Confusion Matrix for MNIST Testing Dataset:

```
[  967.     0.     2.     0.     0.     5.     4.     1.     1.     0.]
[    0.  1121.     2.     2.     0.     1.     4.     1.     4.     0.]
[    9.     1.   961.     9.    10.     1.    13.     9.    17.     2.]
[    1.     1.    15.   951.     1.    15.     1.    10.    11.     4.]
[    1.     1.     7.     0.   938.     0.     7.     2.     2.    24.]
[    7.     4.     5.    32.     7.   808.    12.     2.    10.     5.]
[    9.     3.     4.     1.     5.     9.   926.     0.     1.     0.]
[    2.    13.    22.     5.     8.     1.     0.   955.     3.    19.]
[    4.     7.     7.    14.     8.    23.    10.     6.   892.     3.]
[    8.     7.     0.    12.    31.     6.     1.    13.     7.   924.]
```

Testing Accuracy on USPS Dataset: 0.38751937596879843

Confusion Matrix for USPS Testing Dataset:

```
[ 584.    2.  422.   18.  280.  257.   63.   51.    7.  316.]
[ 110.  432.  279.  131.  257.  172.   43.  540.   20.   16.]
[ 123.   17. 1400.   57.   37.  213.   60.   57.   22.   13.]
[  77.    3.  182. 1117.    9.  494.    5.   71.   27.   15.]
[  16.   59.   89.   12. 1165.  260.   21.  214.   71.   93.]
[ 104.   17.  248.  104.   20. 1393.   52.   40.   18.    4.]
[ 194.    6.  488.   24.   88.  411.  743.   11.    7.   28.]
[  45.  218.  438.  270.   57.  411.   14.  476.   49.   22.]
[  75.   24.  222.  186.   90. 1013.   87.   39.  242.   22.]
[  19.  164.  235.  284.  189.  158.    6.  523.  224.  198.]
```

4. Evaluating Testing Accuracy for MNIST and USPS Dataset by varying C values for configuration *(kernel = 'rbf' and gamma = 0.05)*

C	Testing Accuracy on MNIST	Testing Accuracy on USPS
1	0.9826	0.27096354817740886
2	0.9836	0.27376369318465925
3	0.9837	0.2738636881844092
4	0.9839	0.2738136906845342
5	0.984	0.2738136906845342
10	0.984	0.2738136906845342
20	0.984	0.2738136906845342

Challenges faced:

1. SVM with configuration as kernel = 'rbf' and gamma = 1 takes significant amount of time for training, it goes beyond 4hrs of computational time.

2. SVM with default configuration takes about 30 minutes to complete the training process, which is again a considerable amount of time to yield results.

Inference:

1. SVM gives a comparatively lower accuracy when tested on USPS Dataset when compared to the accuracy observed from a Neural Network Implementation but gives a better accuracy for MNIST Dataset.

4 Random Forest

1. Number of Trees: 500

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=500, n_jobs=1, oob_score=False, random_state=None, verbose=0, warm_start=False)

Testing Accuracy on MNIST: 0.9672

Confusion Matrix for MNIST Testing Dataset:

```
[ 969.    0.    0.    0.    0.    2.    3.    1.    4.    1.]
[   0. 1124.    2.    3.    0.    1.    3.    1.    1.    0.]
[   6.    0. 1002.    3.    3.    0.    4.    8.    6.    0.]
[   0.    0.    7.  977.    0.    6.    0.    9.    9.    2.]
[   1.    0.    2.    0.  957.    0.    4.    0.    2.   16.]
[   2.    0.    0.   12.    3.  862.    6.    1.    4.    2.]
[   6.    3.    0.    0.    2.    3.  940.    0.    4.    0.]
[   1.    2.   17.    0.    1.    0.    0.  996.    2.    9.]
[   4.    0.    5.    7.    3.    5.    4.    4.  932.   10.]
[   5.    5.    2.   10.   10.    3.    1.    4.    5.  964.]
```

