Kobe Davis (901199860)

Prof. Doliotis

CS 445

24 January 2020

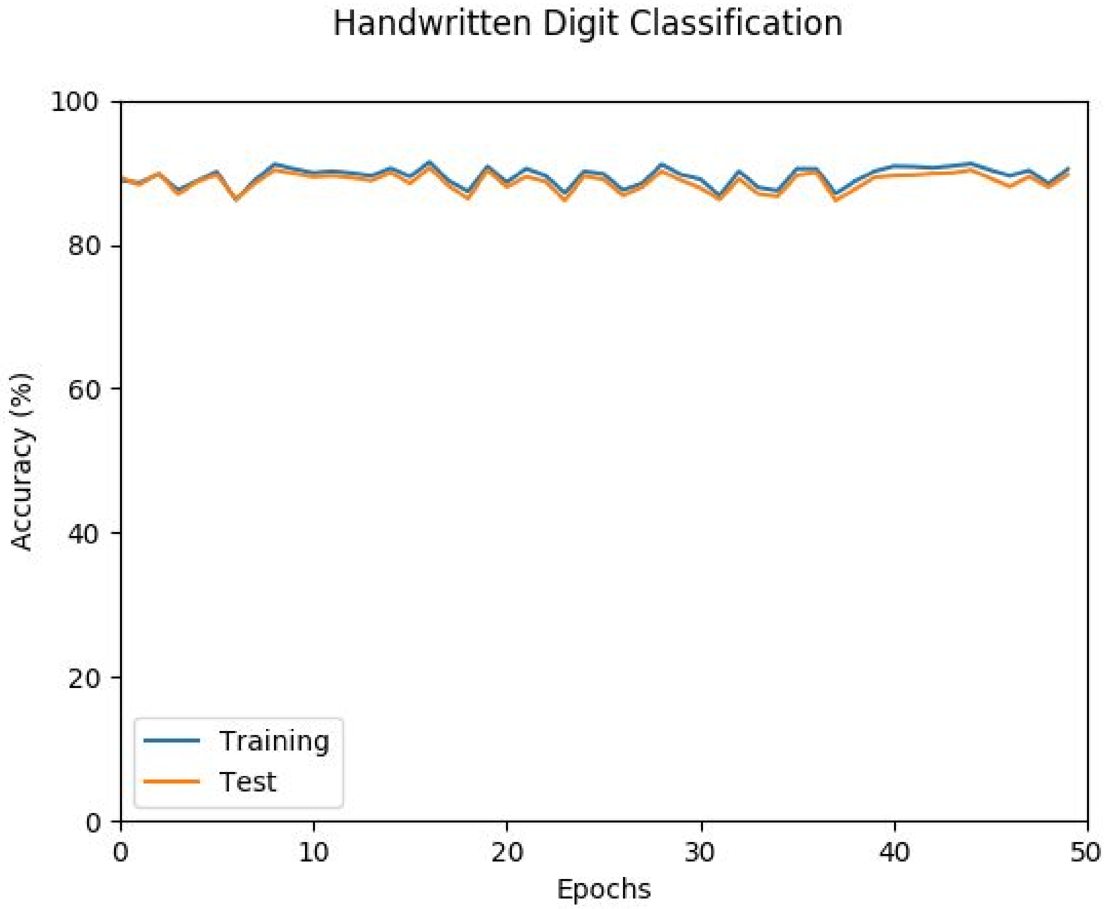
Assignment 1: Perceptron Learning

Introduction

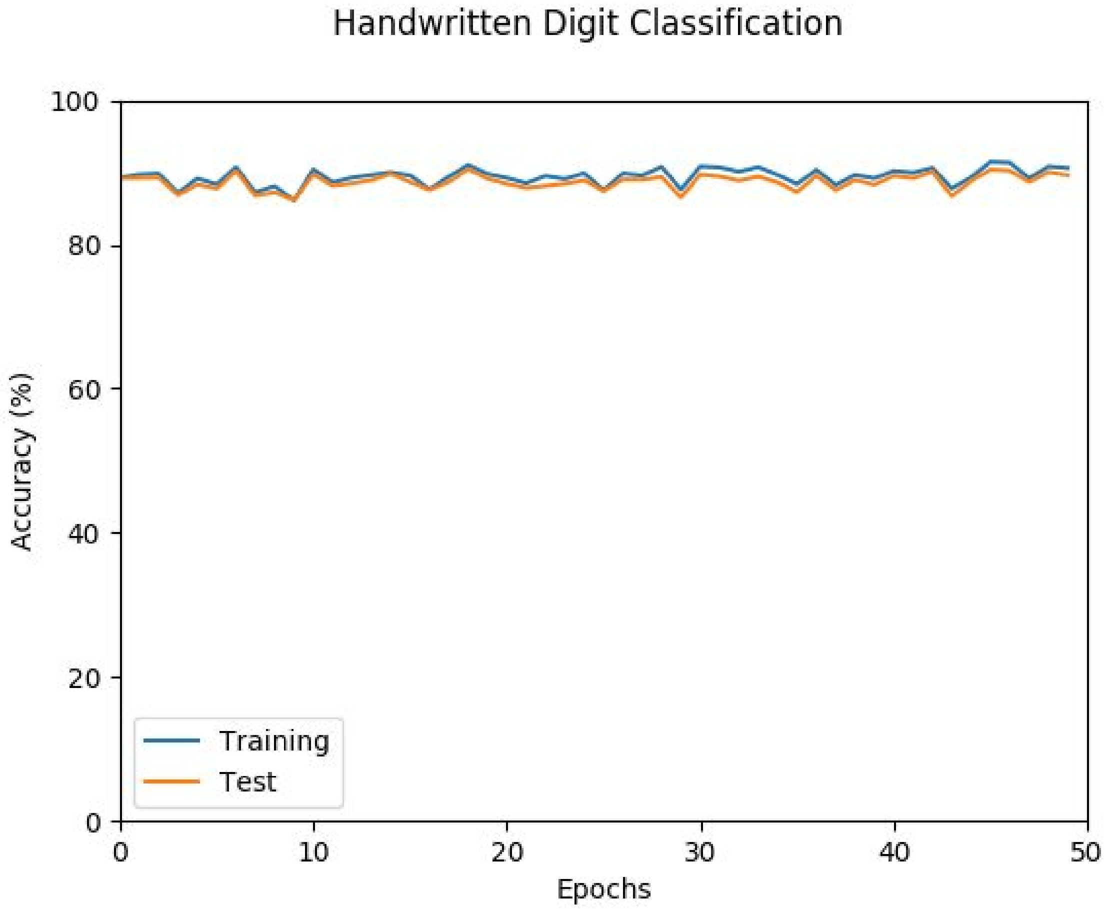
This assignment involved training 10 perceptrons to recognize hand-written digits in the form of 28x28 pixels. First we scaled the pixel values between 0 and 1, and we shuffled the input data. Following this, we input the data into the network and if the network output was incorrect, we updated the weights. Running a cycle over the training data is called an epoch; after each epoch we computed the accuracy of our network over both the training data and the test data. Lastly, we generated a confusion matrix demonstrating visually where and how often our networks misclassified the hand-written digits.

