Lawrence Hsu

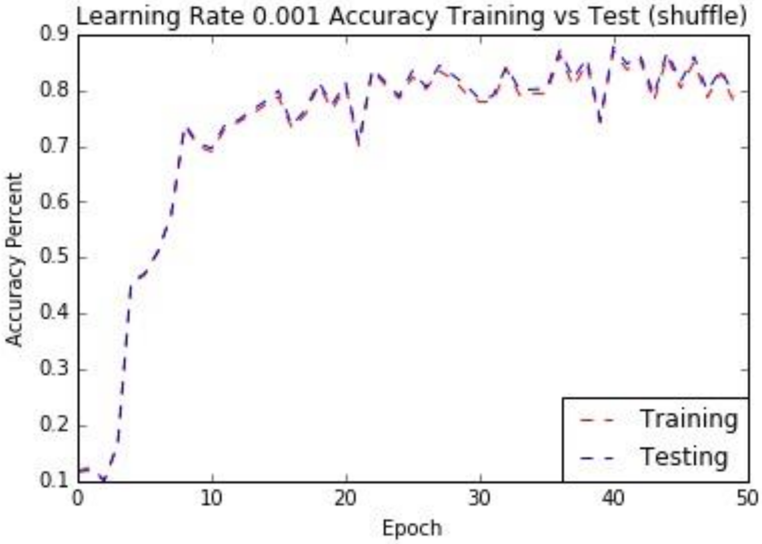
Project Description:

This experiment was to train ten different perceptrons where each perceptrons was trained to one particular number (0-9). There were 60000 training examples and 10000 testing examples. Each example had 785 inputs which each perceptron would take those inputs and give one output. After each calculation of the output, weights were updated. This was repeated for 50 epochs. Furthermore, three learning rates were used to see the effects of accuracy of the classification of the test set.

Disclaimer: For the confusion matrix, I determine which numbers are easily misclassified based on the overall row. Secondly, I do not use the diagonal values determine which numbers were easily classified because I do not know if there were equal amounts of examples of each number.

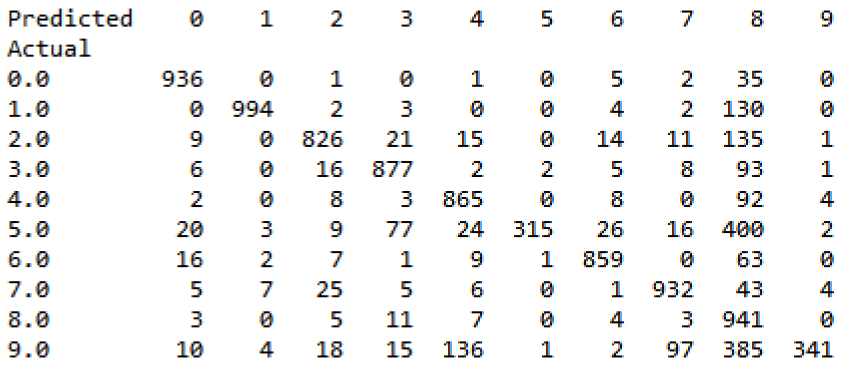
Learning Rate 0.001:

I see that the curves are oscillations as the accuracy increases and decreases within range of 0.7 to 0.9. There is no indication of overfitting because the testing curve would have a lower accuracy as the number of epoch increases while the training curve continues to increase accuracy.



(I am basing this off of the overall row). It seems like 9 is easily mistaken as other numbers across all numbers(particularly 8 and 4). Number 5 is another number that gets easily mistaken as well (particularly with 8). Seems like 0,3, and 6 are correctly classified based on low count from the other items in that row.

