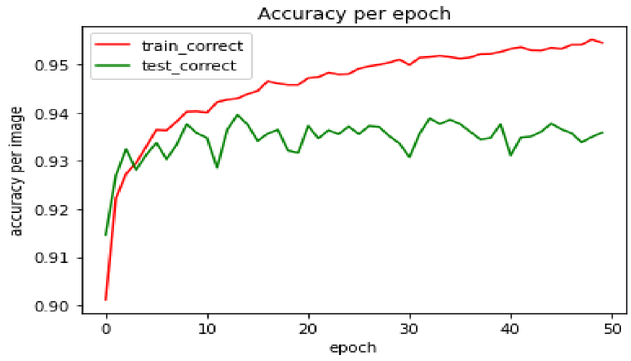
Keaton Spiller

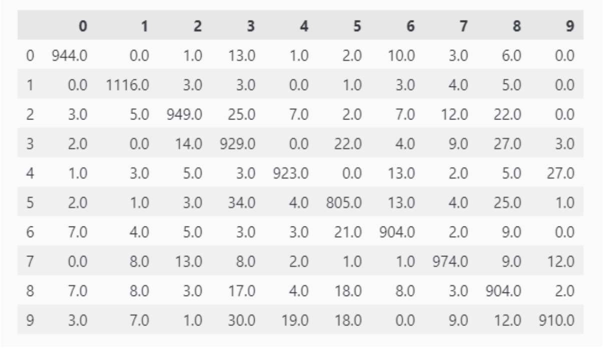
CS 445 Winter 2022

Assignment 1 Report

Experiment 1: Varying the number of hidden units for n = 20, 50, and 100.

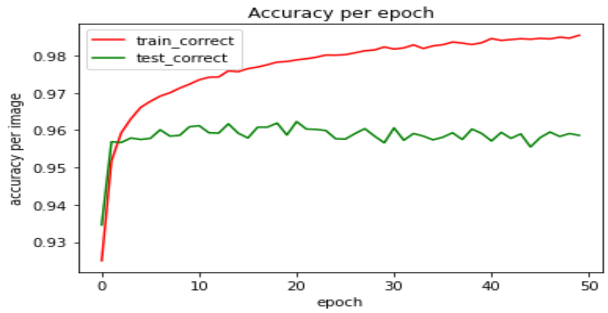
More hidden nodes led to higher levels of accuracy in prediction, from ~93% with n = 20, ~95% in n = 50 and 96% for n = 100. Training converged slightly sooner with larger values of hidden nodes. n = 20 took all 50 epochs, while n = 50 took 30 epochs and n = 100 ~20 epochs. When we use 100 hidden nodes, the networks could be overfitting the data, because the train fit hits 99% and perfect fit's could represent memorizing too much of the training data. The test data prediction has ~95 % accuracy which is pretty good, yet ~5% error in prediction could be due to overfitting the training data. My perceptron from HW 1 hovered around 85% accuracy, and the multilayer perceptron quickly surpassed 95% accuracy with additional hidden nodes without any change in eta values.