Plots

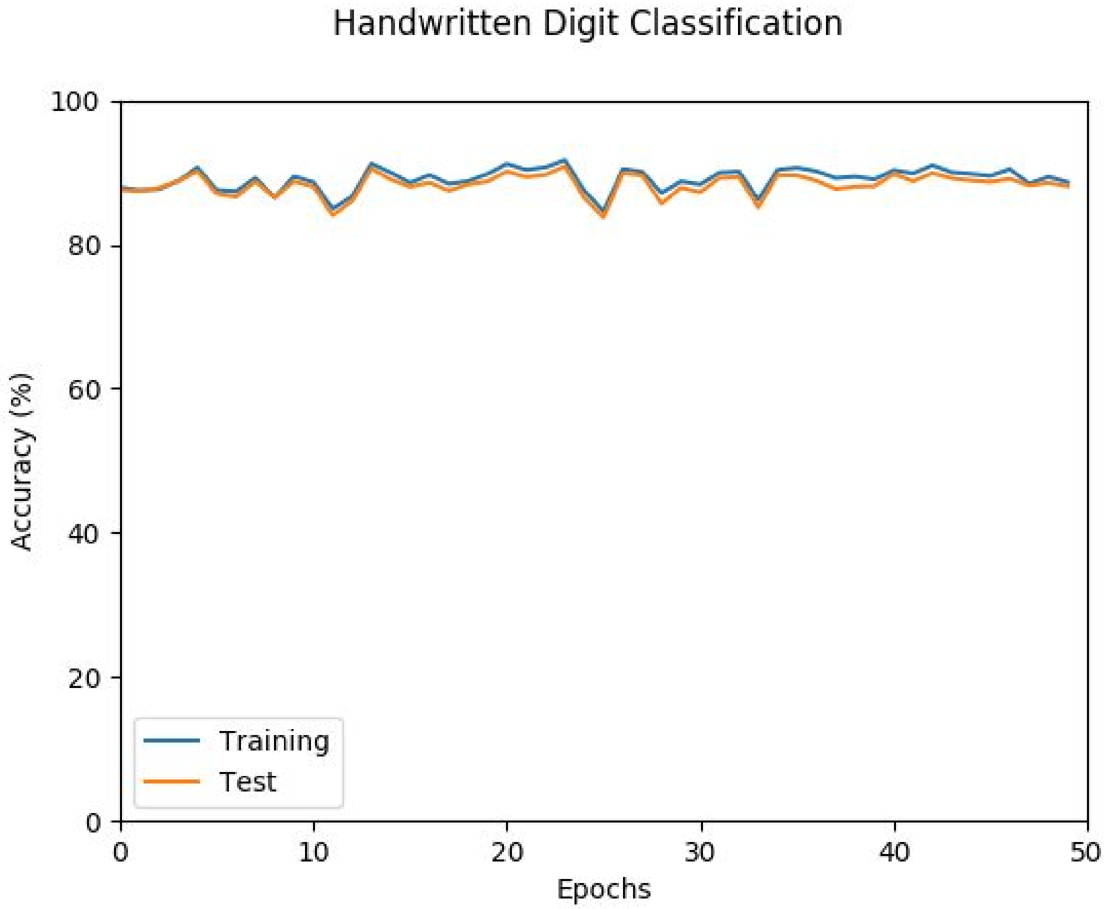
Learning Rate .1:



Learning Rate .01:



Learning Rate .001:



Confusion Matrices

Learning Rate .1:

	0	1	2	3	4	5	6	7	8	9
0	950	0	6	5	3	4	6	2	2	2
1	0	1115	3	7	0	3	3	1	3	0
2	5	10	948	23	10	2	13	7	11	3
3	3	2	27	933	1	33	1	4	4	2
4	2	3	8	9	870	0	16	2	4	68
5	8	3	9	47	11	773	15	4	17	5
6	10	2	13	3	4	29	894	0	3	0
7	0	11	29	46	11	4	0	865	2	60
8	4	26	44	109	10	51	11	5	696	18
9	6	7	2	30	13	9	1	10	5	926

Learning Rate .01:

	0	1	2	3	4	5	6	7	8	9
0	952	0	3	2	4	6	5	1	6	1
1	0	1140	6	1	0	1	3	1	9	0
2	4	8	928	8	14	4	8	10	43	5
3	6	2	41	848	7	54	0	17	31	4
4	1	5	6	2	939	0	4	3	13	9
5	1	2	12	28	27	756	5	9	39	4
6	10	2	18	1	40	34	809	1	43	0
7	1	7	18	7	9	2	1	947	13	23
8	6	11	13	8	24	29	3	12	865	3
9	7	6	1	9	89	7	0	60	26	904

Learning Rate .001:

	0	1	2	3	4	5	6	7	8	9
0	929	0	29	3	2	2	8	2	3	2
1	0	1112	13	0	0	1	3	1	3	1
2	2	6	973	0	10	2	9	7	19	4
3	3	3	167	693	6	47	2	18	33	38
4	1	3	13	1	900	0	7	3	9	45
5	9	4	43	18	15	710	17	8	53	15
6	7	3	31	0	6	15	892	2	1	1
7	0	8	27	3	11	1	0	877	4	97
8	4	10	78	4	11	14	11	10	816	16
9	5	4	5	1	51	4	1	15	16	907

Discussion

While all learning rates managed to classify the hand-written digits fairly well, taking a closer look we can see that eights and fives are dragging behind, albeit only slightly, the other numbers in accuracy. In the .001 confusion matrix three seems to stand out also, as being misclassified fairly often. While I don’t why this might be, my guess is that curved lines may be more difficult for the network to classify, verses the simpler, more angled one and 4. Unfortunately, this line of thought doesn’t hold up when observing the success rate in 2 and 9.

While the plots don’t do the best job of showing minute detail, while working on the program I observed the smaller learning consistently reaching higher accuracy than the other two learning rates. Perhaps this is due to the network learning “too much” before it is able to view the rest of the dataset, which the smaller learning rate allows it to do. Though this is just speculation.