Figure 10. Graph of the training and test set accuracy with learning rate of 0.001



```
Predicted     0    1    2    3    4    5    6    7    8    9
Actual
0.0         936    0    1    0    1    0    5    2   35    0
1.0           0  994    2    3    0    0    4    2  130    0
2.0           9    0  826   21   15    0   14   11  135    1
3.0           6    0   16  877    2    2    5    8   93    1
4.0           2    0    8    3  865    0    8    0   92    4
5.0          20    3    9   77   24  315   26   16  400    2
6.0          16    2    7    1    9    1  859    0   63    0
7.0           5    7   25    5    6    0    1  932   43    4
8.0           3    0    5   11    7    0    4    3  941    0
9.0          10    4   18   15  136    1    2   97  385  341
```

Learning Rate 0.01:

I see that the curves are oscillations as the accuracy increases and decreases within range of 0.7 to 0.8. Compared to learning rate 0.001, there are less oscillations. There is no indication of overfitting because the testing curve would have a lower accuracy as the number of epoch increases while the training curve continues to increase accuracy.

Figure 7. Confusion matrix of learning curve 0.001 with shuffling

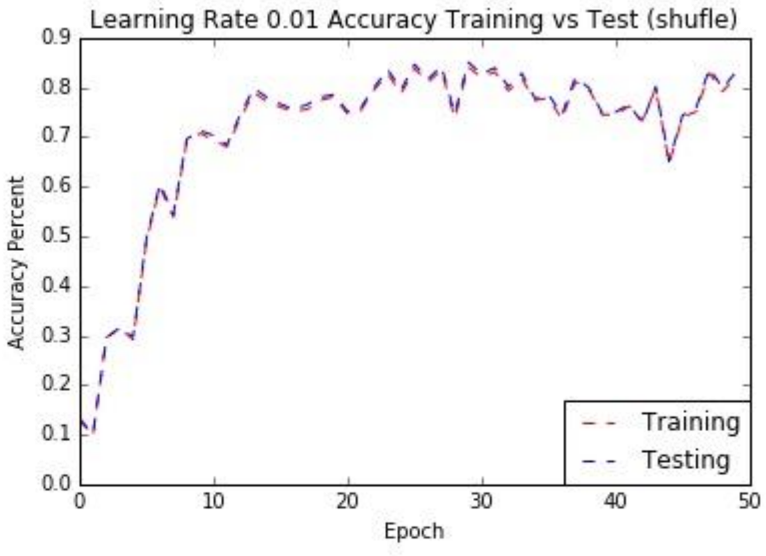
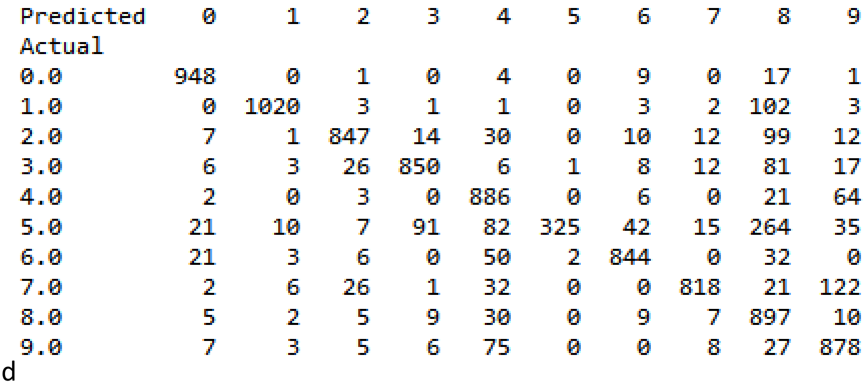


Figure 11. Graph of the training and test set accuracy with learning rate of 0.01

7 and 5 tend to get confused with other numbers. 7 particularly gets classified as 9 and 5 tends to get classified as 8. 0, 8, and 9 seems to be easily classified correctly based on the low count from the other items in that row.



```
Predicted    0     1    2    3    4    5    6    7    8    9
Actual
0.0        948     0    1    0    4    0    9    0   17    1
1.0          0  1020    3    1    1    0    3    2  102    3
2.0          7     1  847   14   30    0   10   12   99   12
3.0          6     3   26  850    6    1    8   12   81   17
4.0          2     0    3    0  886    0    6    0   21   64
5.0         21    10    7   91   82  325   42   15  264   35
6.0         21     3    6    0   50    2  844    0   32    0
7.0          2     6   26    1   32    0    0  818   21  122
8.0          5     2    5    9   30    0    9    7  897   10
9.0          7     3    5    6   75    0    0    8   27  878
```

asd

Figure 8. Confusion matrix of learning curve 0.01 with shuffling

Learning Rate 0.1:

I see that the curves are oscillations as the accuracy increases and decreases within range of 0.7 to 0.9. Compared to learning rate 0.001 and 0.01, there are more oscillations. There is no indication of overfitting because the testing curve would have a lower accuracy as the number of epoch increases while the training curve continues to increase accuracy.

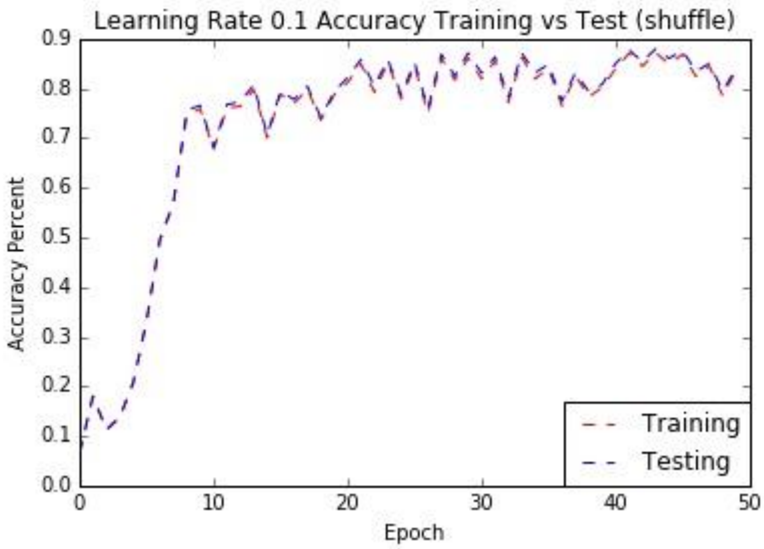
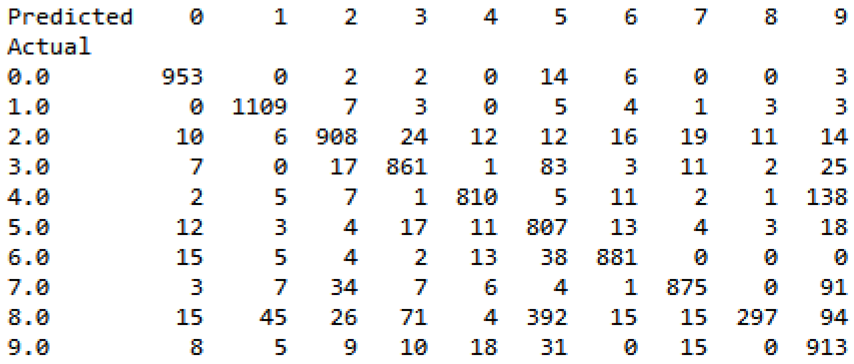


Figure 12. Graph of the training and test set accuracy with learning rate of 0.1

8,7,and 4 tend to get misclassified. 8 in particularly gets misclassified as 5, 7 gets misclassified the most as 9,and 4 tends to get misclassified as 9. 0,1,6, and 9 get classified correctly.



```
Predicted    0     1    2    3    4    5    6    7    8    9
Actual
0.0        953     0    2    2    0   14    6    0    0    3
1.0          0  1109    7    3    0    5    4    1    3    3
2.0         10     6  908   24   12   12   16   19   11   14
3.0          7     0   17  861    1   83    3   11    2   25
4.0          2     5    7    1  810    5   11    2    1  138
5.0         12     3    4   17   11  807   13    4    3   18
6.0         15     5    4    2   13   38  881    0    0    0
7.0          3     7   34    7    6    4    1  875    0   91
8.0         15    45   26   71    4  392   15   15  297   94
9.0          8     5    9   10   18   31    0   15    0  913
```

