CS545: Machine Learning Assignment- 1 (PERCEPTRONS)
Submitted By: Palak Goel
Dated: April 17, 2017

Question-1: A one-paragraph description of the experiment.

I have designed and implemented 10 perceptrons which are trained to detect the handwritten digits(0-9) i.e. one perceptron for each digit. If we classify this learning, it is a supervised learning as we train each perceptron with 60,000 training dataset and then test it with 10,000 testing dataset. Firstly, we read and classified the training and testing data (in labels and values) from CSV file. While reading, we scaled the data by dividing it with 255. Then for each learning rate, we generated the random weights. Since the weights should be in range (-0.5, 0.5), therefore subtracted by 0.5 and then multiplied by 0.1 to bring them in the given range. Then for each input vector, we calculated the dot product (of input with its weight).Updated the weights for the inputs which had wrong outputs. Then after finishing for each epoch, we calculated the accuracy by using method accuracy_score in the sklearn.metrics library. While calculating the accuracy, it may happen than multiple perceptrons fire 1, then, we consider the digit of which perceptron generates the maximum dot product.
Accuracy can be calculated using confusion matrix. Confusion matrix is the matrix which has count of which was the expected digit and the digit actually detected. Thus sum of diagonal/ total sum of matrix is accuracy. We calculate accuracy on training data as well as test data. We followed this process for 50 epochs and plotted them on graph (epoch vs accuracy) for 3 learning rates i.e. 0.1, 0.01, 0.001 using matplotlib.pyplot library.

Question-2 a: Plot of accuracy (fraction of correct classifications) on the training and test set at each epoch (including epoch 0), along with comments as to whether you are seeing either oscillations or over fitting.

<u>For learning rate = 0.1</u>

Explanation:

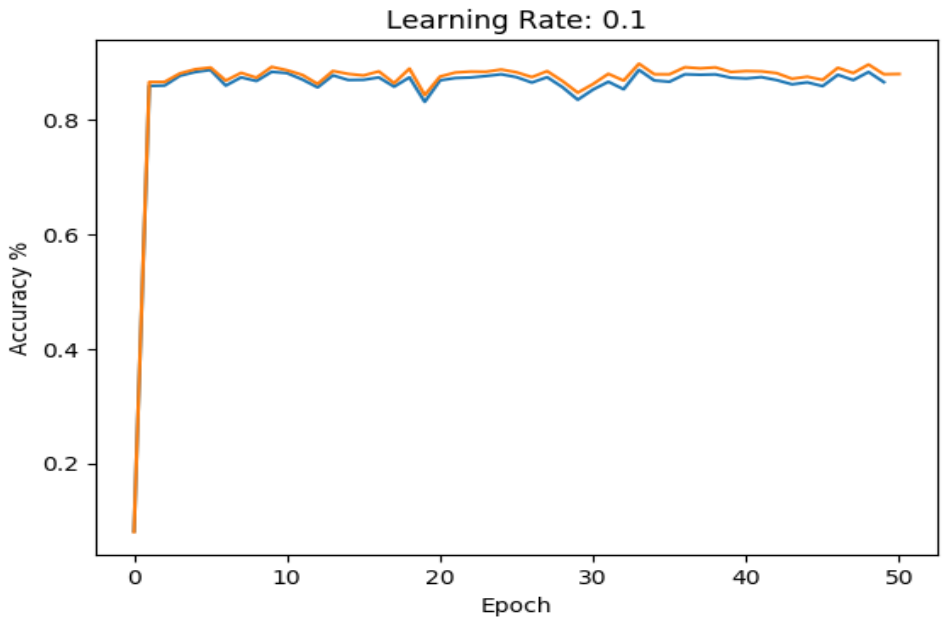
Over fitting is when perceptron is not able to generalize the training data to correctly classify the test data. Thus, the difference between training and test accuracy is large.

Since, there is not much significant difference between the training accuracy and test accuracy. Thus, we can say that these are oscillations in all the three learning rates.

Graph:

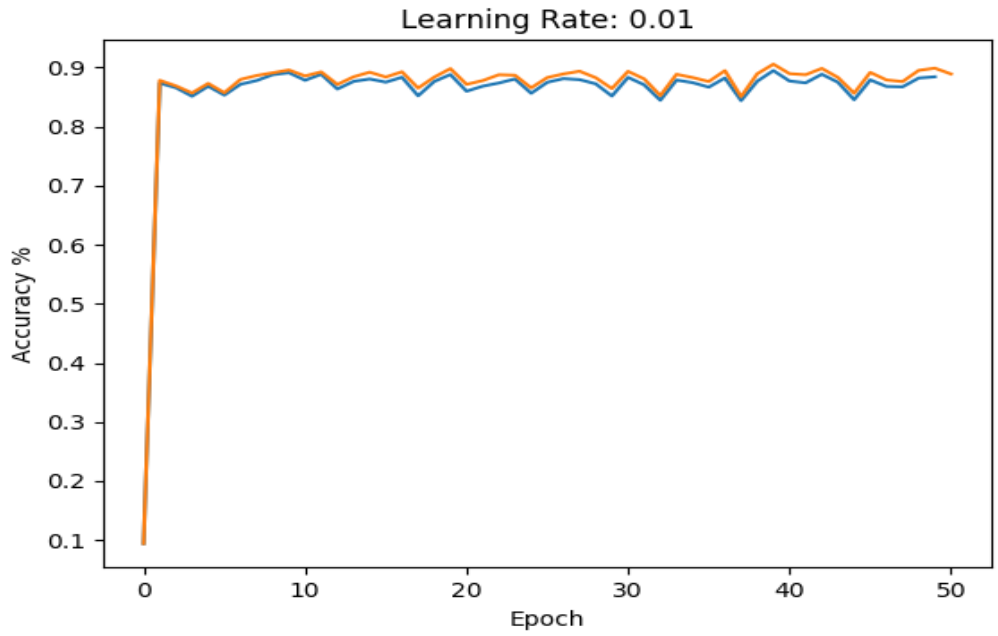
Orange line represents Training accuracy