Testing Accuracy on USPS Dataset: 0.40917045852292616

Confusion Matrix for USPS Testing Dataset:

```
[ 661.    8.  257.   56.  452.  160.   61.   98.    2.  245.]
[  43.  564.  113.  106.   51.   93.   28.  989.   12.    1.]
[  91.   27. 1264.   65.   53.  225.   15.  254.    4.    1.]
[  38.    7.   96. 1286.   56.  339.    4.  154.    4.   16.]
[  11.  193.   56.   23. 1094.  182.   11.  383.   24.   23.]
[ 143.   28.  124.   53.   23. 1501.   17.   98.    6.    7.]
[ 299.   46.  209.   16.   86.  369.  831.  130.    3.   11.]
[  35.  321.  357.  229.   43.  277.   28.  700.    1.    9.]
[  36.   44.  138.  194.   99. 1153.   61.   88.  170.   17.]
[  18.  260.  217.  293.  246.  125.    7.  634.   88.  112.]
```

2. Number of Trees: 700

RandomForestClassifier (bootstrap=True, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=700, n_jobs=1, oob_score=False, random_state=None, verbose=0, warm_start=False)

Testing Accuracy on MNIST Dataset: 0.9711

Confusion Matrix for MNIST Testing Dataset:

```
[ 971.    0.    1.    0.    0.    2.    2.    1.    3.    0.]
[   0. 1123.    3.    3.    0.    2.    2.    0.    1.    1.]
[   6.    0. 1001.    5.    2.    0.    4.    8.    6.    0.]
[   1.    0.    8.  975.    0.    6.    0.    9.    8.    3.]
[   1.    0.    1.    0.  955.    0.    5.    1.    2.   17.]
[   3.    0.    0.   10.    3.  862.    6.    2.    4.    2.]
[   6.    3.    0.    0.    2.    3.  940.    0.    4.    0.]
[   1.    3.   17.    1.    1.    0.    0.  991.    2.   12.]
[   3.    0.    5.    7.    4.    5.    3.    4.  933.   10.]
[   7.    5.    1.    9.   13.    3.    1.    5.    5.  960.]
```

Testing Accuracy on USPS Dataset: 0.40817040852042

Confusion Matrix for USPS Testing Dataset:

```
[ 654.   13.  272.   55.  452.  159.   61.   97.    1.  236.]
[  37.  561.  111.   98.   52.  105.   20. 1002.   13.    1.]
[  74.   28. 1268.   69.   54.  224.   14.  262.    5.    1.]
[  36.    6.   82. 1313.   48.  325.    2.  165.    5.   18.]
[  14.  199.   49.   21. 1075.  191.   13.  389.   28.   21.]
[ 133.   23.  131.   67.   22. 1494.   16.  100.    9.    5.]
[ 309.   43.  218.   24.   82.  364.  822.  124.    4.   10.]
[  32.  310.  385.  233.   34.  267.   26.  701.    4.    8.]
[  41.   33.  150.  204.   92. 1138.   58.   93.  174.   17.]
[  15.  261.  223.  310.  239.  137.    9.  618.   87.  101.]
```

3. Number of Trees: 1000

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=1000, n_jobs=1, oob_score=False, random_state=None, verbose=0, warm_start=False)

Testing Accuracy on MNIST Dataset: 0.972

Confusion Matrix for MNIST Testing Dataset:

```
[ 970.    0.    1.    0.    0.    2.    3.    1.    3.    0.]
[   0. 1124.    2.    3.    0.    2.    2.    0.    1.    1.]
[   6.    0. 1003.    3.    2.    0.    4.    8.    6.    0.]
[   0.    0.    9.  976.    0.    5.    0.    9.    8.    3.]
[   1.    0.    1.    0.  957.    0.    5.    0.    2.   16.]
[   2.    0.    0.   11.    3.  861.    6.    2.    5.    2.]
[   5.    3.    0.    0.    2.    3.  941.    0.    4.    0.]
[   1.    2.   17.    1.    1.    0.    0.  992.    2.   12.]
[   4.    0.    7.    7.    2.    5.    3.    4.  932.   10.]
[   5.    5.    3.   10.    9.    2.    1.    5.    5.  964.]
```

