INTRODUCTION

The main aim of this project is to implement the various classification methods to predict the class (the number) of the images given. We have MNIST and the USPS datasets. We train the models on the MNIST dataset and perform the predictions on part of the MNIST dataset and the whole USPS datasets. The main reason we do predictions on USPS datasets is to check whether our model can predict other types of datasets accurately or not.

In the project we have used mainly 4 classification algorithms. They are:

1. Logistic Regression with Softmax (Softmax Regression)
2. Neural Networks
3. Random Forest Classification
4. Support Vector Machine (SVM)

These four algorithms are explained in detail in the following chapters.

Mnist dataset

Usps dataset

LOGISTIC REGRESSION WITH SOFTMAX (SOFTMAX REGRESSION)

In regression analysis, **logistic regression** (or logit regression) is estimating the parameters of a logistic model; it is a form of binomial regression. Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail, win/lose, alive/dead or healthy/sick.

The logistic regression uses the sigmoid function for classification. But the problem here is logistic regression with sigmoid can only perform binary classification i.e, only for two classes. So to perform multi-class logistic regression we need to generalize it to many classes.

This can be done in two ways. Either by One vs. All method or using Softmax function instead of the Sigmoid function. We chose to do this by the use of the Softmax function which is also called as Softmax regression.

Softmax Regression generalizes the logistic regression to several classes. The softmax regression uses the softmax function instead of logistic function. The function is represented below:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

In the softmax regression, we convert the classes to one-hot representation which makes it easier for the machine to predict. The softmax function uses regularization and loss function in order to get accurate results. These are used to minimize the error and optimize the weights.

Regularization:

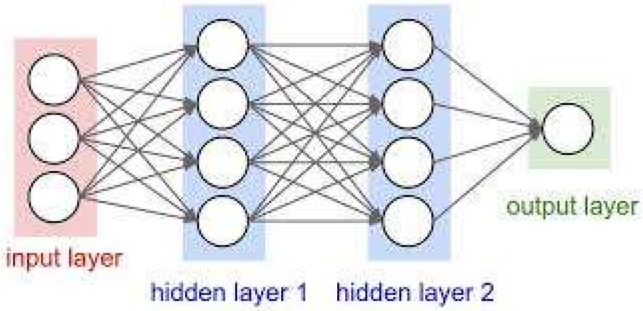
$$R(W) = \sum_{k} \sum_{d} W_{k,d}^{2}$$

Loss:

$$L = \frac{1}{m} \sum_{i=1}^{m} L_i + \frac{1}{2} \lambda \sum_{k} \sum_{d} W_{k,d}^{2}$$

NEURAL NETWORKS

Neural Networks is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. It is an algorithm that uses many machine learning paradigms. In the Neural Network we transfer data from one layer to another. We process the data while doing it. There are mainly three layers Input layer, Neural layer and the Output layer. We chose to use multiple layers in neural network model for our current case, so as to accommodate the classification problem. I have used to dense layers in the neural network classification. Our Neural network model roughly looks like this.

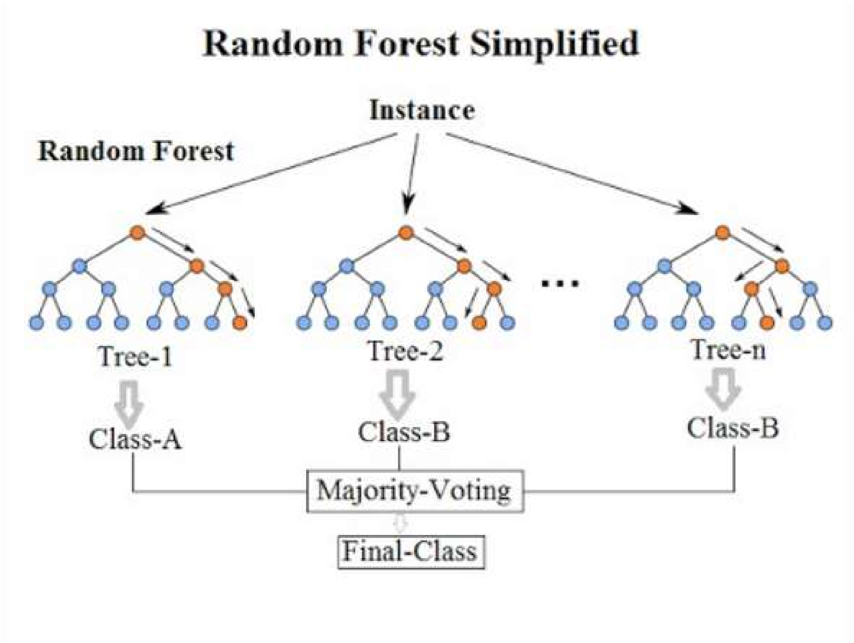


TRAINING:
In the data training process, the method makes use of error range generated from error function and tries to minimize the error by changing the weight of the transmitted values. These weights are changed by every epoch. An epoch is one complete iteration of complete training dataset.

