UBitName: abhave personNumber:50289049

CSE 574 Introduction to Machine Learning

Project 3 Report

The task was to implement machine learning methods for the task of classification. We had to implement an ensemble of four classifiers for a given task. Then the results of the individual classifiers were to be combined to make a final decision.

The classification task was that of recognizing a 28X28 grayscale handwritten digit image and identify it as a digit among 0, 1, 2, ..., 9.

For the purpose of this task we had 2 different datasets:

1. MNIST Dataset
For both training and testing of our classifiers, we used the MNIST dataset. The MNIST database is a large database of handwritten digits that is commonly used machine learning. The database contains 60,000 training images and 10,000 testing images.

2. USPS Dataset
We use USPS handwritten digit as another testing data for this project to test whether your models could be generalized to a new population of data. Each digit has 2000 samples available for testing, so a total of 20000 samples are available to test on.

The pipeline for processing and running the 4 classifiers is as follows:

1. Data preprocessing
2. Partitioning the data
3. Train the model parameter
4. Tune the hyperparameters
5. Test on MNIST dataset
6. Test on USPS dataset
7. Perform Majority Voting for 4 classifiers

1 - Data Processing and Partition

(a) We have 2 datasets namely USPS and MNIST. We read the MNIST dataset and separate the target and the features and divide these two datasets into training set, testing set, and validation set.
The division is 50,000 training, 10,000 validation and 10,000 testing samples for MNIST. We keep the whole USPS dataset as the testing set of 20,000 samples.

2 –Logistic Regression Solution

What is Logistic Regression?

Logistic Regression is basically a binary classifier which estimates or predicts 0 or 1 values. Since we want to do digit recognition for 0-9 digits, we use SoftMax as it is a multiclass classifier. SoftMax regression is a method in machine learning which allows for the classification of an input into discrete classes. Unlike the commonly used logistic regression, which can only perform binary classifications, softmax allows for classification into any number of possible classes. The softmax function is given by [1]:

$$p(\mathcal{C}_k|\mathbf{x}) = y_k(\mathbf{x}) = \frac{exp(a_k)}{\sum_j exp(a_j)},$$

(a) Features and Targets

Here are the steps for calculating the weights:

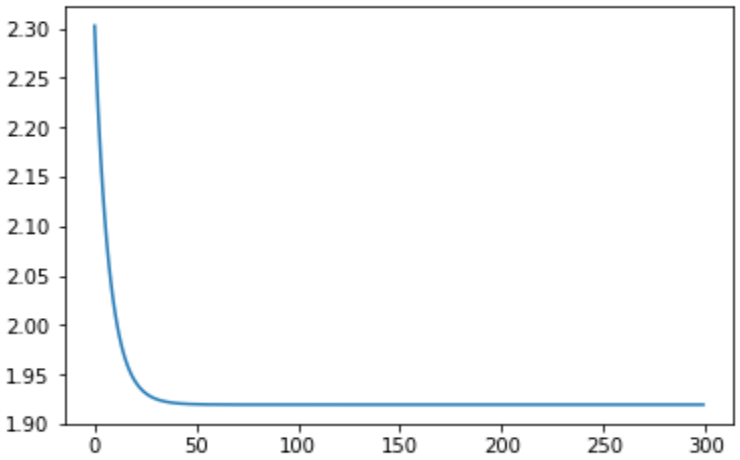
(1) Initialize weights
We randomly initialize the weights as a vector of zeroes. We could use randomly initialize W by any value.

(2) Update weights iteratively

(3) Calculating error
When the model is trained using the training data and the model is functioning, we have no way of knowing whether the predicted value is correct. So, to test that out we use the error function. The error function or cost function is called as cross entropy.
The cost function is as follows:

$$E(\mathbf{x}) = -\sum_{k=1}^{K} t_k \ln y_k,$$

We keep the number of epochs as 300 and learning rate as 0.05 and lambda as 1.



