CS 545

Machine Learning

Programming #1

Submitted by Andrew Huffman

February 6, 2020

Experiment 1: Vary number of hidden units.
Do experiments with n = 20, 50, and 100. (Remember to also include a bias unit with weights to every hidden and output node.)

For each value of *n,* train your network on the training set, as described above, changing the weights after each training example. After each epoch, calculate the network's accuracy on the training set **and** the test set for your plot. Train your network for 50 epochs. In your report, give a plot of both training and test accuracy as a function of epoch number (graph both of these in the same plot).

After training is complete, create a confusion matrix for each of your trained networks, summarizing results on the test set.

Discuss your results in a paragraph in your report. Include answers to the following questions:

(1) How does the number of hidden units affect the final accuracy on the test data?
As the number of hidden units increases, the final accuracy on the test data increases, albeit not at a large rate. For example, the difference between the final test accuracy for 50 hidden nodes compared to 100 hidden nodes is less than one percent. That being said, the 50-node and 100-node networks clearly achieve higher test accuracy than the 20-node network

(2) How does it affect the number of epochs needed for training to converge?
The training accuracy curves start to flatten out around the same number of epochs for each case.

(3) Is there evidence that any of your networks has overfit to the training data? If so, what is that evidence?
There is not large evidence of overfitting. The test accuracy is oscillating a lot for the 20-node network so it's more difficult to compare. For the 50-node and 100-node networks, it appears that training curves have an upward trajectory while the test curves have a somewhat flatter trajectory. This would indicate that the network is starting to overfit since it's increasing accuracy on the training data is not generalizing to the test data.

(4) How do your results compare to the results obtained by your perceptron in HW 1?
These results are certainly more accurate than the results from the single perceptron in HW 1. Also, the test accuracy curves are much smoother and not as erratic as in HW 1. The performance of these MLPs is much better.

Confusion Matrix for n = 20

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	946	0	3	1	6	10	5	3	4	2
1	0	1120	1	7	0	1	2	2	2	0
2	4	7	975	18	3	2	6	7	10	0
3	0	0	5	975	3	13	1	3	3	7

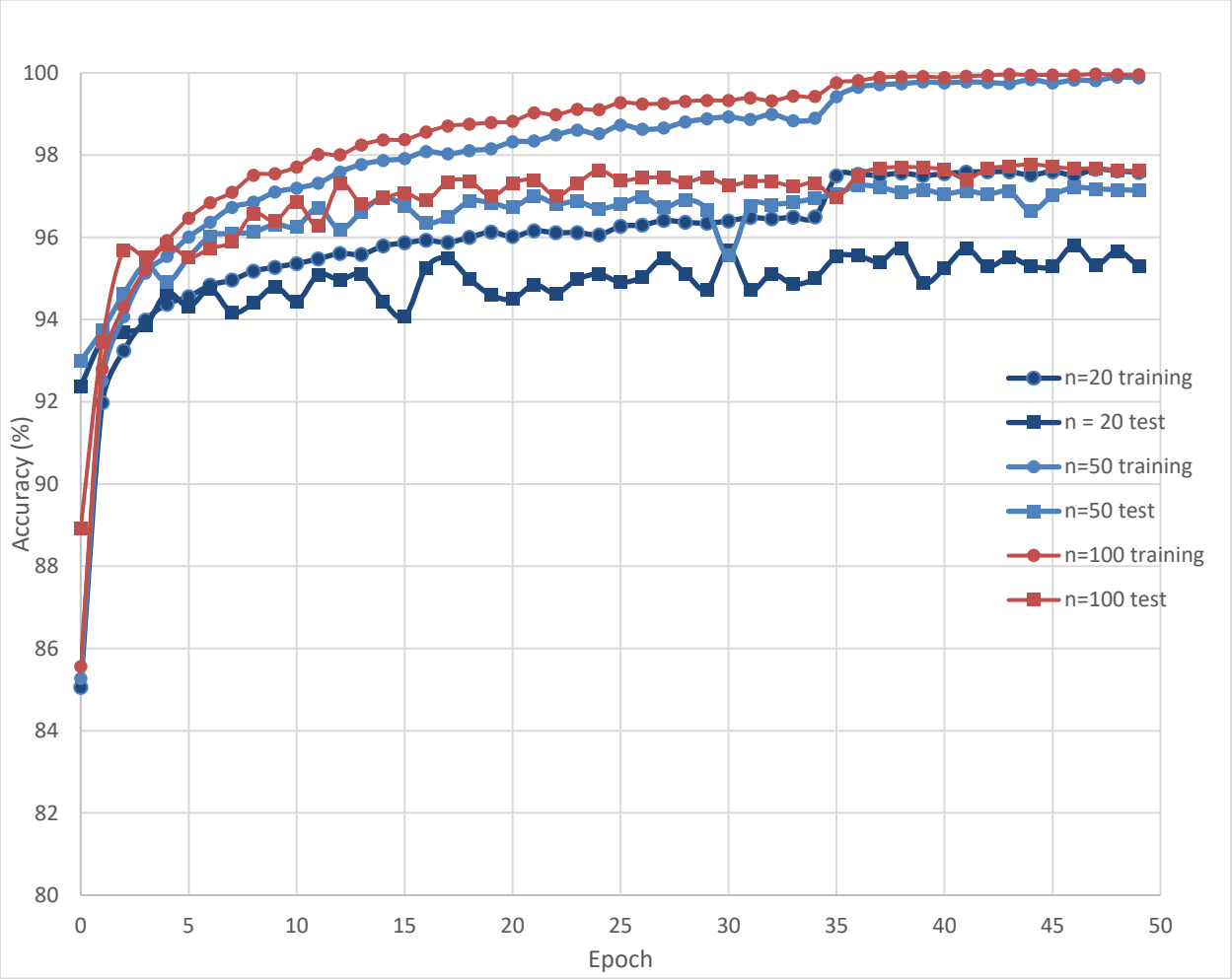
4	1	0	4	2	955	0	11	0	0	9
5	3	2	0	33	3	843	2	0	4	2
6	5	3	1	0	12	19	916	0	2	0
7	1	4	6	10	5	1	0	983	4	14
8	5	1	5	26	9	13	10	5	895	5
9	2	6	1	16	44	8	1	5	5	921