Testing Accuracy on USPS Dataset: 0.4097704885244262

Confusion Matrix for USPS Testing Dataset:

```
[ 647.   11.  261.   58.  445.  166.   61.  103.    1.  247.]
[  36.  545.  116.  101.   46.  100.   21. 1021.   13.    1.]
[  86.   28. 1278.   66.   48.  209.   16.  262.    4.    2.]
[  33.    3.   87. 1299.   51.  326.    4.  177.    5.   15.]
[  11.  190.   50.   21. 1088.  191.   15.  397.   19.   18.]
[ 128.   24.  127.   58.   24. 1507.   13.  107.    7.    5.]
[ 292.   44.  212.   18.   84.  361.  838.  138.    2.   11.]
[  39.  305.  364.  242.   33.  268.   32.  706.    3.    8.]
[  40.   31.  138.  204.   92. 1150.   63.   95.  173.   14.]
[  16.  257.  223.  301.  222.  133.    7.  634.   93.  114.]
```

4. Number of Trees: 2000

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=2000, n_jobs=1, oob_score=False, random_state=None, verbose=0, warm_start=False)

Testing Accuracy on MNIST Dataset: 0.9723

Confusion Matrix for MNIST Testing Dataset:

```
[ 971.    0.    0.    0.    0.    2.    2.    1.    3.    1.]
[   0. 1123.    3.    3.    0.    2.    2.    0.    1.    1.]
[   6.    0. 1000.    5.    3.    0.    4.    8.    6.    0.]
[   0.    0.    9.  977.    0.    5.    0.    9.    8.    2.]
[   1.    0.    2.    0.  957.    0.    4.    0.    2.   16.]
[   2.    0.    0.   10.    3.  862.    6.    1.    5.    3.]
[   5.    3.    0.    0.    3.    3.  940.    0.    4.    0.]
[   1.    2.   17.    0.    1.    0.    0.  996.    1.   10.]
[   3.    0.    5.    6.    3.    6.    3.    4.  934.   10.]
[   5.    5.    2.   11.   10.    3.    1.    4.    5.  963.]
```

Testing Accuracy on USPS Dataset: 0.41142057102855145

Confusion Matrix for USPS Testing Dataset:

```
[ 650.   12.  265.   50.  453.  179.   51.   93.    0.  247.]
[  40.  560.  121.  101.   47.   99.   20.  998.   13.    1.]
[  89.   24. 1286.   61.   49.  215.   14.  255.    4.    2.]
[  36.    6.   83. 1313.   55.  316.    3.  170.    4.   14.]
[  13.  197.   52.   23. 1098.  183.   14.  381.   24.   15.]
[ 134.   26.  123.   59.   22. 1508.   18.  100.    6.    4.]
[ 295.   45.  195.   21.   81.  383.  845.  127.    1.    7.]
[  34.  309.  387.  218.   35.  279.   29.  698.    2.    9.]
[  37.   33.  154.  194.   97. 1167.   58.   85.  164.   11.]
[  17.  247.  217.  311.  244.  136.    9.  620.   93.  106.]
```

5. Number of Trees: 2500

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=2500, n_jobs=1, oob_score=False, random_state=None, verbose=0, warm_start=False)

Testing Accuracy on MNIST Dataset: 0.9721

Confusion Matrix for MNIST Testing Dataset:

```
[ 970.    0.    0.    0.    0.    2.    2.    1.    4.    1.]
[   0. 1124.    2.    3.    0.    2.    2.    0.    1.    1.]
[   6.    0. 1001.    4.    3.    0.    4.    8.    6.    0.]
[   0.    0.    8.  978.    0.    5.    0.    9.    7.    3.]
[   1.    0.    2.    0.  957.    0.    4.    0.    2.   16.]
[   3.    0.    0.    9.    3.  863.    6.    1.    5.    2.]
[   7.    3.    0.    0.    2.    4.  939.    0.    3.    0.]
[   1.    2.   17.    2.    1.    0.    0.  991.    2.   12.]
[   3.    0.    6.    7.    2.    5.    3.    3.  935.   10.]
[   5.    5.    2.   10.   10.    3.    1.    5.    5.  963.]
```