Logistic Regression

Loss vs Number of Epochs

```
MNIST Training accuracy is: 0.7832
MNIST Testing accuracy is: 0.7992

The confusion matrix for MNIST is as follows:

[[ 945    0    3    5    0    0   19    1    7    0]
 [   0 1082   10    7    0    0    4    1   31    0]
 [  43   51  789   36   15    0   37   21   39    1]
 [   9    7   24  902    0    0    9   18   29   12]
 [   8   16    5    1  779    0   36    2   20  115]
 [ 100   43   11  277   21  242   49   32   84   33]
 [  48   15   16    2    5    3  864    0    5    0]
 [  11   55   27    0    7    0    4  882   11   31]
 [  25   39   14  109    7    0   23   19  717   21]
 [  33   22   12   18   55    0    4   57   18  790]]

USPS accuracy is: 0.289

The confusion matrix for USPS is as follows:


[[ 997   10  360   59  315    1   48   25   45  140]
 [ 368  311  230  220  136    2   53  387  285    8]
 [ 507   62  989  110   48    1  109   94   70    9]
 [ 409    8  130 1101   42   13   58   82  113   44]
 [ 358  113   55   91  917    9   37  149  212   59]
 [ 612   50  246  411   44  178  152  101  154   52]
 [ 903   17  308   82   84    2  530   20   39   15]
 [ 309  291  393  354   46    9   59  267  244   28]
 [ 556   81  260  247  128   37  172   44  426   49]
 [ 211  287  196  420  137    3   28  364  278   76]]
```

3 –Neural Network Solution

A neural network has input and output neurons, which are connected by weighted synapses. The weights affect how much of the forward propagation goes through the neural network. The weights can then be changed during the back propagation — this is the part where the neural network is now learning. This process of forward propagation and backward propagation is conducted iteratively on every piece of data in a training data set. [4] I have implemented neural network using the Keras library which is built on top of TensorFlow library. I have used the SoftMax activation function and have keep the number of epochs as 100 and the neural network consists of 2 layers with batch size of 128.

```
For MNIST
Accuracy: 0.927 and Loss: 0.253

The confusion matrix for MNIST is as follows:

[[ 958    0    2    1    0    4   10    2    3    0]
 [   0 1107    2    4    1    2    4    2   13    0]
 [  13    3  931    7   13    0   12   14   35    4]
 [   1    1   19  933    0   23    3   13   14    3]
 [   1    4    6    0  919    0    9    2    5   36]
 [   9    4    3   48    9  767   13    6   26    7]
 [  11    3    5    0    9   12  912    2    4    0]
 [   4   11   26    4    5    0    0  951    2   25]
 [   6    8    7   19    9   18   12   13  878    4]
 [  11    7    1   13   33   10    1   13    3  917]]


For USPS
Accuracy: 0.382 and Loss: 2.582

The confusion matrix for USPS is as follows:

[[ 582    1  259   67  218  193   80   69   91  440]
 [ 112  343  219  255  203  137   39  553  106   33]
 [ 159   16 1291  140   42  145   69   70   47   20]
 [  64    3  121 1394    8  263   13   52   52   30]
 [  32   46   43   34 1049  190   38  197  177  194]
 [ 129   10  179  190   34 1199  111   70   57   21]
 [ 323    5  397   83   91  252  752   18   41   38]
 [ 135  163  363  446   41  212   19  433  126   62]
 [ 150   28  171  203  105  739  112   58  367   67]
 [  29  135  155  435  154  101   13  458  272  248]]
```

4 –Support Vector Machine Solution

Support Vector Machine is a high margin classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. [5] I have executed the SVM model for all the 3 settings which include kernel as linear and radial basis function and values of gamma as 0.05 and 1 with radial basis function.

For MNIST Accuracy: 0.9364

The confusion matrix for MNIST is as follows:

```
[[ 958    0    5    1    1    3    8    1    1    2]
 [   0 1117    4    4    0    1    2    1    6    0]
 [   6   11  960   13    3    1   12   10   14    2]
 [   4    2   19  944    3   13    1    7   14    3]
 [   2    1    9    0  944    0    5    1    2   18]
 [  15    7    4   39    5  787   11    1   19    4]
 [  10    3   11    1    5   13  912    1    2    0]
 [   0   10   20   10    5    2    0  960    4   17]
 [  11    6    7   24   10   22    8    9  869    8]
 [   7    7    2   13   33    3    0   22    9  913]]
```

