Thomas Pollard

Programming 1 Report

Introduction:

Before starting I want to clarify a few compromises I made in order to be able to complete this project. I understand that these compromises lower the accuracy of the neural network as well as lowers the integrity of the experiments. The main compromise that I made was to the number of training points used per epoch. My code was so inefficient, that training on all 60,000 data points would cause my linux box to terminate the program before finishing. This meant that I had no experiment results after hours of waiting for it to train. In order to get ANY results, I had to lower my training to a subset of around 3000. In order to make sure I was not training repeatedly on the same small subset; I randomized the data between each epoch to make sure I would have the possibility of seeing any training data point.

Also, part of the purpose of the experiments is to see where the training began to overfit the data, in which the accuracy on the training set kept increasing while the accuracy on the test set went down. I understand that this was unlikely to occur with such a small training subset. This is compounded by the fact that I noticed I was randomizing immediately after training in an epoch, but before testing. This means that each epoch I was testing a different random subset of the training data instead of the one I was just training on. Overall, these two things in combination means that my testing on the training set and test set were **essentially** the same thing, testing a set of data that were not trained on.

Thomas Pollard
Programming 1 Report

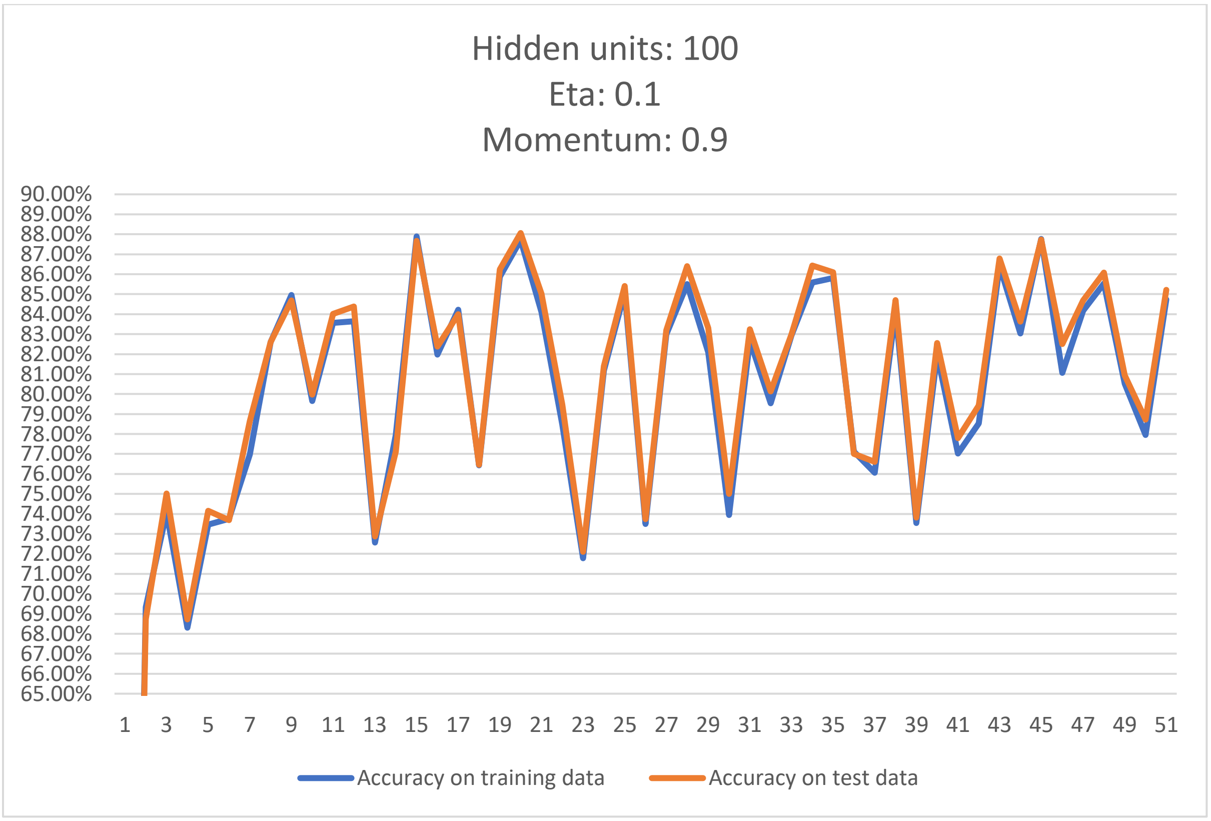
Experiment 1: Vary number of hidden units

Each of these experiments had a momentum of 0.9, an ETA of 0.1, and a training subset of 3000 inputs.