Blue line represents Test accuracy



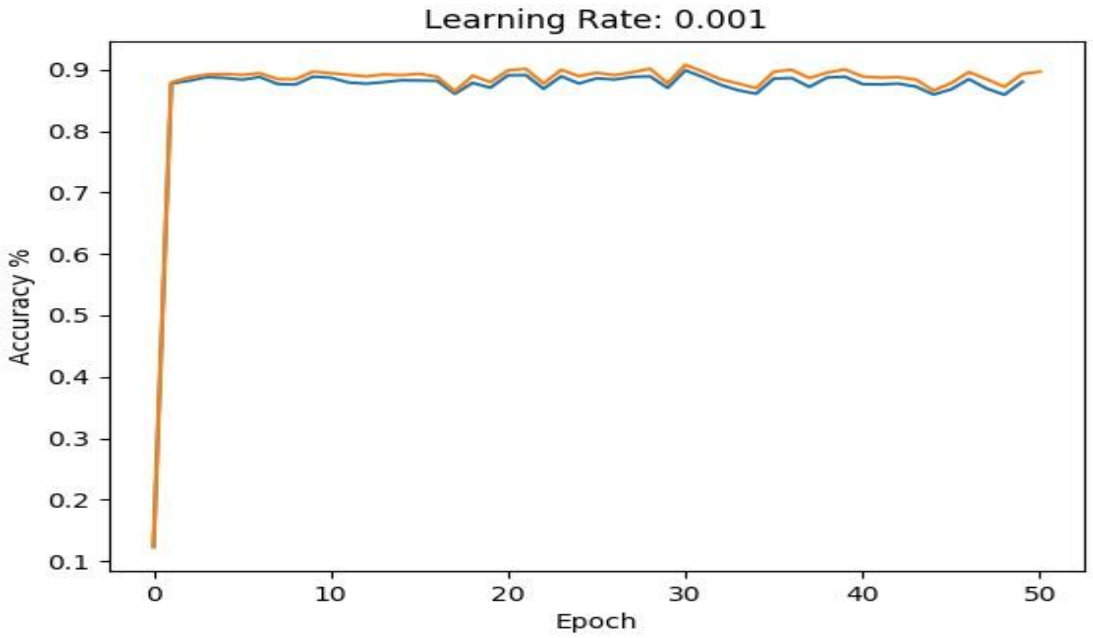
For learning rate = 0.01

Graph:



For learning rate = 0.001

Graph:



Question 2 b: Confusion matrix on the test set, after training has been completed

For learning rate = 0.1

Confusion Matrix:

```
[[ 959    0    1    1    1    5    8    3    1    1]
 [   0 1117    2    3    2    1    4    1    5    0]
 [  11   32  731  152   10   10   36    6   31   13]
 [   4    1    9  935    3   24    3   10    7   14]
 [   3    2    1    4  870    0   21    1    4   76]
 [  11    3    0   59   12  742   25    4   24   12]
 [  10    5    2    1    5   15  918    1    0    1]
 [   2   14   13   22   14    2    0  832    2  127]
 [  16   27    3  139   24   54   29    3  614   65]
 [   6    7    0   16   15    6    2    3    1  953]]
```

For learning rate = 0.01

Confusion Matrix:

```
[[ 958    0    2    2    0   10    2    3    1    2]
 [   0 1115    4    5    1    1    3    1    5    0]
 [   8   25  824  110    5   12   13    8   20    7]
 [   3    1   11  930    2   39    1   11    8    4]
 [   1    2    7   10  828    2   11    3   17  101]
 [  10    3    3   48    8  769   15    5   22    9]
 [  12    3   19    3    4   36  879    0    1    1]
 [   1   10   21   35    7    4    0  898    9   43]
 [  15   30   10  172   10   69   10    6  632   20]
 [   7    7    1   20   11   16    2   16    5  924]]
```

For learning rate = 0.001

Confusion Matrix:

```
[[ 967    0    2    2    1    2    1    3    2    0]
 [   0 1098    4    8    1    2    3    1   18    0]
 [  17   15  840  104   11    4    7    4   26    4]
 [   6    1   21  936    1   21    2    9   10    3]
 [   2    2   10    7  908    1    6    2   11   33]
 [  19    2    3   62   14  738    9    3   38    4]
 [  24    4   22    2    6   30  865    1    4    0]
 [   1    9   28   32   15    3    0  890    7   43]
 [  19   10   14  147   13   26    7    6  723    9]
 [   7    3    2   21   35    6    1   18   19  897]]
```

Question 2 c: Short discussion of confusion matrix: which digits are classified most accurately, and which digits tend to be confused with one another?

