CSE 574: Introduction to Machine Learning

Fall 2018

Submitted by: Sai Kalyan Katta (UB Person Number: 50292522)

Logistic Regression

The **One_to_N** function takes the target values ranging from 0 to 9 and converts them into an n x10 matrix and returns the matrix which is known as the one hot encoding.

The **decode** function converts the one hot encoded matrix into the original array.

The function '**Logistic_regression**' performs the logistic regression on the data passed to the function in which the accuracies are printed at last. First, we calculate the value of xW^T by taking random value for W (dimension of W is n x10) and then calculate the predicted target values Y by applying the softmax function on xW^T which is defined as

$$\text{Softmax}(a) = e^a / \sum e^a$$

done by the function '**Softmax**'.

Then, the gradiant is found calling the '**gradiant**' function which is used to calculate W_next. We calculate the accuracy of our findings of target values Y by comparing it with the original Target values.

The '**gradiant**' function gives the calculation of gradiant which is used in calculating the theta_next values.

$$\text{gradiant} = x^T(Y-t)$$

Where Y is the predicted target values

t is the actual target values

x is the data

The '**Get_Acc**' function gives the accuracy of the findings.

Results:

For MNIST dataset:

```
confusion matrix

  956    0   11    5    2   15   16    3    9   13
    0 1102    7    1    6    6    3   20   10    8
    3    2  888   18    5    6    6   29   10    6
    3    4   19  898    0   43    2    4   29   11
    0    1   15    1  901   15   13   11    8   43
    2    2    0   32    0  726   16    0   26   16
    8    4   17    6   10   17  897    0   12    0
    1    0   21   15    2   10    1  918   13   24
    7   20   45   22    8   43    4    4  840    6
    0    0    9   12   48   11    0   39   17  882

Accuracy = 90.08
```

For USPS dataset:

```
confusion matrix

   601  234  219  108   65  182  380  198  226   52
     4  298   25    3   86   20   13  213   30  188
   375  126 1176  121   36  214  346  318  146  164
    56  350  138 1259   62  184  106  450  208  470
   255  286   67   21 1028   45  105   74  127  155
   111   52   75  236  120 1031  218   78  573   84
   104   41   93   31   41  126  698   35  119   15
    42  299   93   58  128   72   25  300   43  365
   147  296   90  103  292   89   75  287  444  339
   305   18   23   60  142   37   34   47   84  168

Accuracy = 35.01675
```

Neural Network

The model that was created in this implementation is of a sequential composition which is basically a stack of layers. All the nodes in one layer are connected to all nodes in the next layer. Every layer has an activation function that induces non-linearity. The activation function used for the first hidden layer and the second hidden layer are 'sigmoid' and '**softmax**' respectively. The 'sigmoid' activation function maps the input between 0 to 1 so that the neural network so that the values do not increase exponentially whereas the '**softmax**' activation function outputs the probabilities. The optimizer used is the 'sgd' and the loss function used is the 'categorical_crossentropy'. Here, the batch size, epochs, validation_split are taken as 124,1000,0.2 respectively.

Results:

For MNIST dataset:

```
Accuracy = 95.93

  965    0    6    0    1    6    9    2    6    6
    0 1121    4    1    1    2    3    6    3    6
    1    2  982   12    3    0    3   16    2    1
    1    1    6  968    1   15    0    9    9   10
    0    0    9    1  943    1    5    1    5   25
    4    1    3    8    0  838   10    0    8    7
    7    5    6    2    6   12  925    0    7    0
    1    2    9    8    1    2    0  981    4    7
    1    3    5    8    2   10    3    1  926    3
    0    0    2    2   24    6    0   12    4  944
```

For USPS dataset:

```
Accuracy = 39.34196709760982

  531   53   69   22   27   81  145   93  253   37
    0  242   13    1    6   11    2   45    4   18
  148  380 1414  195   69  205  453  182  223   99
  114  225  124 1183   70  175   80  681  408  543
  171  174   26    8  976   11   45   43   72  129
  223  138  189  432  117 1315  265  136  440   49
   60   39   62    4   25   44  911   12   70    7
  144  555   42   32  340   65   15  624   98  546
   96  141   52  110  204   73   33  139  373  273
  513   53    8   13  166   20   51   45   59  299
```

Random Forest
The random forest classifier is implemented by changing the value of number of trees from 6 to 10.

Results:
For trees =6

For MNIST dataset:

```
Accuracy = 92.83

  970    0   12    6    5   17   14    1    9    6
    0 1122    6    4    3    2    3   14    3    7
    3    2  968   28    7    6    9   27   18    5
    0    3   10  918    6   46    2    4   25   15
    0    2    4    2  910    8   13   12    6   48
    3    3    4   22    1  792   11    2   24    9
    1    1    5    1    5    8  903    2   10    3
    1    0   11   10    4    1    0  952    6   12
    1    2   11   13    5    5    2    3  856   12
    1    0    1    6   36    7    1   11   17  892
```

For trees =6

```
Accuracy = 28.466423321166058

 625  90 234 111  79 250 449 131 185 102
  32 574 163 126 211  95  77 365 144 273
 306 183 837 262 138 210 316 427 259 264
  83 103 148 793  89 251  78 210 250 289
 327  77  83  94 790 100 135  86 215 266
 181 101 165 341 172 827 279 168 552 144
 120  60  86  29  55  76 484  37 112  53
 116 747 212 143 340  97 108 514  84 401
  23  19  24  36  35  50  33  30 136  95
 187  46  47  65  91  44  41  32  63 113
```