1) 100 hidden units

Confusion Matrix:

	0	1	2	3	4	5	6	7	8	9
0	958	0	0	1	0	17	1	1	0	2
1	0	1081	6	8	3	16	4	3	8	6
2	56	2	826	27	16	31	25	26	16	7
3	13	0	10	812	1	141	1	8	5	19
4	8	0	0	1	858	14	10	4	0	87
5	24	0	0	13	3	834	5	3	0	10
6	72	2	3	0	8	70	795	7	0	1
7	5	7	26	5	7	8	1	909	1	59
8	18	5	9	33	14	281	8	3	525	78
9	18	0	2	3	18	31	3	9	1	924

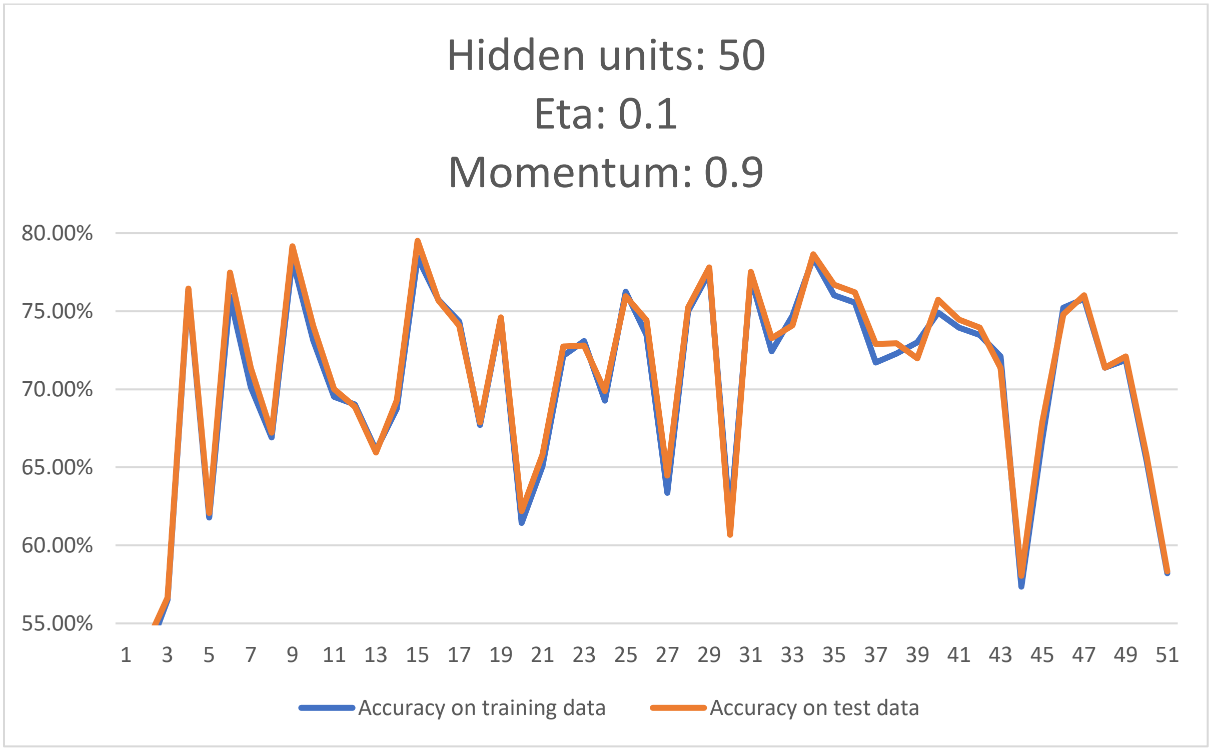


Thomas Pollard
Programming 1 Report

2) 50 hidden units

Confusion Matrix:

	0	1	2	3	4	5	6	7	8	9
0	956	1	0	7	15	1	0	0	0	0
1	0	1102	0	7	25	1	0	0	0	0
2	107	70	436	193	206	0	3	3	14	0
3	49	16	1	890	38	2	0	5	9	0
4	2	0	0	1	978	0	0	0	1	0
5	183	40	2	219	185	246	0	1	16	0
6	158	33	1	2	564	5	189	0	6	0
7	51	46	4	38	206	1	0	680	0	2
8	114	134	0	122	237	7	1	1	358	0
9	30	16	0	11	943	0	0	8	1	0