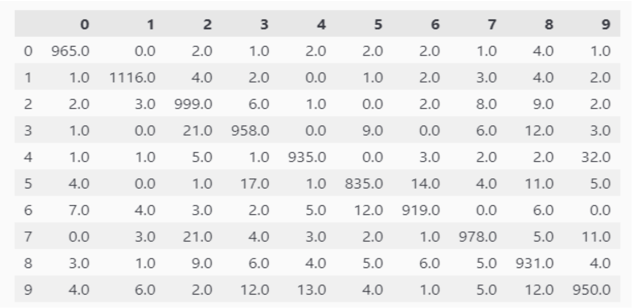




	0	1	2	3	4	5	6	7	8	9
0	944.0	0.0	1.0	13.0	1.0	2.0	10.0	3.0	6.0	0.0
1	0.0	1116.0	3.0	3.0	0.0	1.0	3.0	4.0	5.0	0.0
2	3.0	5.0	949.0	25.0	7.0	2.0	7.0	12.0	22.0	0.0
3	2.0	0.0	14.0	929.0	0.0	22.0	4.0	9.0	27.0	3.0
4	1.0	3.0	5.0	3.0	923.0	0.0	13.0	2.0	5.0	27.0
5	2.0	1.0	3.0	34.0	4.0	805.0	13.0	4.0	25.0	1.0
6	7.0	4.0	5.0	3.0	3.0	21.0	904.0	2.0	9.0	0.0
7	0.0	8.0	13.0	8.0	2.0	1.0	1.0	974.0	9.0	12.0
8	7.0	8.0	3.0	17.0	4.0	18.0	8.0	3.0	904.0	2.0
9	3.0	7.0	1.0	30.0	19.0	18.0	0.0	9.0	12.0	910.0

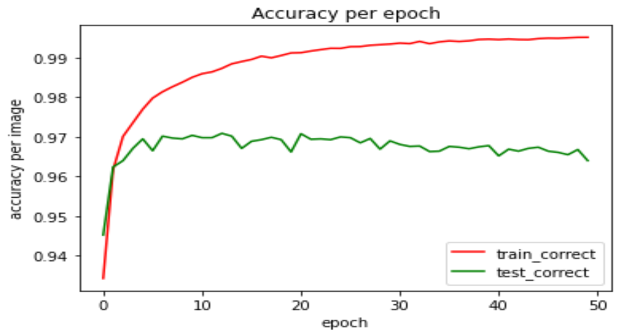
~ 6 minutes for hidden nodes n = 20, eta = 0.1, momentum = 0.9

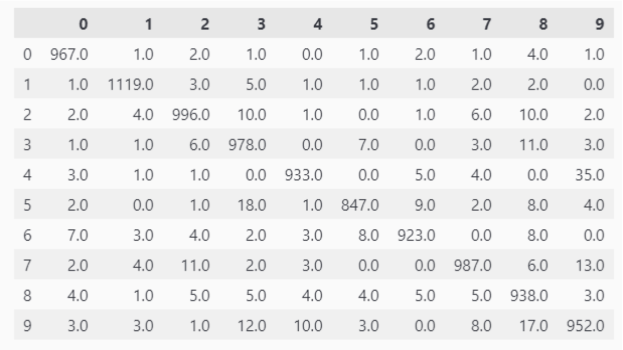




	0	1	2	3	4	5	6	7	8	9
0	965.0	0.0	2.0	1.0	2.0	2.0	2.0	1.0	4.0	1.0
1	1.0	1116.0	4.0	2.0	0.0	1.0	2.0	3.0	4.0	2.0
2	2.0	3.0	999.0	6.0	1.0	0.0	2.0	8.0	9.0	2.0
3	1.0	0.0	21.0	958.0	0.0	9.0	0.0	6.0	12.0	3.0
4	1.0	1.0	5.0	1.0	935.0	0.0	3.0	2.0	2.0	32.0
5	4.0	0.0	1.0	17.0	1.0	835.0	14.0	4.0	11.0	5.0
6	7.0	4.0	3.0	2.0	5.0	12.0	919.0	0.0	6.0	0.0
7	0.0	3.0	21.0	4.0	3.0	2.0	1.0	978.0	5.0	11.0
8	3.0	1.0	9.0	6.0	4.0	5.0	6.0	5.0	931.0	4.0
9	4.0	6.0	2.0	12.0	13.0	4.0	1.0	5.0	12.0	950.0