For USPS Accuracy: 0.285

The confusion matrix for USPS is as follows:

```
[[ 358    1  493  172  239  316   69  166   11  175]
 [  59  282  572  265  240  162   15  339   44   22]
 [ 132   79 1256  131   35  224   61   48   21   12]
 [  65   52  364  884   14  501    8   43   50   19]
 [  28   27  214   90  820  213    8  456   80   64]
 [  46   26  682  249   45  824   37   38   36   17]
 [ 152   17  916   64   81  250  450   38    2   30]
 [  20   71  190  715   61  296   12  518   84   33]
 [ 121   17  278  488  123  648   83   68  154   20]
 [  13   35  200  579  166  105    8  587  146  161]]
```

SVM Settings	kernel=linear gamma=default	kernel=rbf gamma=0.05	kernel=rbf gamma=1
MNIST Accuracy	92.5 %	93.6 %	96.2 %
USPS Accuracy	27.5 %	28.5 %	30.5 %

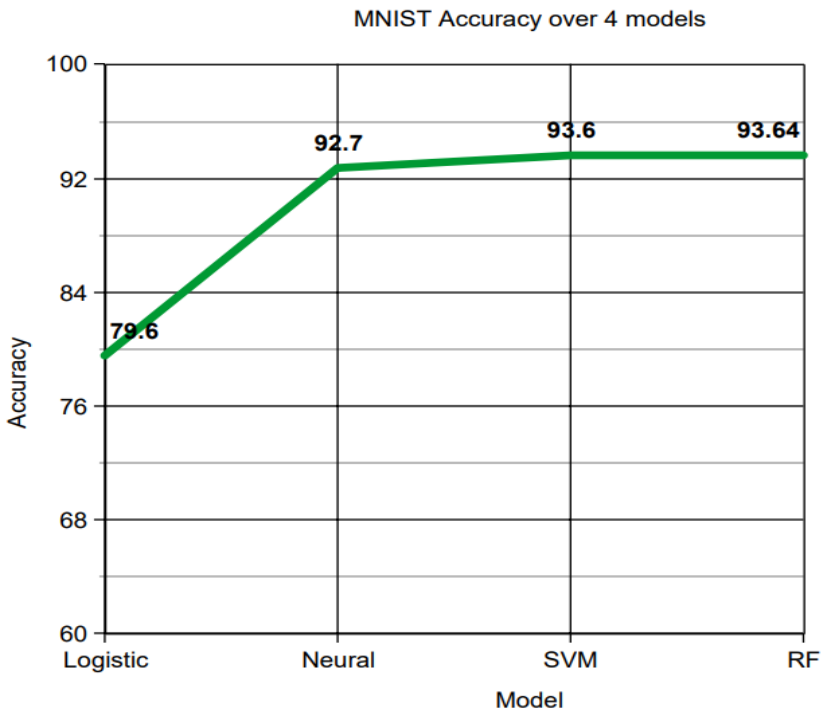
5 –Random Forest Solution

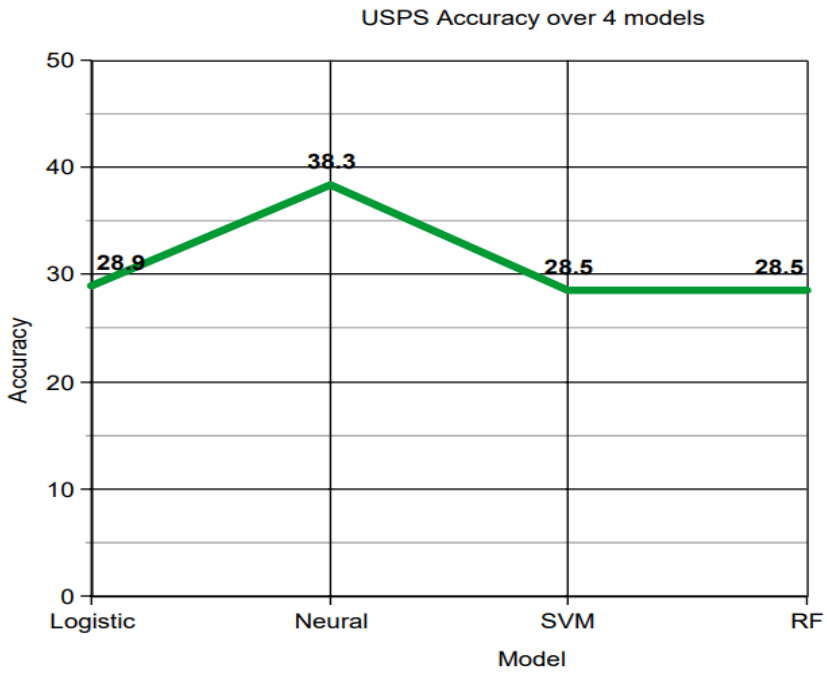
Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. [2] I have implemented random forest and optimized it to give best accuracy and have taken the number of estimators or the number of trees as 100 which gives best accuracy.

```
For MNIST Accuracy: 0.9364

The confusion matrix for MNIST is as follows:

[[ 958    0    5    1    1    3    8    1    1    2]
 [   0 1117    4    4    0    1    2    1    6    0]
 [   6   11  960   13    3    1   12   10   14    2]
 [   4    2   19  944    3   13    1    7   14    3]
 [   2    1    9    0  944    0    5    1    2   18]
 [  15    7    4   39    5  787   11    1   19    4]
 [  10    3   11    1    5   13  912    1    2    0]
 [   0   10   20   10    5    2    0  960    4   17]
 [  11    6    7   24   10   22    8    9  869    8]
 [   7    7    2   13   33    3    0   22    9  913]]
```

```
For USPS Accuracy: 0.285

The confusion matrix for USPS is as follows:

 [[ 358    1  493  172  239  316   69  166   11  175]
  [  59  282  572  265  240  162   15  339   44   22]
  [ 132   79 1256  131   35  224   61   48   21   12]
  [  65   52  364  884   14  501    8   43   50   19]
  [  28   27  214   90  820  213    8  456   80   64]
  [  46   26  682  249   45  824   37   38   36   17]
  [ 152   17  916   64   81  250  450   38    2   30]
  [  20   71  190  715   61  296   12  518   84   33]
  [ 121   17  278  488  123  648   83   68  154   20]
  [  13   35  200  579  166  105    8  587  146  161]]
```

	RF Number of Trees=10	RF Number of Trees=100
Accuracy	91.6%	93.64%


MNIST Accuracy over 4 models
100
92
84
76