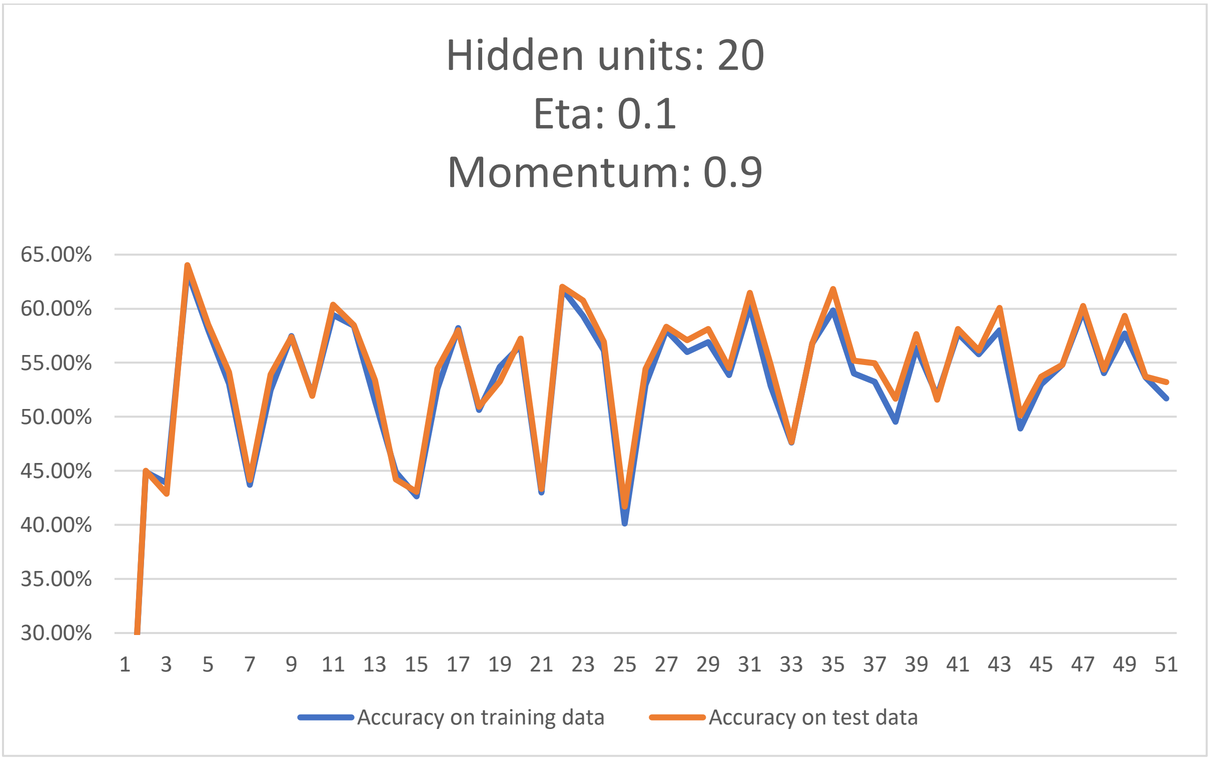


Thomas Pollard
Programming 1 Report

3) 20 hidden units

Confusion Matrix:

	0	1	2	3	4	5	6	7	8	9
0	392	1	56	494	12	20	1	4	0	0
1	0	1015	20	96	4	0	0	0	0	0
2	7	5	865	93	41	1	3	17	0	0
3	0	9	60	904	7	12	1	17	0	0
4	0	1	8	3	953	1	5	11	0	0
5	0	12	61	455	105	239	4	16	0	0
6	4	13	228	53	509	3	146	2	0	0
7	4	9	35	67	131	5	0	772	0	5
8	13	48	156	549	119	45	6	38	0	0
9	3	4	5	43	701	20	0	199	0	34



Thomas Pollard
Programming 1 Report

Experiment 1 summary:

The number of hidden units seems to have a very significant impact on the final accuracy of the test data. My experiments of 20, 50 and 100 hidden units had approximate max accuracies of 63%, 79%, and 88% respectively. However, a larger number of hidden units seems to increase the number of epochs needed to converge. My experiments of 20, 50, and 100 units seemed to converge near 5, 7, and 11 epochs respectively. This may be negatively impacted by the training subset compromise I mentioned in the introduction. There is also a high level of oscillations which I believe is due to the choice of ETA. In my 50 hidden unit run, the accuracy seems to plummet near epoch 42 which could be a sign of overfitting. However, due to my low training set I do not believe that this is overfitting and that it is just a normal oscillation. With 100 hidden units, it achieves better accuracy than the perceptron in my HW 1.