13 minutes 28.3 seconds for hidden nodes n = 50, eta = 0.1, momentum = 0.9



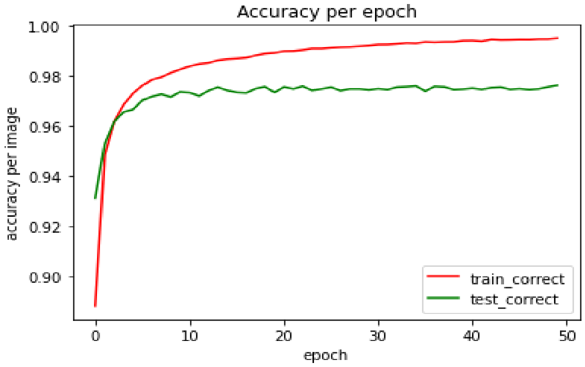


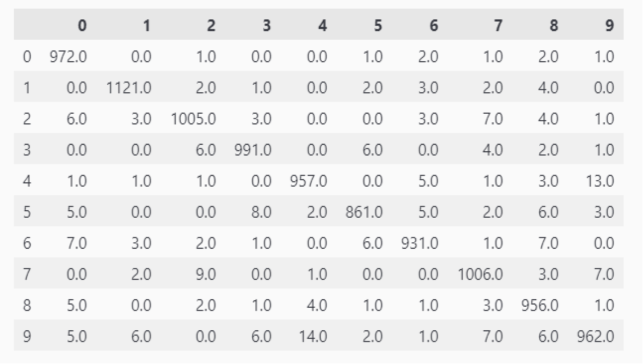
	0	1	2	3	4	5	6	7	8	9
0	967.0	1.0	2.0	1.0	0.0	1.0	2.0	1.0	4.0	1.0
1	1.0	1119.0	3.0	5.0	1.0	1.0	1.0	2.0	2.0	0.0
2	2.0	4.0	996.0	10.0	1.0	0.0	1.0	6.0	10.0	2.0
3	1.0	1.0	6.0	978.0	0.0	7.0	0.0	3.0	11.0	3.0
4	3.0	1.0	1.0	0.0	933.0	0.0	5.0	4.0	0.0	35.0
5	2.0	0.0	1.0	18.0	1.0	847.0	9.0	2.0	8.0	4.0
6	7.0	3.0	4.0	2.0	3.0	8.0	923.0	0.0	8.0	0.0
7	2.0	4.0	11.0	2.0	3.0	0.0	0.0	987.0	6.0	13.0
8	4.0	1.0	5.0	5.0	4.0	4.0	5.0	5.0	938.0	3.0
9	3.0	3.0	1.0	12.0	10.0	3.0	0.0	8.0	17.0	952.0

23 minutes and 8.1 seconds for hidden nodes n = 100, eta = 0.1, momentum = 0.9 (*Longest run time experiment)

Experiment 2: Varying the momentum values with fixed hidden units of 100.

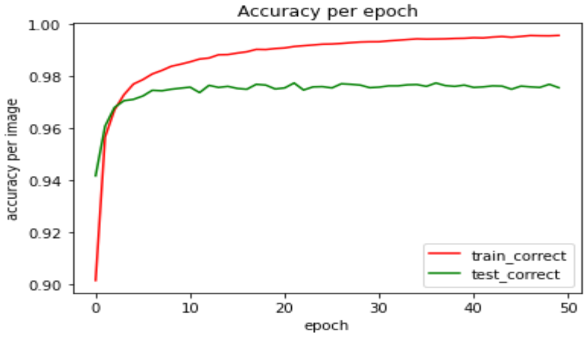
For momentum of 0, 0.25, 0.5, and 0.9 (from Experiment 1) Lower momentum values have a smoother test accuracy, with higher convergence. This could mean that the lower momentum values overfit the data and find lower bounds sooner than networks with larger momentums to push the ball up the hill and find better lower bounds. The higher momentum values give a more jagged test prediction allowing the data to predict with more variation. Training does seem to converge sooner with a smaller number of epochs with smaller momentum's, but this could be due to overfitting. It looks like lower momentum values over fit the data more because the training fit seems too smooth and well tied to the training data.