68
60
Accuracy
92.7
93.6
93.64
79.6
Logistic
Neural
SVM
RF
Model


USPS Accuracy over 4 models
50
40
30
20
10
0
Accuracy
38.3
28.9
28.5
28.5
Logistic
Neural
SVM
RF
Model

6 –Majority Voting

I have implemented hard voting wherein I have checked the frequency of the predictions of Logistic regression, Neural Network, SVM and Random Forest for both MNIST and USPS datasets.

I have selected the predictions that occurs most out of these 4 classifiers and in case of a tie I consider any one of these at random. After running the ensemble classifier, we get a combined accuracy of-

After voting 94.45% MNIST

After voting 37.576% USPS

7 – Questions to be answered / Observations

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?

We have trained the 4 models i.e Logistic Regression, Neural Network, Support Vector Machine and Random Forest on 50,000 samples of the MNIST dataset.
Now we test the model on the MNIST dataset which gives us good results.
Now when we test the same model on the USPS dataset, we do not get good results. This is because the model was trained on the MNIST dataset and even though both datasets are used for the same purpose of digit recognition we get varied results. This proves that even by using the same model/algorithm on 2 different datasets, we get different results. This supports the "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?

Out of all the 4classifers we find that the Support Vector Machine works best or gives the best results on both the MNIST and USPS dataset. So, the best overall performance is given by Support Vector Machine.

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?

I have implemented majority voting for the predictions of Logistic regression, Neural Network, SVM and Random Forest for both MNIST and USPS datasets. I have selected the predictions that occurs most out of these 4 classifiers and in case of a tie I consider any one of these at random.

The overall combined performance is better than that of logistic regression. The combined performance is as follows:

After voting 94.45% MNIST

After voting 37.576% USPS

References:

1- https://medium.com/@awjuliani/simple-softmax-in-python-tutorial-d6b4c4ed5c16

2- https://en.wikipedia.org/wiki/Random_forest

3- https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d

4- https://medium.com/machinevision/overview-of-neural-networks-b86ce02ea3d1

5- https://en.wikipedia.org/wiki/Support_vector_machine