Thomas Pollard
Programming 1 Report

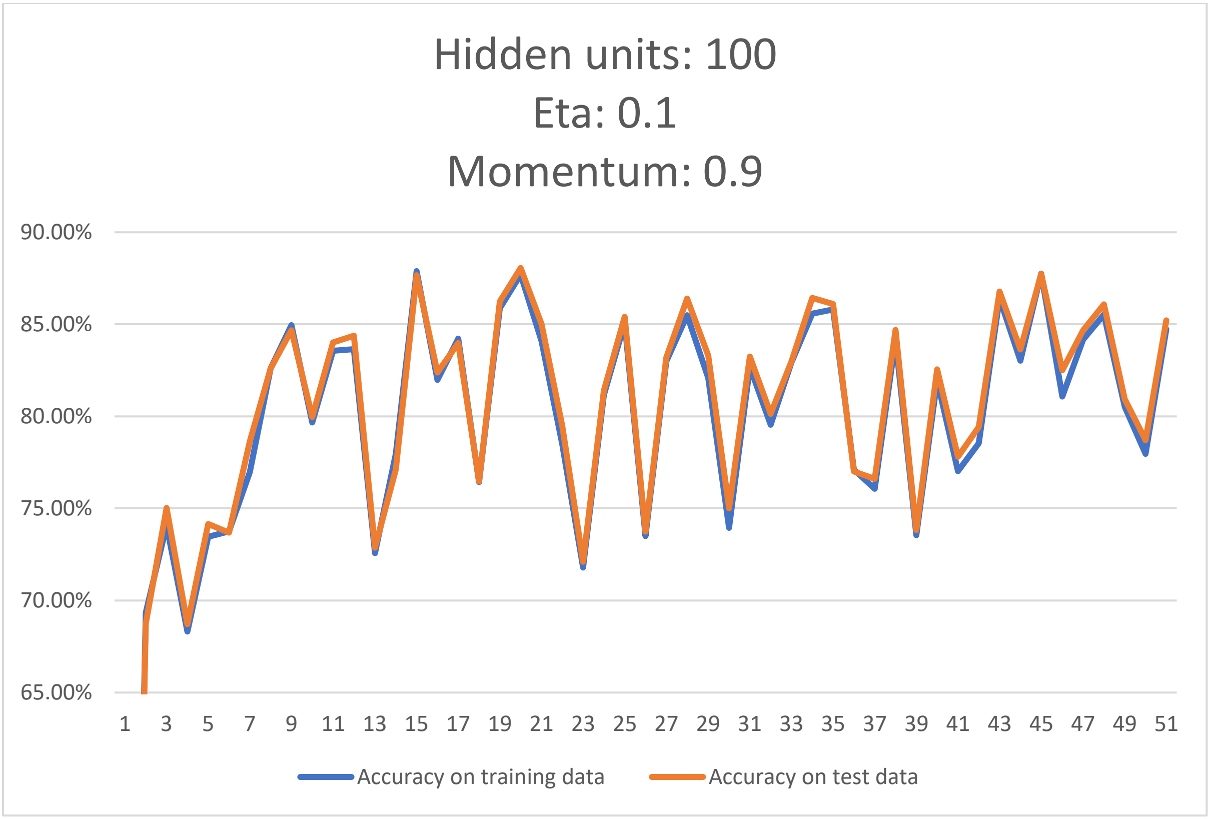
Experiment 2: Vary the momentum value

Each of these experiments had 100 hidden units, an ETA of 0.1, and a training subset of 3000 inputs.

1) Momentum of 0.9

Confusion Matrix:

	0	1	2	3	4	5	6	7	8	9
0	958	0	0	1	0	17	1	1	0	2
1	0	1081	6	8	3	16	4	3	8	6
2	56	2	826	27	16	31	25	26	16	7
3	13	0	10	812	1	141	1	8	5	19
4	8	0	0	1	858	14	10	4	0	87
5	24	0	0	13	3	834	5	3	0	10
6	72	2	3	0	8	70	795	7	0	1
7	5	7	26	5	7	8	1	909	1	59
8	18	5	9	33	14	281	8	3	525	78
9	18	0	2	3	18	31	3	9	1	924