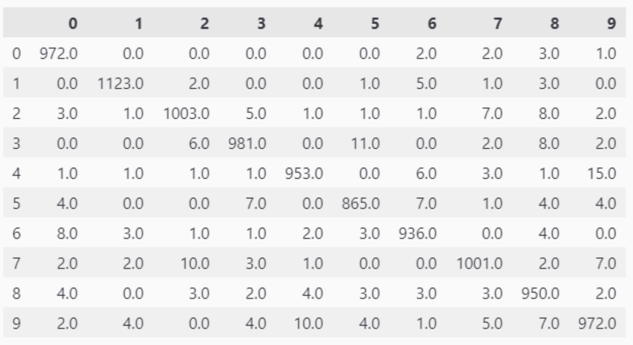




	0	1	2	3	4	5	6	7	8	9
0	972.0	0.0	1.0	0.0	0.0	1.0	2.0	1.0	2.0	1.0
1	0.0	1121.0	2.0	1.0	0.0	2.0	3.0	2.0	4.0	0.0
2	6.0	3.0	1005.0	3.0	0.0	0.0	3.0	7.0	4.0	1.0
3	0.0	0.0	6.0	991.0	0.0	6.0	0.0	4.0	2.0	1.0
4	1.0	1.0	1.0	0.0	957.0	0.0	5.0	1.0	3.0	13.0
5	5.0	0.0	0.0	8.0	2.0	861.0	5.0	2.0	6.0	3.0
6	7.0	3.0	2.0	1.0	0.0	6.0	931.0	1.0	7.0	0.0
7	0.0	2.0	9.0	0.0	1.0	0.0	0.0	1006.0	3.0	7.0
8	5.0	0.0	2.0	1.0	4.0	1.0	1.0	3.0	956.0	1.0
9	5.0	6.0	0.0	6.0	14.0	2.0	1.0	7.0	6.0	962.0

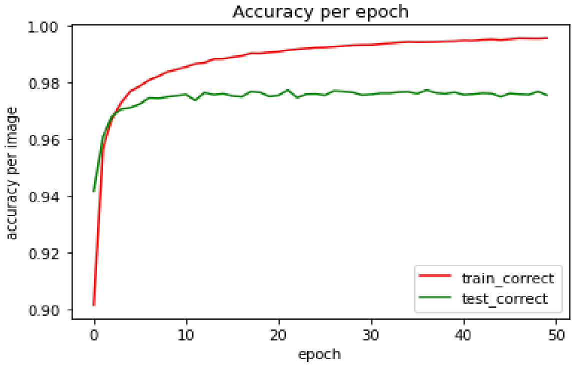
15 minutes 28 seconds, momentum = 0, for hidden nodes n = 100, eta = 0.1

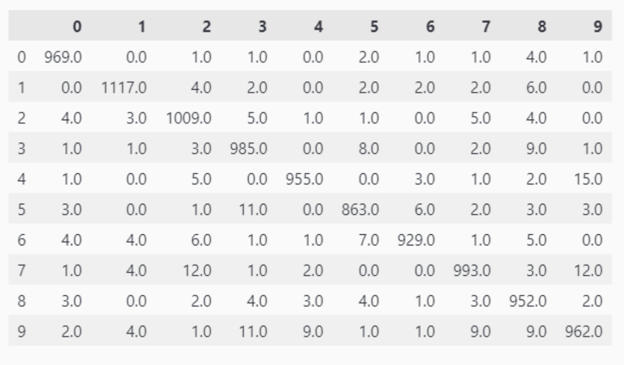




	0	1	2	3	4	5	6	7	8	9
0	972.0	0.0	0.0	0.0	0.0	0.0	2.0	2.0	3.0	1.0
1	0.0	1123.0	2.0	0.0	0.0	1.0	5.0	1.0	3.0	0.0
2	3.0	1.0	1003.0	5.0	1.0	1.0	1.0	7.0	8.0	2.0
3	0.0	0.0	6.0	981.0	0.0	11.0	0.0	2.0	8.0	2.0
4	1.0	1.0	1.0	1.0	953.0	0.0	6.0	3.0	1.0	15.0
5	4.0	0.0	0.0	7.0	0.0	865.0	7.0	1.0	4.0	4.0
6	8.0	3.0	1.0	1.0	2.0	3.0	936.0	0.0	4.0	0.0
7	2.0	2.0	10.0	3.0	1.0	0.0	0.0	1001.0	2.0	7.0
8	4.0	0.0	3.0	2.0	4.0	3.0	3.0	3.0	950.0	2.0
9	2.0	4.0	0.0	4.0	10.0	4.0	1.0	5.0	7.0	972.0