Figure 9. Confusion matrix of learning curve 0.1 with shuffling

Similarity/Differences of different learning rates:

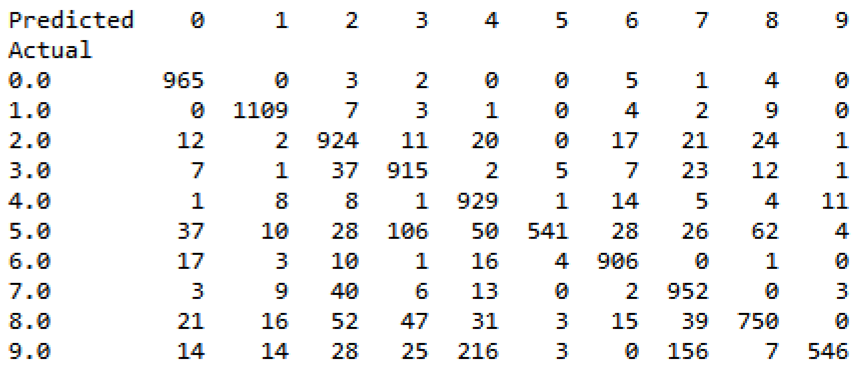
Overall, I do not see any differences in accuracy between the different learning rates. As the accuracy tend to hover around 0.7 to 0.8 roughly. However, some learning rates tend to higher hover at higher range within the overall rate. This suggests that learning rate parameter is important because the value can either overfit the data, hence a lower accuracy. Learning rate 0.1 tends to oscillate more frequently than the other two rates. This might be due overshooting the target more often than the other two rates. Different rates tend to misclassify different numbers. I also noticed different rates reached their maximum accuracy at different epoch. Learning rate 0.001 reached max accuracy around epoch 38, learning rate 0.01at epoch 22, and learning rate 0.1 at epoch 44. Lastly, the initial growth in accuracy is smoother than in 0.1 rate than the other two rates. This might be due to learning rate allowing it to skip noise rather than consider it rather re-adjust.

PS: I also tried to see if not shuffling the training set to see if it would affect the accuracy (see appendix). It seems like it do not and that but be due to the large dataset. If I remember correctly, as the dataset gets larger, the error tends to smooth out.

Instruction on Code:

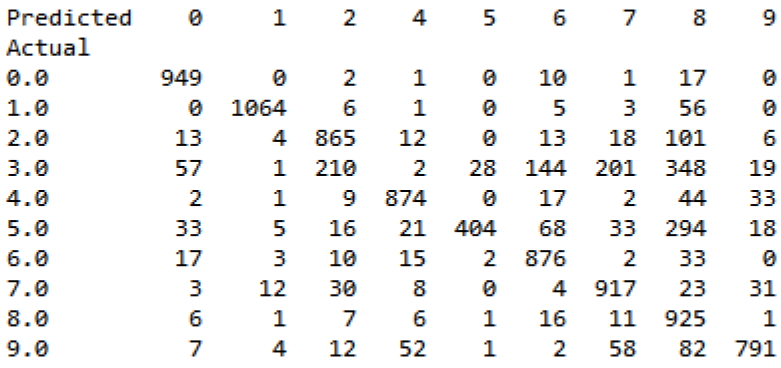
1. Initialize the libraries and functions
2. Run the codes that has a comment header saying, “Codes to run…”
    a. Note: do not run everything at once, only as pairs as indicated by the space between codes otherwise the figures will lump together into one jpg.
    b. If you wish to see the individual accuracy list of the training and test of each learning rate, change the graphw and grapht to whatever you want.
    c. (optional) if you wish to see the non shuffle training set, go into runeverything fxn and remove random.shuffle(ori)

Appendix.