We chose to use the dropout regularization technique, so as to avoid overfitting of the model. The activation functions ReLU and Softmax have been used. Cross Entropy is used to determine the loss function for the model and the RMSprop optimizer have been used.

RANDOM FOREST CLASSIFICATION

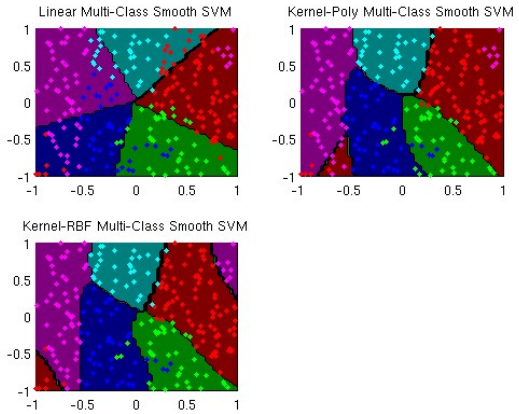
Random forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests is correct for decision trees' habit of overfitting to their training set. The simple random forest diagram is shown below:



SUPPORT VECTOR MACHINE (SVM)

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. SVM can be performed with various kernel options like linear, rbf e.tc. I chose to use ‘RBF’.



CONFUSION MATRIX

In the field of machine learning and especially in statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix).

Each row of the Matrix represents the instances in a predicted class while each column represents the instances in an actual class.

Performance measures can be derived from the confusion matrix. Some of the important measures that are derived are:

1. Error rate
2. Accuracy
3. Sensitivity
4. Specificity

ENSEMBLE WITH MAXIMUM VOTING

In statistics and machine learning, **ensemble methods** use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a machine learning ensemble consists of only a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives.

In our model we have used the majority voting process. In the majority voting process, predictions are taken from our four classifiers, if two or more classes predict the same outcome that result is taken as the prediction. In the case that all four classifiers predict different outcomes, we take the prediction of the best classifier algorithm, in my case I have taken the Neural Networks algorithm. The hard voting ensemble algorithm is represented below:

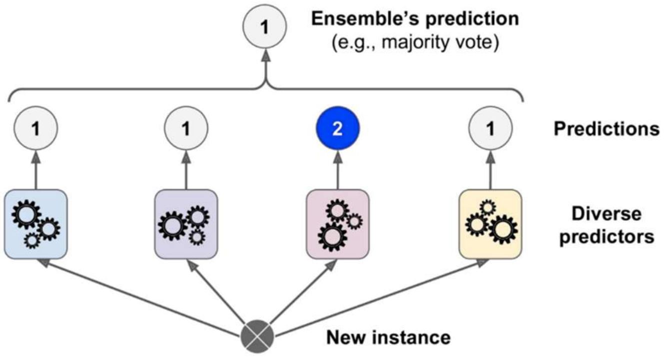


Figure 7-2. Hard voting classifier predictions

OUTPUT

LOGISTIC REGRESSION (SOFTMAX)

```
lam = 0.001
epochs = 1000
learningRate = 0.01

loss:  0.629679249902
Training Accuracy:  0.85808
Testing Accuracy:  0.8718
Confusion Matrix:
 [[ 946    0    3    3    0    4   15    1    8    0]
  [   0 1092    5    3    1    4    4    0   26    0]
  [  16   19  850   26   19    0   26   23   47    6]
  [   5    3   22  879    1   32    8   19   27   14]
  [   3    8    5    0  861    1   17    2   10   75]
  [  26   14    5   79   23  650   28    9   41   17]
  [  20    5   13    2   13   19  880    0    6    0]
  [   4   36   28    1   13    0    4  887   10   45]
  [   9   14   14   38   11   23   18   14  813   20]
  [  13   13   11   12   51   11    1   26   11  860]]
```

The rows & columns represent the predictions and the label values (correct values)

Here we can see that the cells of diagonal matrix (which represents the correct predictions) has more values when compared to other cells. This proves that we have a good accuracy.