Testing Accuracy for USPS Dataset: 0.41097054852742637

Confusion Matrix for USPS Testing Dataset:

```
[ 651.   10.  268.   52.  454.  170.   51.   94.    0.  250.]
[  37.  559.  113.  101.   54.  104.   18.  999.   14.    1.]
[  84.   27. 1277.   64.   49.  221.   11.  259.    5.    2.]
[  36.    5.   85. 1319.   53.  309.    4.  171.    3.   15.]
[  11.  201.   49.   22. 1093.  187.   13.  386.   22.   16.]
[ 131.   27.  128.   57.   21. 1509.   20.   98.    6.    3.]
[ 295.   46.  200.   18.   82.  369.  840.  140.    1.    9.]
[  34.  323.  370.  225.   39.  271.   32.  695.    3.    8.]
[  38.   35.  140.  193.   98. 1160.   59.   92.  169.   16.]
[  16.  259.  212.  313.  235.  129.    7.  634.   88.  107.]
```

5 Ensemble

1. Simple Majority Voting:

The max voting method is generally used for classification problems. In this technique, 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 [4].

For example, when you asked 5 of your colleagues to rate your movie (out of 5); we’ll assume three of them rated it as 4 while two of them gave it a 5. Since the majority gave a rating of 4, the final rating will be taken as 4. You can consider this as taking the **mode** of all the predictions [4].

1.1 Ensemble of Logistic Regression, NN, SVM and Random Forest

Accuracy by performing Ensembling of Logistic Regression, NN, SVM & Random Forest: 98.07

Confusion Matrix for Ensembled (MNIST) Testing Dataset:

```
[ 973.    0.    1.    0.    0.    1.    1.    1.    3.    0.]
[   0. 1126.    2.    2.    0.    2.    1.    0.    2.    0.]
[   6.    0. 1013.    1.    1.    0.    0.    7.    4.    0.]
[   0.    0.    4.  992.    0.    5.    0.    6.    3.    0.]
[   2.    0.    0.    0.  961.    0.    5.    0.    2.   12.]
[   2.    0.    0.    5.    2.  877.    2.    1.    2.    1.]
[   4.    3.    0.    0.    2.    3.  945.    0.    1.    0.]
[   0.    9.   11.    2.    0.    0.    0.  998.    2.    6.]
[   3.    0.    2.    5.    5.    2.    0.    2.  952.    3.]
[   4.    7.    2.    6.    9.    3.    0.    5.    3.  970.]
```

Accuracy by performing Ensembling (USPS) of Logistic Regression, NN, SVM & Random Forest: 39.391969598479925

Confusion Matrix for Ensembled (USPS) Testing Dataset:

```
[ 713.   14.  747.   21.  146.  103.   11.   21.    9.  215.]
[  71.  581.  411.  122.  119.  109.   16.  528.   39.    4.]
[  79.   45. 1705.   31.   14.   78.    5.   35.    7.    0.]
[  43.   21.  519. 1189.    3.  188.    1.   16.   19.    1.]
[  18.  167.  384.   13.  938.  178.    3.  192.   92.   15.]
[ 136.   41.  531.   57.    6. 1188.   10.   15.   15.    1.]
[ 357.   30.  814.    5.   32.  123.  617.    5.   13.    4.]
[  59.  316.  766.  173.   11.  138.    3.  524.    9.    1.]
[ 160.   51.  805.  106.   29.  453.   22.   41.  330.    3.]
[  25.  280.  792.  131.  128.   59.    1.  337.  154.   93.]
```

1.2 Performing Ensembling (MNIST) of Logistic Regression, NN, SVM & Random Forest

Testing Accuracy for MNIST: 97.4

Confusion Matrix for MNIST Dataset:

```
[ 972.    0.    1.    0.    0.    1.    2.    1.    3.    0.]
[   0. 1126.    2.    2.    0.    2.    1.    0.    2.    0.]
[   6.    2. 1010.    2.    1.    0.    0.    7.    4.    0.]
[   1.    0.   10.  984.    0.    5.    0.    6.    4.    0.]
[   1.    0.    2.    0.  963.    0.    4.    0.    2.   10.]
[   2.    1.    0.    9.    2.  871.    3.    1.    2.    1.]
[   5.    3.    0.    0.    4.    4.  942.    0.    0.    0.]
[   1.    9.   18.    1.    1.    0.    0.  991.    1.    6.]
[   4.    2.    4.    8.    8.    8.    5.    3.  929.    3.]
[   9.    6.    1.    9.   13.    6.    0.    9.    4.  952.]
```

