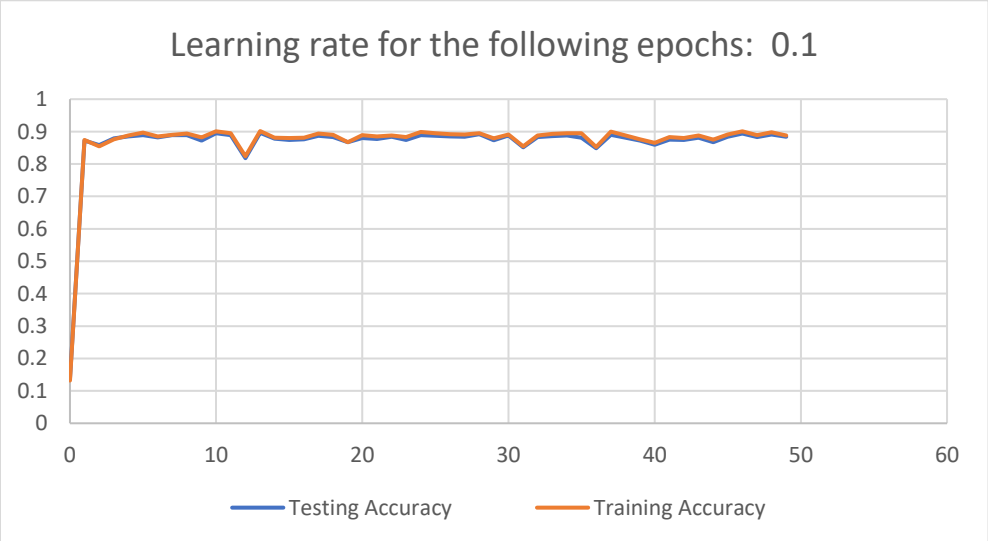
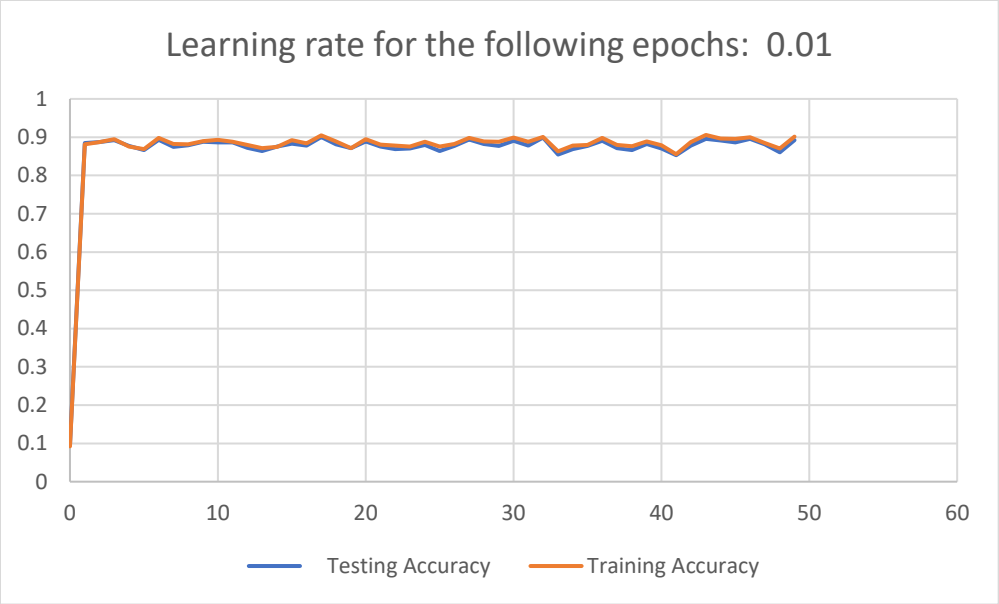
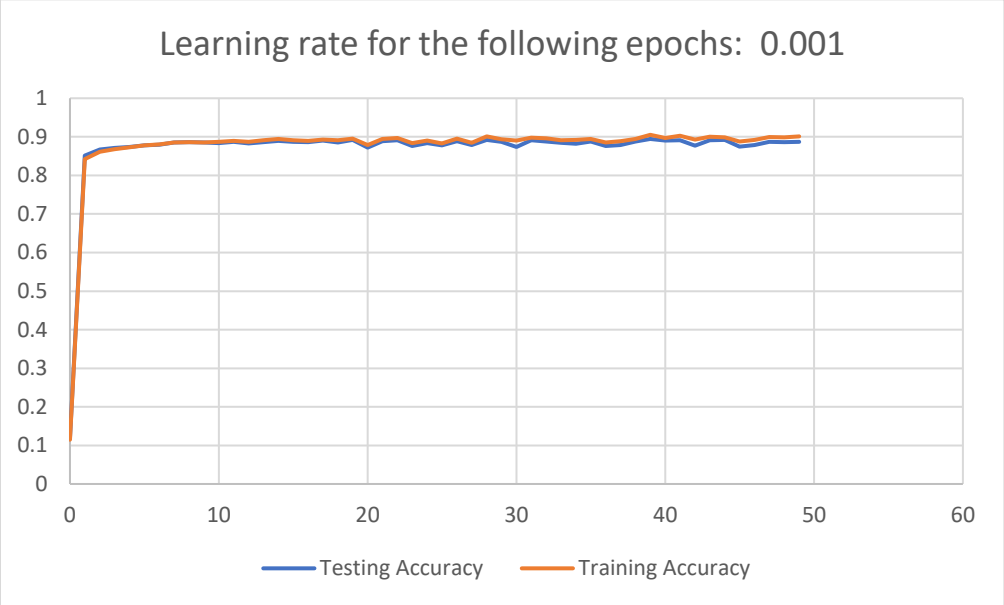
CS 545 Machine Learning
Homework 1 Perceptrons
Haomin He