We can see that many 7s are getting predicted as 1s, many 2s are getting predicted as 8s. But the highest error is taking place between 4s and 9s. This is because these numbers have higher similarities. Although, the model is good, it is still not good enough.

```
USPS Accuracy:  0.33206660333
USPS Confusion Matrix:
 [[ 946    0    3    3    0    4   15    1    8    0]
  [   0 1092    5    3    1    4    4    0   26    0]
  [  16   19  850   26   19    0   26   23   47    6]
  [   5    3   22  879    1   32    8   19   27   14]
  [   3    8    5    0  861    1   17    2   10   75]
  [  26   14    5   79   23  650   28    9   41   17]
  [  20    5   13    2   13   19  880    0    6    0]
  [   4   36   28    1   13    0    4  887   10   45]
  [   9   14   14   38   11   23   18   14  813   20]
  [  13   13   11   12   51   11    1   26   11  860]]
```

In this as we can see we have many cells outside diagonal which have high values. This means m any of our predictions are wrong and our model does not predict correctly to the USPS dataset.

- This is same with all other classifiers.

NEURAL NETWORK:

```
input_size = 784
drop_out = 0.2
first_dense_layer_nodes  = 650
second_dense_layer_nodes = 10
validation_data_split = 0.2
num_epochs = 1000
model_batch_size = 50000
tb_batch_size = 100
early_patience = 100
```

```
Errors: 220  Correct :9780
Testing Accuracy: 97.8
Test Confusion Matrix:
 [[ 973    0    0    3    0    0    1    1    2    0]
  [   0 1128    3    1    0    1    1    0    1    0]
  [   5    1 1005    2    3    0    2    7    6    1]
  [   0    0    1  998    0    2    0    3    4    2]
  [   1    0    2    1  961    0    3    3    2    9]
  [   2    0    0   17    1  860    2    3    5    2]
  [   3    4    0    1    4    4  937    0    5    0]
  [   1    3   11    5    3    0    0  998    1    6]
  [   1    0    4    8    4    5    1    5  942    4]
  [   2    5    0    8    7    3    0    6    0  978]]
```

In this model, we can see that most of the non-diagonal cells contain only single digit numbers. This shows us that the predictions of this model has been very accurate, and also the non- diagonal cells has very low values, when compared to the diagonal matrix values. This shows us that the model is very accurate. This model is very good when compared to the other four models.

```
====================USPS==========================
Errors: 13242  Correct :6757
USPS Accuracy: 33.78668933446672
USPS Confusion Matrix:
 [[ 278    0  295  139   76  100  262  659   46  145]
  [  15  288  658  113  406   92   26  243  127   32]
  [  21   15 1562   66   17   85  141   57   31    4]
  [   2   16  443 1047    3  338   63   57   25    6]
  [   6   15  274   58  772   51   71  608  113   32]
  [   7    2  901  178    5  633  132  100   38    4]
  [  36   10  611   31   44  200  735  291   29   13]
  [   4   54  165  504   24   39   29 1081   96    4]
  [  61    3  186  577   88  219  176  425  257    8]
  [   3   27  162  325  103   17   30 1008  221  104]]
```

RANDOM FOREST

```
Test Accuracy: 0.966
Test Confusion Matrix
 [[ 971    0    1    0    0    1    2    1    3    1]
  [   0 1121    2    4    0    2    2    0    3    1]
  [   6    0  998    6    1    0    4   10    7    0]
  [   0    0   15  965    0    9    0    9    9    3]
  [   2    0    2    0  949    0    5    0    2   22]
  [   3    0    0   15    4  852    8    2    6    2]
  [   7    3    1    0    4    4  934    0    5    0]
  [   1    2   23    1    2    0    0  986    2   11]
  [   4    0    4    7    4    7    5    3  927   13]
  [   6    4    2   11   12    4    1    6    6  957]]
```

Even this model gives highly accurate results. But the problem with this model is, when compared to Neural Network, this has a ‘little’ higher concentration outside the diagonal of the matrix. As we can see many 7s are predicted as 2s and many 4s are predicted as 9s. This is a little concern to us.

```
USPS Accuracy: 0.386469323466
USPS Confusion Matrix
 [[ 599   10  225   51  465  148   71  168    1  262]
  [  22  601   90  119   51   67   18 1016   15    1]
  [  78   45 1172   83   58  181   18  356    6    2]
  [  42   12   86 1240   53  297    1  247    3   19]
  [   9  236   53   22 1074  136    9  418   20   23]
  [ 135   33  119   93   39 1379   24  166    3    9]
  [ 325   70  220   34  109  318  728  182    4   10]
  [  35  368  327  213   45  242   31  731    2    6]
  [  62   76  157  201  128 1022   73  135  119   27]
  [  16  303  224  263  256  122    7  666   57   86]]
```

