CS4442b/9542b: Artificial Intelligence II, Winter 2018

Assignment 1: Hand Written Digit Classification

Name: Harry Wu

Problem 3:

b)

k = 1; err = 0.0835
k = 3; err = 0.0790
k = 5; err = 0.0850
k = 7; err = 0.0885

Error decreases then increases again as we increase k. Having a small k is sensitive to noise and a k that is too large may underfit the data by over-smoothing. It appears that 3 is the optimal k-value for this data set.

c)

173	0	0	0	0	0	1	1	0	0
0	234	0	0	0	0	0	0	0	0
4	9	192	0	1	0	3	8	2	0
0	0	0	188	0	6	2	4	3	4
0	2	0	0	199	0	3	1	0	12
2	1	1	5	2	158	3	1	2	4
3	1	0	0	2	1	171	0	0	0
1	12	3	0	2	1	0	178	0	8
3	1	1	9	3	6	2	3	160	4
0	0	0	2	5	4	0	4	2	177

The classes (5,10) and (8,2) are most frequently confused with each other. These are the digit pairs of (4, 9) and (1,7) which follows the observation that 4 closely resembles 9 and 1 closely resembles 7. However, this confusion is not symmetric suggesting that the confusion is not made in the other direction (at least not to that extent).

Problem 5

b)

Iterations = 100:			
	Training Error	=	0.2806
	Test Error	=	0.2782
Iterations = 1000:			
	Training Error	=	0.1948
	Test Error	=	0.2231
Iteration = 10000:			
	Training Error	=	0.1759
	Test Error	=	0.1654

From the errors observed, it seems that the logistic regression classifier is working properly. Training error is greater than test error in all three cases. It can be seen that error decreases as we increase the number of iterations. This is due to the fact that the classifier generates solutions with a random search. By generating more random solutions, it will have a better chance of finding a better solution.

Problem 6

b)

Training Error	=	0.0356
Test Error	=	0.0627

The error is much lower than the errors obtained in 5b. With this logistic regression classifier gradient descent is used instead of relying on random search meaning that each iteration will almost guarantee improvement over the last, particularly for convex loss functions.

Problem 8

b)

168	0	1	0	0	0	4	0	1	1
0	224	1	1	2	1	2	0	2	1
1	8	180	12	0	0	9	2	6	1
1	0	2	179	1	10	4	3	5	2
0	2	1	2	187	0	8	2	2	13
6	1	1	10	6	128	4	1	17	5
4	1	6	1	2	2	157	3	2	0
1	6	12	6	6	3	0	150	0	21
1	3	10	15	4	6	5	2	138	8
1	1	1	4	8	4	1	4	5	165

Training Error	=	0.0620
Test Error	=	0.1620

The classes (8,10) are often confused for each other. These are the digits (7,9). This is different from the most confused values in Problem 3c. However, with this classifier, we also see a high degree of confusion for some other digit pairs such as (4, 9), (5,8), and (3,8). From a quick glance, these numbers do show resemblances to each other as well. However, it seems that this classifier performs worse the the k-NN classifier from Problem 3c.

Problem 9

b)

170	0	0	1	0	0	3	0	0	1
0	229	0	1	0	0	2	0	2	0
1	2	188	3	1	0	8	4	9	3
1	1	6	175	0	9	1	7	6	1
1	1	0	0	197	1	4	1	2	10
4	0	3	9	2	145	4	2	7	3
5	1	2	0	5	3	158	3	1	0
1	4	11	4	1	0	0	172	0	12
1	3	3	11	5	10	2	2	148	7
1	0	0	5	9	3	0	8	1	167

Training Error	=	0.0484
Test Error	=	0.1255

This classifier shows a little improvement in performance compared to the one in Problem 8b. We can see that both training error and test error is lower when using this classifier. Some digit pairs that are often confused are (7,9), (8,3), (7,2), and (4,0).

Problem 10

c)

172	0	0	0	0	0	2	0	0	1
0	230	0	0	1	0	2	0	1	0
2	2	199	1	1	0	6	5	2	1
0	0	2	191	0	3	1	4	4	2
0	0	1	0	198	0	5	2	1	10
3	0	0	5	2	159	2	3	3	2
3	1	1	0	5	2	165	0	1	0
1	6	10	1	2	0	0	178	0	7
1	2	1	7	4	4	2	3	165	3
1	0	0	5	9	2	0	3	2	172

Training Error	=	0.0593
Test Error	=	0.0855

Using the neural network and p10b to apply the trained network to the test data, a significant improvement can be seen. Training error and test error are much improved over what is obtained in Problem 8b and Problem 9b. The most common confusions are the digit pairs of (4, 9), (7,2), and (9,4).

d)

172	0	1	0	0	0	2	0	0	0
0	228	0	0	1	0	3	0	2	0
5	3	200	3	1	0	4	1	1	1
0	0	2	189	0	7	1	3	4	1
0	0	2	0	200	1	2	2	1	9
4	0	0	8	2	160	1	1	2	1
3	2	0	0	6	4	161	0	1	1
0	6	8	1	1	0	0	180	0	9
1	2	3	4	2	3	1	4	167	5
1	1	0	3	5	2	0	1	3	178

Training Error	=	0.0460
Test Error	=	0.0680

H = [88,21]

Regularization Strength = 0.7

The results in part (b) are already performing very well; Therefore, it is difficult to find any huge improvement. However, adding a hidden layer and tuning the number of units in each layer and the regularization strength can yield slight improvements such as what I accomplished using the parameters listed above. Digit pairs that still tend to get confused are (4, 9), (7,9), and (2,7).