I have added the digits 0-9 for rows and columns for more making it more clear.

For learning rate = 0.1

```
     0     1    2    3    4    5    6    7    8    9
0 [ 959    0    1    1    1    5    8    3    1    1]
1 [   0 1117    2    3    2    1    4    1    5    0]
2 [  11   32  731  152   10   10   36    6   31   13]
3 [   4    1    9  935    3   24    3   10    7   14]
4 [   3    2    1    4  870    0   21    1    4   76]
5 [  11    3    0   59   12  742   25    4   24   12]
6 [  10    5    2    1    5   15  918    1    0    1]
7 [   2   14   13   22   14    2    0  832    2  127]
8 [  16   27    3  139   24   54   29    3  614   65]
9 [   6    7    0   16   15    6    2    3    1  953]]
```

As we can see from the matrix,

Digit 0 is classified 959 times accurately but mostly confused by digit 6 (i.e. 8 times) and 5 (i.e. 5 times).

Digit 1 is classified 1117 times most accurately but mostly confused by digit 8 (i.e. 5 times).

Digit 2 is classified 731 times accurately but mostly **confused by digit 3 (i.e. 152 times**).

Digit 3 is classified 935 times accurately but mostly confused by digit 5 (i.e. 24 times) and 9 (14 times).

Digit 4 is classified 870 times accurately but mostly confused by digit 9 (i.e. 76 times).

Digit 5 is classified 742 times accurately but mostly confused by digit 3 (i.e. 59 times) and 6 (25 times).

Digit 6 is classified 918 times accurately but mostly confused by digit 5 (i.e. 15 times).

Digit 7 is classified 832 times accurately but mostly confused by digit 9 (i.e. 127 times).

Digit 8 is classified 614 times accurately but mostly **confused by digit 3 (i.e. 139 times).**

Digit 9 is classified 953 times accurately but mostly confused by digit 3 (i.e. 16 times).

<u>For learning rate = 0.01</u>

```
     0     1     2     3     4     5     6     7     8     9
0 [ 958     0     2     2     0    10     2     3     1     2]
1 [   0  1115     4     5     1     1     3     1     5     0]
2 [   8    25   824   110     5    12    13     8    20     7]
3 [   3     1    11   930     2    39     1    11     8     4]
4 [   1     2     7    10   828     2    11     3    17   101]
5 [  10     3     3    48     8   769    15     5    22     9]
6 [  12     3    19     3     4    36   879     0     1     1]
7 [   1    10    21    35     7     4     0   898     9    43]
8 [  15    30    10   172    10    69    10     6   632    20]
9 [   7     7     1    20    11    16     2    16     5   924]]
```

In similar fashion as above, we can observe that Digit 1 is most accurately classified as 1 itself 1115 times. Digit 8 and 2 are mostly confused with digit 3 by 172 times and 110 times respectively.

For learning rate = 0.001

```
     0     1     2     3     4     5     6     7     8     9
0 [ 967     0     2     2     1     2     1     3     2     0]
1 [   0  1098     4     8     1     2     3     1    18     0]
2 [  17    15   840   104    11     4     7     4    26     4]
3 [   6     1    21   936     1    21     2     9    10     3]
4 [   2     2    10     7   908     1     6     2    11    33]
5 [  19     2     3    62    14   738     9     3    38     4]
6 [  24     4    22     2     6    30   865     1     4     0]
7 [   1     9    28    32    15     3     0   890     7    43]
8 [  19    10    14   147    13    26     7     6   723     9]
9 [   7     3     2    21    35     6     1    18    19   897]]
```

In similar fashion as above, we can observe that Digit 1 is most accurately classified as 1 itself 1098 times. Digit 8 and 2 are mostly confused with digit 3 by 147 times and 104 times respectively.

Question-3 : Short discussion comparing results of different learning rates. Did you see any difference in the results when using different learning rates?

While training the perceptron, learning rate plays a significant role. If it is too low, we cannot train our model sufficient enough. And greater accuracy will not be achieved. On the other hand, if we keep learning rate very high, it will, at the end, keep on oscillating and will not be able to reach the maximum value. Thus, we have to choose this learning rate very carefully in order to make our perceptron work well.

Comparing the results we got for three different learning rates. We got oscillations for all 3 rates. But oscillations in the rate 0.001 are getting pacified as compared to the rate 0.01. We can say these are oscillations because there is not much difference between the training accuracies and test accuracies i.e. the perceptron is generalizing the data.