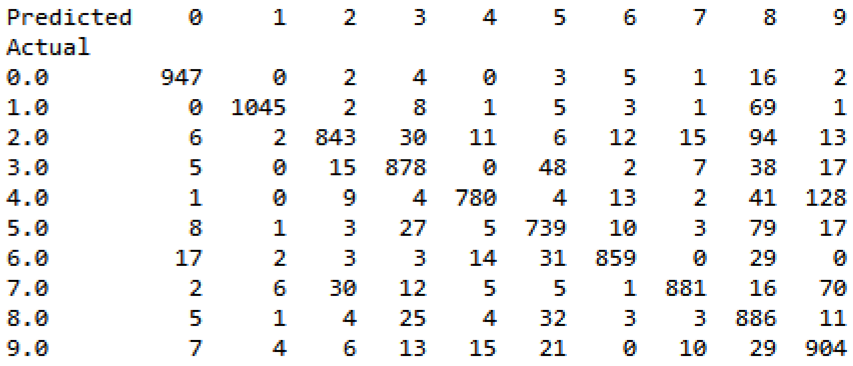
```
Predicted    0     1    2    3    4    5    6    7    8    9
Actual
0.0        965     0    3    2    0    0    5    1    4    0
1.0          0  1109    7    3    1    0    4    2    9    0
2.0         12     2  924   11   20    0   17   21   24    1
3.0          7     1   37  915    2    5    7   23   12    1
4.0          1     8    8    1  929    1   14    5    4   11
5.0         37    10   28  106   50  541   28   26   62    4
6.0         17     3   10    1   16    4  906    0    1    0
7.0          3     9   40    6   13    0    2  952    0    3
8.0         21    16   52   47   31    3   15   39  750    0
9.0         14    14   28   25  216    3    0  156    7  546
```

Fig 1. Confusion matrix of learning curve 0.001 with no shuffling



```
Predicted    0     1    2    4    5    6    7    8    9
Actual
0.0        949     0    2    1    0   10    1   17    0
1.0          0  1064    6    1    0    5    3   56    0
2.0         13     4  865   12    0   13   18  101    6
3.0         57     1  210    2   28  144  201  348   19
4.0          2     1    9  874    0   17    2   44   33
5.0         33     5   16   21  404   68   33  294   18
6.0         17     3   10   15    2  876    2   33    0
7.0          3    12   30    8    0    4  917   23   31
8.0          6     1    7    6    1   16   11  925    1
9.0          7     4   12   52    1    2   58   82  791
```

Fig 2. Confusion matrix of learning curve 0.01 with no shuffling



```
Predicted    0     1    2    3    4    5    6    7    8    9
Actual
0.0        947     0    2    4    0    3    5    1   16    2
1.0          0  1045    2    8    1    5    3    1   69    1
2.0          6     2  843   30   11    6   12   15   94   13
3.0          5     0   15  878    0   48    2    7   38   17
4.0          1     0    9    4  780    4   13    2   41  128
5.0          8     1    3   27    5  739   10    3   79   17
6.0         17     2    3    3   14   31  859    0   29    0
7.0          2     6   30   12    5    5    1  881   16   70
8.0          5     1    4   25    4   32    3    3  886   11
9.0          7     4    6   13   15   21    0   10   29  904
```