For trees = 8

For MNIST dataset:

```
Accuracy = 94.17
  968    0   17    4    3    7   13    3    9    9
    0 1123    2    2    3    1    4    8    6    5
    1    2  979   18    5    4   10   27   15    3
    2    2    7  940    2   40    1    7   15   13
    1    0    3    1  930    5   10   10   12   29
    1    2    1   19    1  808    9    2   10   11
    4    4    4    0    5    6  909    0    4    2
    0    2    7    9    0    3    0  951    9    7
    3    0    9   12    5   11    2    0  889   10
    0    0    3    5   28    7    0   20    5  920
```

For USPS dataset:

```
Accuracy = 29.856492824641233

 568  63 231 141  54 162 357  69 171  78
  88 551 148  65 243  74  86 473 181 327
 295 133 919 212 120 205 307 275 210 309
 106 141 165 927  75 278 123 225 278 283
 345 108  84  97 812  97 159  76 164 247
 192  72 151 348 192 883 294 207 607 134
 143  48  76  29  68  90 500  43 116  33
  97 858 192 137 323 163 124 595 102 430
  22  16  18  16  52  22  18  18 126  69
 144  10  15  28  61  26  32  19  45  90
```

For trees =10

For MNIST dataset:

```
Accuracy = 94.23

  967    1    8    3    4    7   13    4    7    8
    0 1117    0    1    1    0    3   10    5    4
    0    5  979   19    3    5    1   23   15    5
    1    4    4  937    1   33    3    5   16   13
    0    0   10    2  935    8    4    9   10   32
    4    1    1   23    0  807    8    2   17    7
    3    3    7    1    7   12  920    0    9    1
    2    2   14   10    2    5    0  958    3    7
    3    2    9    8    5    9    5    6  881   10
    0    0    0    6   24    6    1   11   11  922
```

For USPS dataset:

```
Accuracy = 30.72153607680384

  613   56  184  115   37  199  390   69  156   47
   54  479  120   51  217   89   62  376  115  282
  277  210 1006  186  148  184  298  355  290  306
   95  138  144  967  101  281   86  228  243  325
  330   74   75   63  850   66  125   62  166  217
  188   76  189  414  197  916  311  238  646  137
  109   32   54   17   48   82  502   49  105   45
  149  907  198  131  313  137  163  579  106  448
   18   14   16   19   19   23   23   17  114   75
  167   14   13   37   70   23   40   27   59  118
```

SVM

The SVM classification is implemented by changing the kernel, C and the gamma value.
The changes in the values are:
linear kernel default parameters
sgd with default parameters

For the case 1:

For MNIST dataset:

```
Accuracy = 93.64

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

For USPS dataset:

```
Accuracy = 28.536426821341067

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

For case 2:

```
Accuracy = 94.18

  969    1    0    1    0    4    4    1    0    0
    0 1110    5    3    5    0    5    0    7    0
    7    2  979   11    5    1    3    9   13    2
    5    4   16  924    4   25    1   16   12    3
    3    1    3    2  933    2    7    2    3   26
   10    0    7   28    6  813    8    4   10    6
   14    4    3    2    5    4  920    0    6    0
    1    7   24    9    3    2    0  962    4   16
    7    3   16   18   10   15   10    6  878   11
    8    3    7   13   17   12    0    7   12  930
```

Combined Models

For MNIST dataset:

```
  971    0    9    3    2   11   10    2    9    9
    0 1124    5    1    1    2    3   12    5    7
    1    2  985   15    5    4    6   25    7    3
    1    1    5  966    1   37    1    3   16   14
    0    0    5    0  954    4    7    5    8   36
    2    1    0   10    0  809   13    0   20    7
    3    4    6    1    4    9  918    0    9    0
    1    1    8    7    0    1    0  972    5   12
    1    2    8    6    2   10    0    2  891    3
    0    0    1    1   13    5    0    7    4  918

Accuracy = 95.08
```

For USPS dataset:

```
  666  128  171   83   50 1999 2000 2000 2000 2000
   16  371   47   18  118    0    0    0    0    0
  387  366 1405  235  109    0    0    0    0    0
   93  265   94 1264   72    1    0    0    0    0
  238  205   31   10 1052    0    0    0    0    0
  166  104  127  300  139    0    0    0    0    0
   56   23   43    5   12    0    0    0    0    0
   83  452   61   29  241    0    0    0    0    0
   27   76   15   42  140    0    0    0    0    0
  268   10    5   14   67    0    0    0    0    0

Accuracy = 23.791189559477974
```

Comparison:
From the results above it is seen that the neural networks and the SVM classification have performed well in predicting the target values. It is seen that the accuracies from the models for the USPS datasets when they are trained with MNIST dataset are very low. Thereby, supporting the "No Free Lunch Theory" which states that there is no universal training dataset that gives more accuracies for any possible testing datasets. It is also seen that the overall combined performance is better than the individual classifiers.

Advantages of the classifiers:

Neural networks:
The neural networks can handle very large amounts of data.
They can induce the non-linearity in the input data.

Logistic Regression:
It is more robust to the non-linearity.
Widely used and more efficient.
No homogeneity of the variance assumption.

SVM:
Just like the neural networks, SVM classification is very good in avoiding the non-linearity in the data.

Random Forests:
Random Forests are good at handling very large amounts of data.