1. A one-paragraph description of the experiment.
I build a perceptron class which includes all the features and functionality a perceptron has. For instance, different perceptron has different input vectors and weights. Choose small random initial weights that's between -0.05 and 0.05. Calculates dot product of input vector and its weight. Also need to update weight if target value is not equal to output value. Output units $t - y$ could be zero, and thus the weights to that output unit would not be updated. In main python file, I declare ten perceptrons to represent 0-9, the ten target digits. Load training and testing csv files into two big arrays. Except the first value of each row (target value), scale the rest of value to be between 0 and 1 (divide each value by 255). Insert bias unit one into each row. Loop through perceptrons and calculate $w * x^k$. The output with the highest value of $w * x^k$ is the prediction for the training example. Based on this, calculate accuracy (number of right prediction cases divide total cases). Use two-dimensional array to build confusion matrix. Increase matrix cell value according to target and prediction values.

2. For each learning rate:
• 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 overfitting.







As you can see from the above, testing accuracy and training accuracy are overfitting with each other.

• Confusion matrix on the test set, after training has been completed.

Learning rate: 0.1									
Confusion	Matrix:								
959	0	1	4	0	6	4	3	1	2
0	1125	2	1	0	1	4	1	1	0
11	32	863	36	10	7	9	18	43	3
2	6	15	920	2	21	5	10	15	14
2	9	7	2	889	1	8	9	7	48
11	7	4	62	5	729	16	11	32	15
10	3	7	5	6	21	901	0	4	1
2	19	16	11	7	1	1	945	4	22
10	46	6	40	11	35	7	17	791	11
8	11	2	17	65	11	0	54	12	829

Learning rate: 0.01									
Confusion	Matrix:								
967	0	0	2	0	3	5	2	1	0
0	1103	3	2	0	2	5	1	19	0
16	28	860	18	4	4	19	22	56	5
4	4	15	916	0	27	7	13	20	4
10	8	8	3	836	0	18	9	22	68
12	2	2	91	8	704	24	6	38	5
12	5	3	1	2	13	912	0	10	0
4	13	15	11	3	3	1	945	8	25
14	22	5	31	6	35	11	14	832	4
11	9	0	18	23	16	0	45	24	863

Learning rate: 0.001									
Confusion	Matrix:								
951	0	4	4	2	11	2	1	3	2
0	1075	11	17	0	7	2	3	20	0
7	4	884	44	8	9	12	15	42	7
6	0	18	930	0	19	1	12	19	5
2	1	15	14	830	6	2	12	45	55
12	2	5	55	7	756	8	8	32	7
14	2	25	5	7	46	838	1	18	2
3	4	15	19	5	6	0	938	6	32
11	5	13	99	7	38	3	11	782	5
9	5	1	33	12	12	0	44	49	844

• Short discussion of confusion matrix: which digits are classified most accurately, and which digits tend to be confused with one another?
Digit 1 and digit 0 are classified most accurately. Digit 5 and digit 8 tend to be confused with one and another.

3. Short discussion comparing results of different learning rates. Did you see any difference in the results when using different learning rates?
The learning rate affects how quickly a model can converge to a local optimal. The smaller learning rate takes longer to converge, but result is more accurate. As you can see from learning rate 0.001 accuracy diagram, it is more overfitting to the test dataset compare to other learning rates.