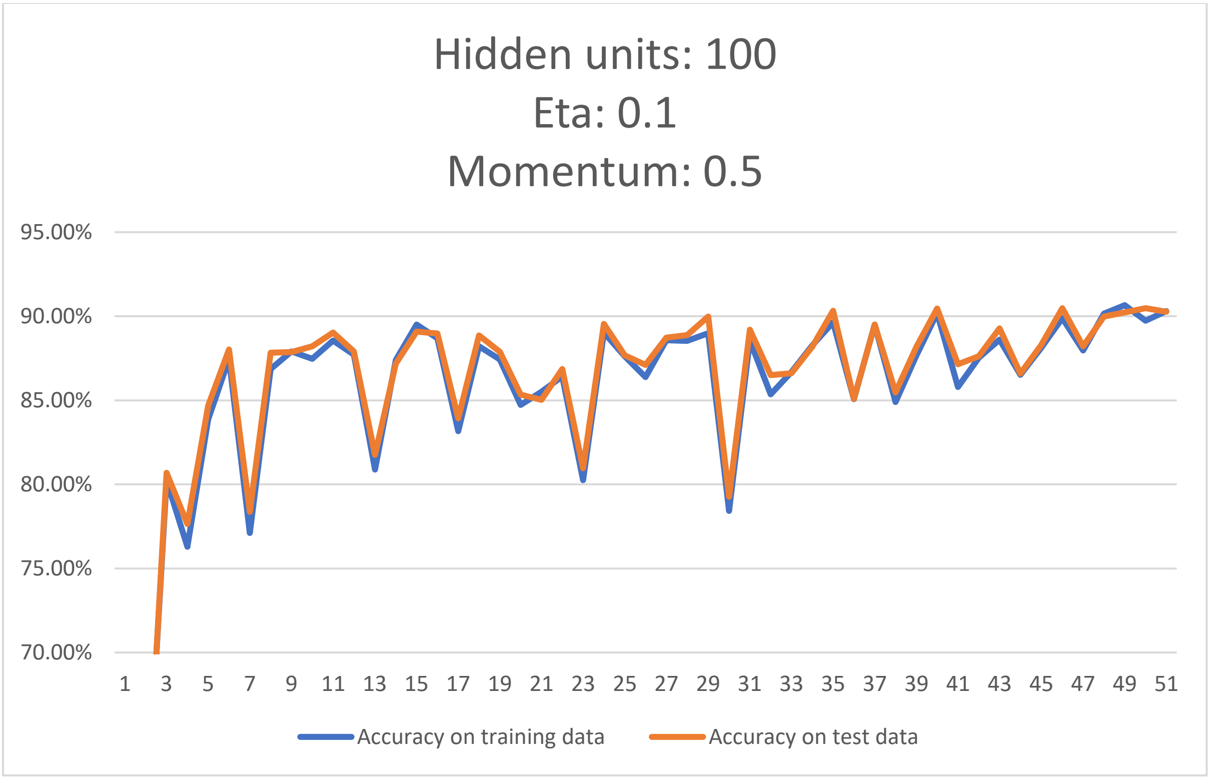


Thomas Pollard
Programming 1 Report

2) Momentum of 0.5

Confusion Matrix:

	0	1	2	3	4	5	6	7	8	9
0	951	0	5	2	3	10	2	5	2	0
1	0	1109	4	1	1	7	2	3	8	0
2	7	2	957	0	11	5	6	22	20	2
3	5	4	55	784	3	91	2	27	34	5
4	2	5	10	0	919	3	7	4	18	14
5	11	5	5	6	11	813	4	9	26	2
6	13	4	17	0	14	30	868	6	6	0
7	1	9	27	1	8	1	0	971	4	6
8	8	7	20	7	8	37	7	13	865	2
9	13	7	7	5	88	15	0	70	14	790