Confusion Matrix for n = 50

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	969	0	0	0	1	2	3	0	2	3
1	0	1123	3	2	0	1	0	0	5	1
2	2	1	1002	5	3	1	3	7	7	1
3	0	0	4	982	1	8	1	3	3	8
4	0	0	5	1	954	0	3	4	1	14
5	2	1	0	15	1	858	5	2	6	2
6	5	3	2	2	9	9	926	0	2	0
7	1	3	9	2	1	1	0	998	0	13
8	7	1	4	12	5	2	1	0	936	6
9	2	4	0	8	12	5	1	7	4	966

Confusion Matrix for n = 100

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	966	1	3	1	0	2	2	1	3	1
1	0	1125	3	1	0	0	1	1	3	1
2	3	4	1008	3	3	0	1	5	4	1
3	0	0	3	992	1	6	0	3	2	3
4	2	0	3	1	958	1	1	1	1	14
5	3	0	0	14	3	865	2	0	3	2
6	5	2	2	1	3	5	938	0	2	0
7	2	3	10	3	1	0	0	993	4	12
8	4	1	3	9	4	4	4	5	937	3
9	2	2	0	7	11	2	1	3	1	980


100
98
96
94
92
90
88
86
84
82
80
Accuracy (%)
0
5
10
15
20
25
30
35
40
45
50
Epoch
n=20 training
n = 20 test
n=50 training
n=50 test
n=100 training
n=100 test

Experiment 2: Vary the momentum value. Here, fix the number of hidden units to 100, and vary the momentum value during training. Use momentum values of 0, 0.25, and 0.5. Train networks using these values as in Experiment 1, and plot the results as in
Experiment 1. (Include your plot for n=100 and momentum = 0.9 that you completed in Experiment 1.)

Create a confusion matrix for each of your trained networks, summarizing results on the test set.
Discuss your results in a paragraph in your report. Include answers to the following questions:

(1) How does the momentum value affect the final accuracy on the test data?
It appears that the momentum value did not significantly affect the final accuracy on the test data as the final test accuracy for each of the curves is between 97%-98%.

(2) How does it affect the number of epochs needed for training to converge?
It appears that the momentum value did not significantly affect the number of epochs needed for training to converge as each of the four training curves flatten at around the same epoch.

(3) Again, is there evidence that any of your networks has overfit to the training data? If so, what is that evidence?
There appears to be some evidence of overfitting as the training curves increasing at the at high number of epochs while the test curves are not increasing so much. This indicates that networks' increasing levels of accuracy on the training data are not generalizing to increasing levels of accuracy on the test data. This discrepancy appears to be more pronounced for higher momentum values.