Fig 3. Confusion matrix of learning curve 0.1 with no shuffling

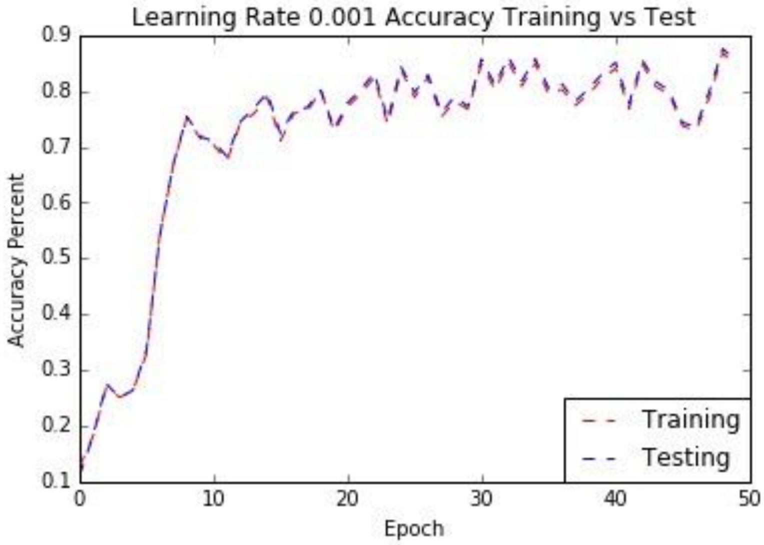


Fig 4. Graph of the training and test set accuracy with learning rate of 0.001 (no shuffling)

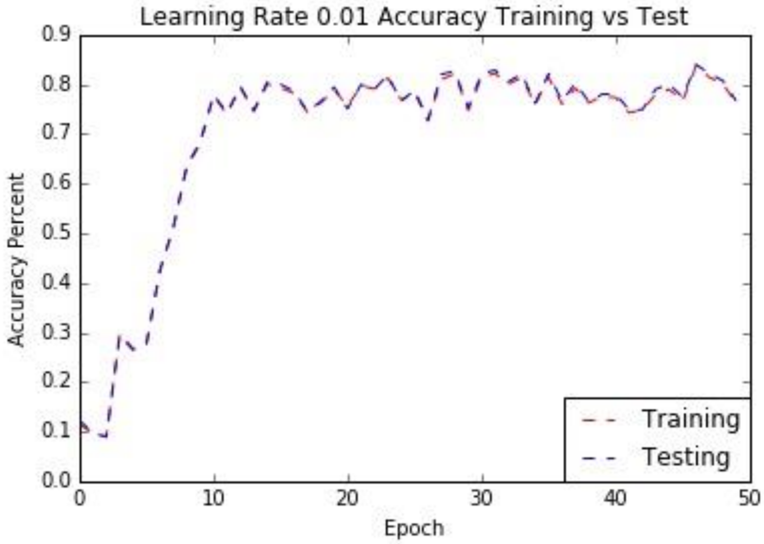


Fig 5. Graph of the training and test set accuracy with learning rate of 0.01 (no shuffling)

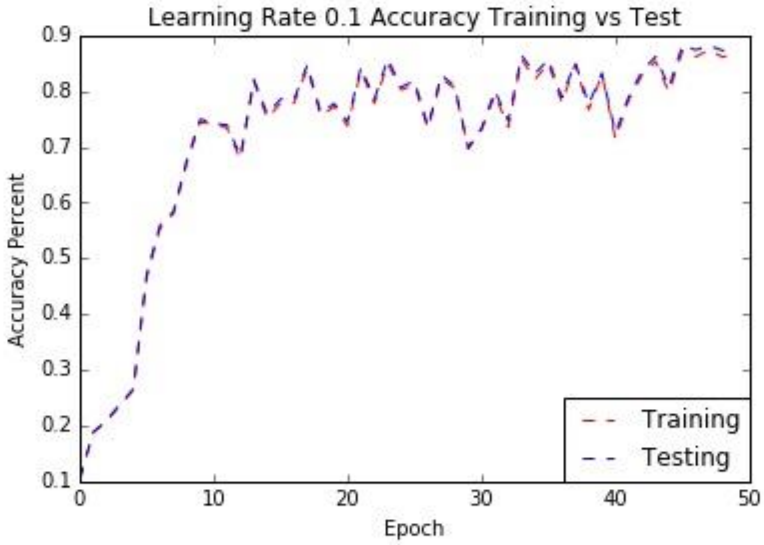


Fig 6. Graph of the training and test set accuracy with learning rate of 0.1 (no shuffling)