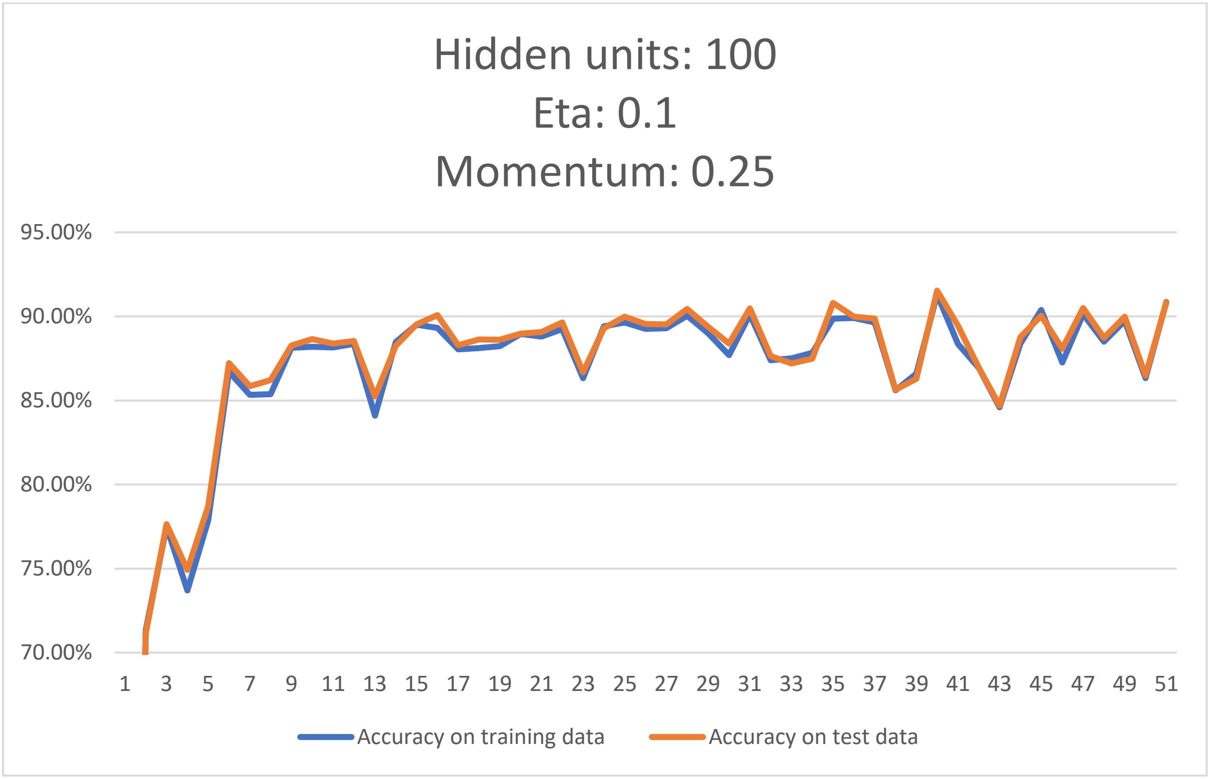


Thomas Pollard
Programming 1 Report

3) Momentum of 0.25

Confusion Matrix:

	0	1	2	3	4	5	6	7	8	9
0	949	0	2	2	0	14	4	6	3	0
1	0	1117	2	3	0	6	3	2	2	0
2	11	6	942	8	8	5	11	11	29	1
3	1	6	26	909	0	31	2	15	20	0
4	2	8	15	2	905	2	9	5	5	29
5	10	5	5	37	4	799	7	6	15	4
6	22	4	16	0	9	30	872	3	2	0
7	3	10	34	5	6	1	0	961	2	6
8	8	21	10	31	6	40	16	7	830	5
9	16	9	2	22	42	10	0	91	19	798

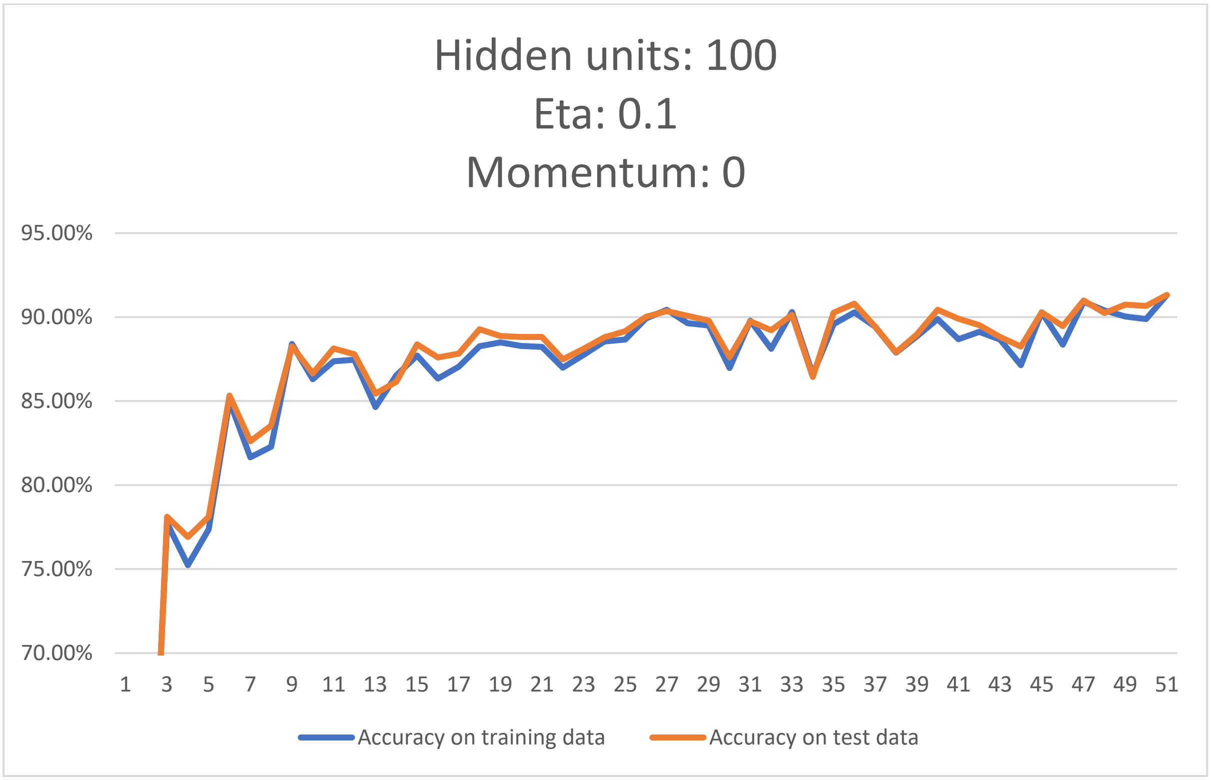


Thomas Pollard
Programming 1 Report

4) Momentum of 0

Confusion Matrix:

	0	1	2	3	4	5	6	7	8	9
0	960	0	4	2	0	5	6	2	1	0
1	0	1123	3	2	0	2	2	2	1	0
2	11	9	951	8	4	3	20	11	14	1
3	3	5	31	925	0	21	2	10	11	2
4	1	9	7	0	900	0	20	2	2	41
5	8	8	6	48	7	773	20	7	7	8
6	14	3	9	1	3	12	914	2	0	0
7	1	17	30	8	6	0	0	933	2	31
8	6	42	10	63	11	30	20	15	769	8
9	13	13	3	16	42	3	0	31	4	884



Experiment 2 summary:

It seems that momentum created more oscillations in my experiments. I think that this is because the weights are likely to continue moving in one direction because of previous change even when it may not be the correct way to go. A lower momentum did seem to correlate with a larger number of epochs needed to converge.

Thomas Pollard
Programming 1 Report

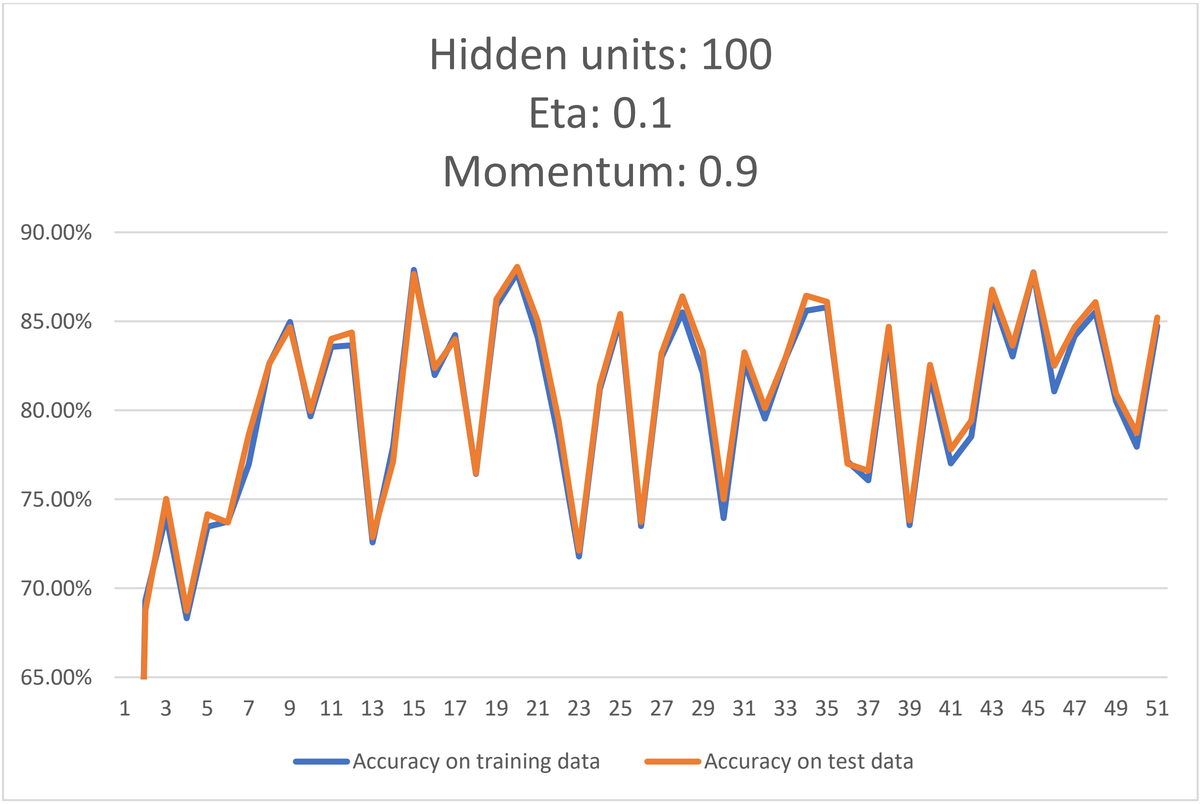
Experiment 3: Vary the number of training examples

Each of these experiments had 100 hidden units, ETA of 0.1, and a momentum of 0.9. Unfortunately because I was already constrained with the number of training examples I could process, this mean I could only go lower.