Confusion Matrix for a = 0

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	968	0	3	0	2	2	2	1	2	0
1	0	1124	3	0	0	0	2	2	4	0
2	3	2	1004	4	4	0	2	7	6	0
3	0	0	5	986	0	3	0	4	4	8
4	1	0	5	1	961	1	2	1	1	9
5	4	1	0	11	3	858	8	1	3	3
6	7	2	5	1	6	8	925	0	3	1
7	1	0	9	0	1	0	0	1007	1	9
8	4	0	5	8	5	4	2	1	942	3
9	1	3	0	8	19	2	0	7	0	969

Confusion Matrix for a = 0.25

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	965	0	1	0	2	1	4	1	4	2
1	0	1121	3	1	0	0	2	2	5	1

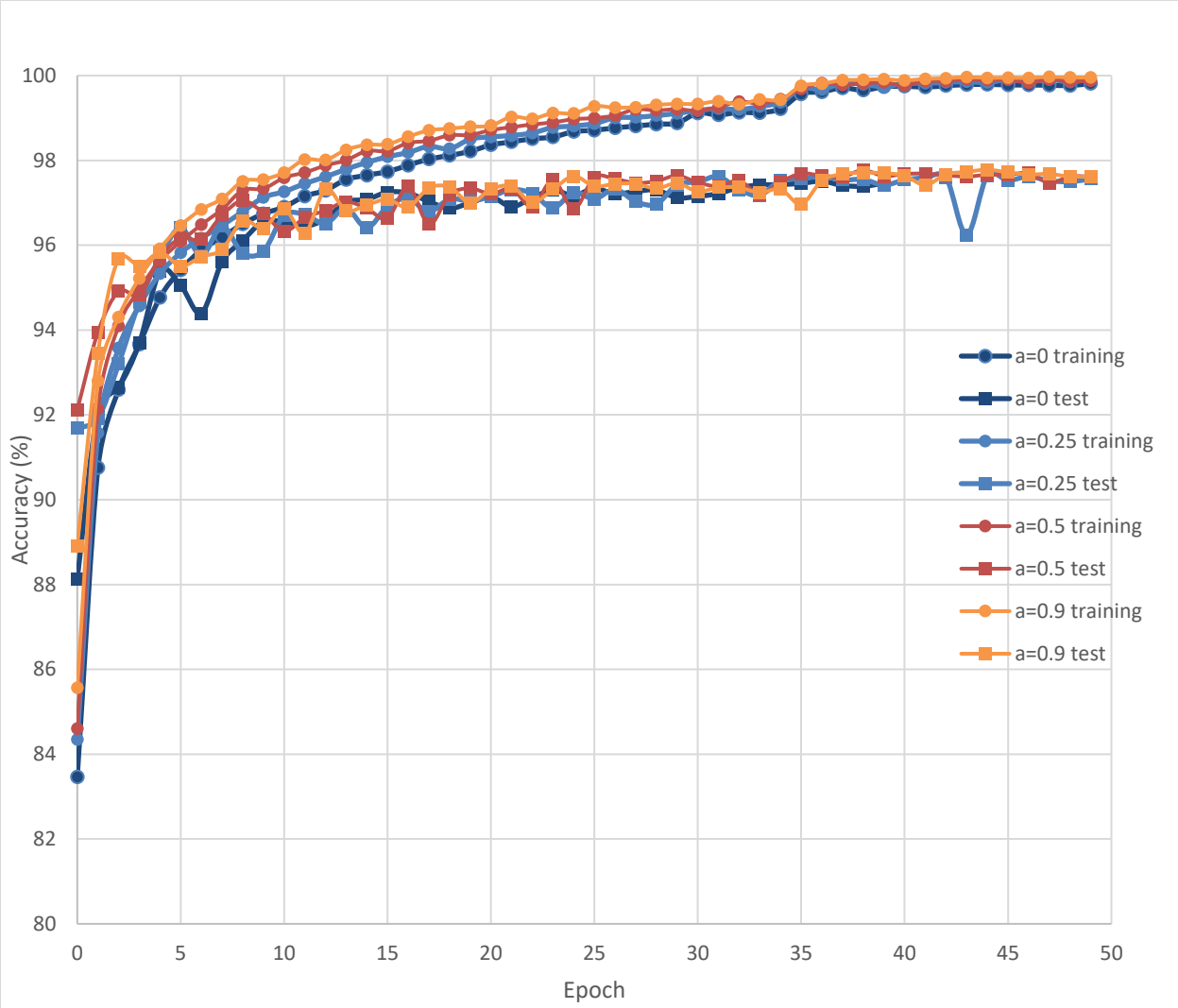
2	4	0	1001	5	3	1	1	6	11	0
3	0	0	4	981	0	10	0	3	8	4
4	2	0	1	1	962	0	4	1	0	11
5	2	0	0	12	1	867	7	1	1	1
6	3	1	1	1	4	5	937	0	6	0
7	1	5	6	6	1	0	0	1000	1	8
8	4	0	2	8	4	6	4	3	941	2
9	1	2	0	8	12	0	0	4	1	981