16 minutes, 23.3 seconds for momentum = 0.25, for hidden nodes n = 100, eta = 0.1



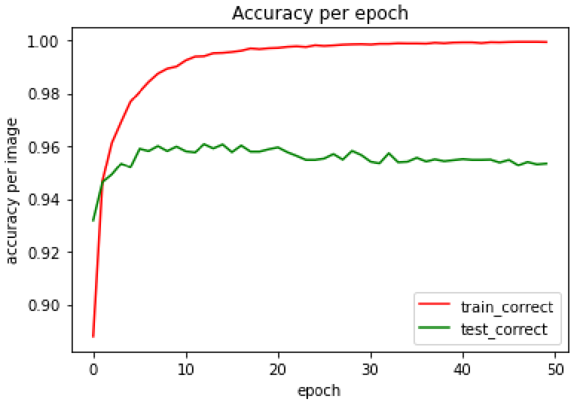


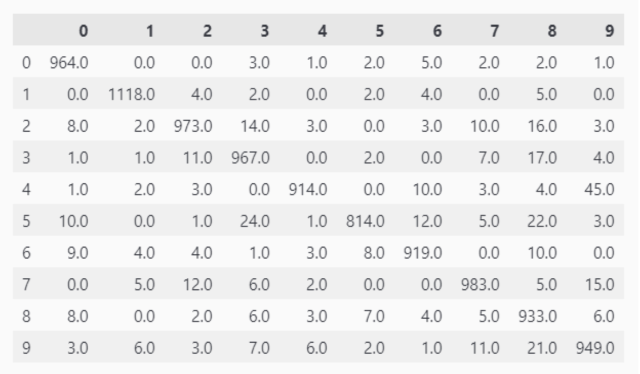
	0	1	2	3	4	5	6	7	8	9
0	969.0	0.0	1.0	1.0	0.0	2.0	1.0	1.0	4.0	1.0
1	0.0	1117.0	4.0	2.0	0.0	2.0	2.0	2.0	6.0	0.0
2	4.0	3.0	1009.0	5.0	1.0	1.0	0.0	5.0	4.0	0.0
3	1.0	1.0	3.0	985.0	0.0	8.0	0.0	2.0	9.0	1.0
4	1.0	0.0	5.0	0.0	955.0	0.0	3.0	1.0	2.0	15.0
5	3.0	0.0	1.0	11.0	0.0	863.0	6.0	2.0	3.0	3.0
6	4.0	4.0	6.0	1.0	1.0	7.0	929.0	1.0	5.0	0.0
7	1.0	4.0	12.0	1.0	2.0	0.0	0.0	993.0	3.0	12.0
8	3.0	0.0	2.0	4.0	3.0	4.0	1.0	3.0	952.0	2.0
9	2.0	4.0	1.0	11.0	9.0	1.0	1.0	9.0	9.0	962.0

16 minutes 30.1 seconds for momentum = 0.5, for hidden nodes n = 100, eta = 0.1

Experiment 3: Testing smaller subsets of Training examples. Half vs quarter subsets with fixed n=100, and momentum = 0.9.

Using hidden units n = 100 and momentum = 0.9 higher training data led to higher test accuracy, with ~ 0.95% accuracy for the quarter training examples, and ~96% for half the training examples. Larger training examples takes longer to converge, the quarter data converged around 15 epochs and the half data converged around 20 epochs Since the training accuracy is so close to 100% we see overfitting. This is because we don't want the neural network to perfectly fit the data and have no room to adjust to variations unseen from training.