Testing Accuracy for USPS: 40.42702135106755

Confusion Matrix for USPS:

```
[ 666.    8.  569.   34.  218.  127.   36.   27.   40.  275.]
[ 131.  533.  310.  154.  204.  106.   21.  481.   54.    6.]
[ 119.   31. 1589.   53.   27.   87.   19.   60.   14.    0.]
[  71.   13.  272. 1340.   10.  222.    2.   37.   27.    6.]
[  31.  125.  169.   19. 1046.  190.   13.  190.  167.   50.]
[ 123.   26.  359.  119.   15. 1254.   32.   42.   26.    4.]
[ 347.   13.  650.   18.   55.  191.  663.   18.   25.   20.]
[ 110.  269.  636.  244.   23.  137.    8.  492.   67.   14.]
[ 169.   40.  424.  164.   59.  649.   50.   53.  371.   21.]
[  28.  219.  441.  310.  138.   82.    3.  420.  228.  131.]
```

1.3 Possible drawbacks of Simple Majority Voting:

1. Suppose for eg. A combination of three ensemble classifiers predicted values [1, 2, 3]. In such case, we have no option left but to choose a random value. If that random value is the target value, our pick turned out to be the correct one but if it's not the correct pick, then we incur a loss. This is a major drawback of Simple Majority Voting.

2. Another one is, if we have ensemble of four Classifiers, predicted values are [1, 1, 2, 2]. In such case, there is a tie, we cannot determine which is the element we have to choose as every element's count is same. In such cases, again we have to pick randomly. Thus, these randomly picked predicted values sometimes make our overall ensemble prediction wrong, and our accuracy suffers. We can observe this scenario in the above readings. Our final testing accuracy is 98.07% for ensemble of classifiers (NN, SVM, RF) but SVM itself has an accuracy of 98.4%. Our main intention for forming an ensemble of classifiers was to increase the overall correct predictions and in turn the accuracy, but due to Simple Majority Voting, there was a phase where the ensemble technique picked up a random predicted value, which turned out to be wrong, thus the accuracy decreased to some extent.

NN, which is a single classifier, can be very powerful unlike most classifiers (single or ensemble) which are kernel machines and data-driven. NN can generalize from unseen data and act as universal functional approximators [3].

5 Insights on "No Free Lunch Theorem" and Accuracy of Different Models

The "No Free Lunch" theorem states that there is no one model that works best for every problem. Let us define our problem first; Our problem is to find the accuracy on both MNIST Dataset and USPS Dataset. Thus, we used four models to train our Dataset. We carried out our training on MNIST Dataset and we tested both MNIST as well as USPS Dataset. What we observed is,

1. The testing accuracy for MNIST Dataset was far superior than what we observed for USPS Dataset.
2. We achieved a maximum testing accuracy for MNIST Dataset through SVM i.e. 98.40% but the accuracy for USPS Dataset was far more inferior to MNIST accuracy, we got a maximum testing accuracy of 45.63%.
3. For every other model, we got similar results where the testing accuracy for MNIST Dataset was significantly higher than that of USPS Dataset.

From the above observations, we can infer that, we have two problems; First is to find the testing accuracy for MNIST Dataset and second one is to evaluate accuracy for USPS Dataset. Both are different problems and that is the reason why using just one model which supposedly worked for evaluating the accuracy of MNIST Dataset, didn't worked for USPS Dataset.

6 Execution

Do execute main.ipynb file. It has all the code segments required to implement all four classifiers mentioned.

7 References

1. Teaching Assistants (CSE 574 – Introduction to Machine Learning)

2. http://rasbt.github.io/mlxtend/user_guide/classifier/SoftmaxRegression/#methods

3. https://blogs.sas.com/content/sgf/2017/03/10/are-ensemble-classifiers-always-better-than-single-classifiers/

4. https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/

5. https://chemicalstatistician.wordpress.com/2014/01/24/machine-learning-lesson-of-the-day-the-no-free-lunch-theorem/