SVM (SUPPORT VECTOR MACHINE)

```
Testing Accuracy:  0.9748
Confusion matrix:
 [[ 972    0    0    0    0    2    3    1    2    0]
  [   0 1126    3    1    0    1    1    1    2    0]
  [   3    2 1008    2    1    0    1    9    5    1]
  [   0    0    4  985    0    5    0    7    8    1]
  [   1    0    5    0  959    0    3    0    2   12]
  [   5    0    0   13    1  860    4    1    6    2]
  [   6    2    1    0    2    5  940    0    2    0]
  [   0    9   14    2    2    0    0  990    1   10]
  [   3    0    3    9    5    2    3    3  945    1]
  [   3    6    1    7   14    2    1    8    4  963]]
```

This model too has much concentration in the diagonal cells. This is the best method next to the Neural networks as this method still has values with good concentration in few areas outside of the diagonal.

```
USPS Accuracy:  0.399119955998
Confusion matrix:
 [[ 592    1  397   39  176  330   54   70    2  339]
  [  87  405  360  144  141  152   26  651   18   16]
  [  70    6 1617   45   19  163   28   41    6    4]
  [  28    6  246 1190    1  477    0   43    3    6]
  [  14   41  164   26  985  282   16  344   51   77]
  [  66   12  283   68   11 1494   25   28   10    3]
  [ 147    5  685   24   50  289  771   13    1   15]
  [  43  144  468  439   25  317    6  528   17   13]
  [  71    9  258  261   51 1060   57   39  187    7]
  [  12   83  260  372  131  154    4  621  150  213]]
```

ENSEMBLE:

```
Ensemble Test Accuracy:  0.971
Ensemble Test Confusion Matrix:
 [[ 973    0    1    0    0    0    2    1    3    0]
  [   0 1126    3    1    0    1    1    0    3    0]
  [   8    2 1003    2    1    0    1    8    6    1]
  [   0    0   13  978    0    3    0    5    8    3]
  [   1    0    3    0  953    0    4    1    2   18]
  [   5    0    0   21    2  849    4    1    8    2]
  [   7    3    1    0    4    6  933    0    4    0]
  [   1    6   22    1    3    0    0  982    1   12]
  [   3    0    4    8    4    1    4    6  942    2]
  [   6    7    2    7    7    1    1    4    3  971]]
```

This confusion matrix is a result of the maximum voting system of all the four matrices. This mig ht look like it has a few error concentrations in some areas but, it is still better than above four, as it has more cells than any other classifiers with 0 values.

```
Ensemble USPS Accuracy:  0.4030701535076754
Ensemble USPS Confusion Matrix:
 [[ 647    2  371   58  272  120   86  155   10  279]
  [  90  427  350  161  223   75   18  548   97   11]
  [ 110   22 1560   50   31   97   45   66   13    5]
  [  54    8  240 1332   12  259   12   56   13   14]
  [  13  103  127   32 1089  138   21  331   99   47]
  [ 115   16  351  126   16 1246   39   66   19    6]
  [ 312   13  523   31   74  222  722   75   12   16]
  [  54  229  370  400   33  171   25  648   60   10]
  [ 123   28  188  319   94  769   85  127  253   14]
  [  17  196  210  355  160   88   10  629  198  137]]
```

Even the ensemble’s confusion matrix has high values outside of the diagonal representing that our model cannot predict correct values for the USPS dataset.

QUESTIONS AND ANSWERS (CONCLUSION)

1.Do your results support the “No Free Lunch” theorem?

Yes, our problem supports the ‘no free lunch’ theorem.
We know that no free lunch theorem states that a model trained for one dataset or problem doesn’t work with same accuracy for other problems or datasets.
As we can see from our output, the accuracy of the predictions is considerably low of USPS when compared to the test set of MNIST. This is because, they are of different datasets.
Our results support “No Free Lunch” theorem.

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

Neural network is the best classifier among our four classifiers. This can be derived after analyzi ng all the confusion matrices of different classifiers. Analysis if each classification is done in the output section of the report.
Neural Network is the best method because, even if the other classifiers have good diagonal val ues, few errors are concentrated in one section, whereas Neural Networks doesn’t have this pr oblem. This makes neural networks more precise and better.

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?

Yes, the accuracy after combining the results is better than any of the individual methods. This is because few of the individual errors are covered in the ensemble method. Let’s say one method gave output as 9 and other three gave it as 4. The result is 4 but the former method is predicting wrong. Few such errors are removed from the predictions by combining the results. The overall result is not considerably better than individual ones, but still better than them.

REFERENCES

1. https://classeval.wordpress.com/introduction/basic-evaluation-measures/
2. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html
3. https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72
4. https://houxianxu.github.io/2015/04/23/logistic-softmax-regression/
5. https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html
6. https://www.kaggle.com/atorin/mnist-digit-recognition-with-random-forests
7. https://en.wikipedia.org/wiki/Softmax_function
8. https://en.wikipedia.org/wiki/Ensemble_learning