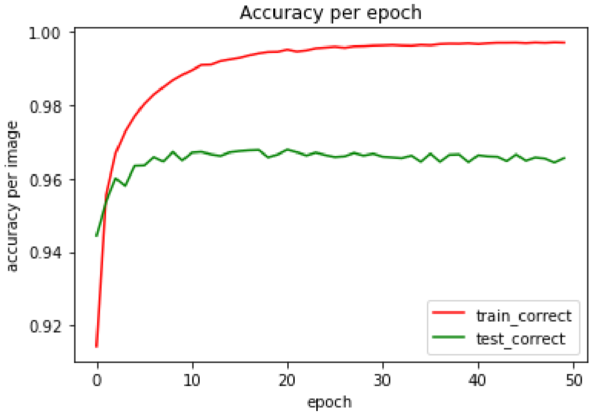


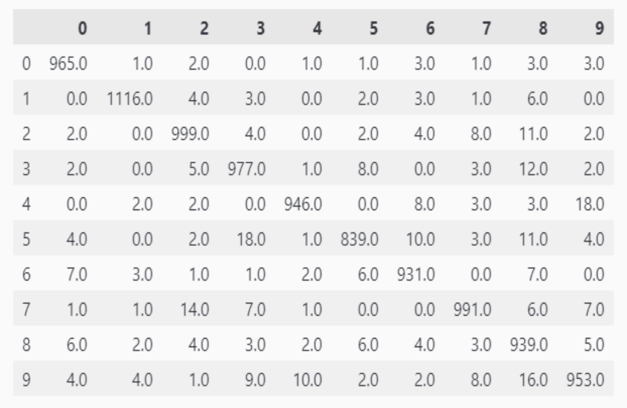


	0	1	2	3	4	5	6	7	8	9
0	964.0	0.0	0.0	3.0	1.0	2.0	5.0	2.0	2.0	1.0
1	0.0	1118.0	4.0	2.0	0.0	2.0	4.0	0.0	5.0	0.0
2	8.0	2.0	973.0	14.0	3.0	0.0	3.0	10.0	16.0	3.0
3	1.0	1.0	11.0	967.0	0.0	2.0	0.0	7.0	17.0	4.0
4	1.0	2.0	3.0	0.0	914.0	0.0	10.0	3.0	4.0	45.0
5	10.0	0.0	1.0	24.0	1.0	814.0	12.0	5.0	22.0	3.0
6	9.0	4.0	4.0	1.0	3.0	8.0	919.0	0.0	10.0	0.0
7	0.0	5.0	12.0	6.0	2.0	0.0	0.0	983.0	5.0	15.0
8	8.0	0.0	2.0	6.0	3.0	7.0	4.0	5.0	933.0	6.0
9	3.0	6.0	3.0	7.0	6.0	2.0	1.0	11.0	21.0	949.0

For quarter training data with 15,000 training pictures for hidden nodes n = 100, eta = 0.1, momentum = 0.9 ,

4 minutes 54.6 seconds





	0	1	2	3	4	5	6	7	8	9
0	965.0	1.0	2.0	0.0	1.0	1.0	3.0	1.0	3.0	3.0
1	0.0	1116.0	4.0	3.0	0.0	2.0	3.0	1.0	6.0	0.0
2	2.0	0.0	999.0	4.0	0.0	2.0	4.0	8.0	11.0	2.0
3	2.0	0.0	5.0	977.0	1.0	8.0	0.0	3.0	12.0	2.0
4	0.0	2.0	2.0	0.0	946.0	0.0	8.0	3.0	3.0	18.0
5	4.0	0.0	2.0	18.0	1.0	839.0	10.0	3.0	11.0	4.0
6	7.0	3.0	1.0	1.0	2.0	6.0	931.0	0.0	7.0	0.0
7	1.0	1.0	14.0	7.0	1.0	0.0	0.0	991.0	6.0	7.0
8	6.0	2.0	4.0	3.0	2.0	6.0	4.0	3.0	939.0	5.0
9	4.0	4.0	1.0	9.0	10.0	2.0	2.0	8.0	16.0	953.0

half training data with 30,000 training pictures for hidden nodes n = 100, eta = 0.1, momentum = 0.9

11 minutes 4.2 seconds