Confusion Matrix for a = 0.5

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	964	1	2	1	1	4	1	1	2	3
1	0	1125	2	0	0	0	1	1	5	1
2	3	1	1008	2	5	0	1	7	5	0
3	1	0	5	985	0	8	0	3	4	4
4	1	0	4	1	962	2	3	1	0	8
5	3	1	0	16	2	857	8	0	2	3
6	6	3	2	0	6	3	936	0	2	0
7	1	2	8	1	1	0	0	1007	2	6
8	6	1	1	5	4	2	2	2	945	6
9	2	2	0	8	14	4	1	3	2	973

Confusion Matrix for a = 0.9

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	966	1	3	1	0	2	2	1	3	1
1	0	1125	3	1	0	0	1	1	3	1
2	3	4	1008	3	3	0	1	5	4	1
3	0	0	3	992	1	6	0	3	2	3
4	2	0	3	1	958	1	1	1	1	14
5	3	0	0	14	3	865	2	0	3	2
6	5	2	2	1	3	5	938	0	2	0
7	2	3	10	3	1	0	0	993	4	12
8	4	1	3	9	4	4	4	5	937	3
9	2	2	0	7	11	2	1	3	1	980


100
98
96
94
92
90
88
86
84
82
80
Accuracy (%)
0
5
10
15
20
25
30
35
40
45
50
Epoch
a=0 training
a=0 test
a=0.25 training
a=0.25 test
a=0.5 training
a=0.5 test
a=0.9 training
a=0.9 test

Experiment 3: Vary the number of training examples. In this experiment, fix the number of hidden units to 100 and momentum 0.9. Instead of using all of the training examples, train two networks, using respectively one quarter and one half of the training examples for training. Make sure that in each case your training data is approximately balanced among the 10 different classes. Plot the results, as in the previous experiments, plotting accuracy on both the training and test data at the end of each epoch.

Create a confusion matrix for each of your trained networks, summarizing results on the test set.
Discuss your results in a paragraph in your report. Include answers to the following questions:

(1) How does the size of the training data affect the final accuracy on the test data?
Reducing the size of the training data set significantly reduced the final accuracy of the network on the test data. Approximately, the final accuracy on the test data appears to have decreased at the ratio as the decrease in the size of the training set. The final accuracy using 50% of the training data is roughly 50% of the final accuracy using the full data set. Likewise, the final accuracy using 25% of the training data is roughly 25% of the final accuracy achieved using the full data set.

(2) How does it affect the number of epochs needed for training to converge?
It appears that the training accuracy flattens out sooner when using smaller subsets of the full training data set.

(3) Again, is there evidence that any of your networks has overfit to the training data? If so, what is that evidence?
When using 50% or 25% of the training data, there seems to be less evidence of overfitting relative to using the full training data set. For 50% and 25% it is harder to observe a distinction in the trajectory of the training and testing accuracy curves.

Confusion Matrix using 25% of Training Data

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	213	0	0	0	0	4	1	0	1	2
1	0	301	1	2	0	0	2	1	0	0
2	1	1	230	8	2	2	2	3	7	0
3	1	1	2	245	0	5	0	0	1	0
4	1	0	0	0	252	0	1	0	0	2
5	0	1	0	2	1	201	0	0	5	0
6	3	0	0	0	3	1	229	0	0	0
7	0	4	2	2	0	0	0	245	1	3
8	2	1	2	4	1	2	1	1	212	0
9	1	0	0	1	9	1	0	0	1	263

Confusion Matrix using 50% of Training Data

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	443	0	1	0	1	1	1	2	1	1
1	0	583	4	1	0	1	0	0	2	0
2	2	0	476	2	9	1	2	3	5	1
3	0	1	5	486	0	7	0	2	6	4
4	0	0	1	1	472	0	2	0	0	4
5	3	1	0	5	0	441	2	0	5	1
6	3	2	3	0	4	4	481	1	1	0
7	2	3	5	1	0	0	0	500	2	6
8	1	1	4	5	0	0	2	1	452	0
9	1	1	0	3	11	3	0	5	4	496