1) 3000 training examples

Confusion Matrix:

	0	1	2	3	4	5	6	7	8	9
0	958	0	0	1	0	17	1	1	0	2
1	0	1081	6	8	3	16	4	3	8	6
2	56	2	826	27	16	31	25	26	16	7
3	13	0	10	812	1	141	1	8	5	19
4	8	0	0	1	858	14	10	4	0	87
5	24	0	0	13	3	834	5	3	0	10
6	72	2	3	0	8	70	795	7	0	1
7	5	7	26	5	7	8	1	909	1	59
8	18	5	9	33	14	281	8	3	525	78
9	18	0	2	3	18	31	3	9	1	924

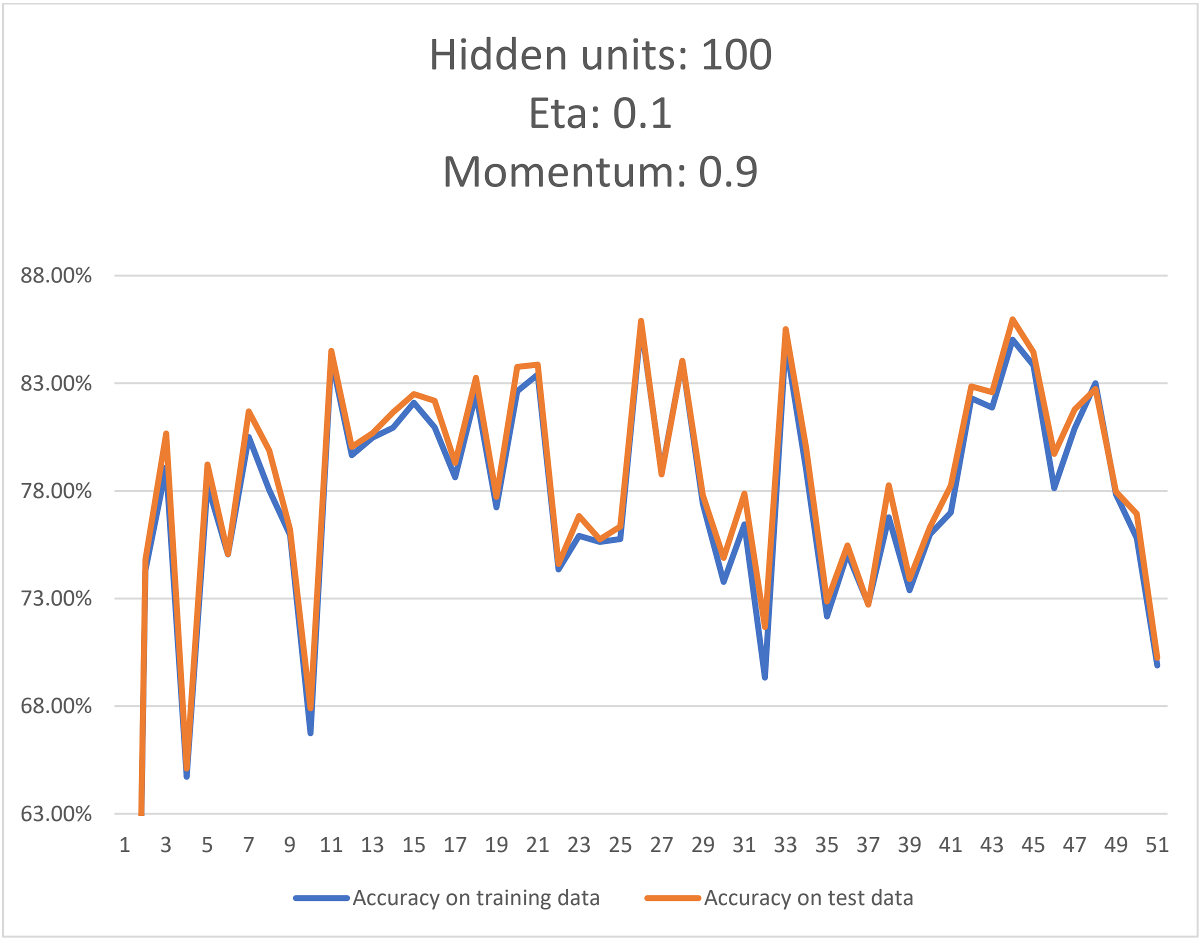


Thomas Pollard
Programming 1 Report

2) 2000 training examples

Confusion Matrix:

	0	1	2	3	4	5	6	7	8	9
0	826	0	118	0	1	23	7	1	4	0
1	0	868	178	0	0	4	4	0	81	0
2	0	0	1001	0	9	0	10	1	10	1
3	1	0	760	6	4	142	2	4	83	8
4	1	1	55	0	890	4	15	0	10	6
5	13	1	122	0	18	686	9	0	41	2
6	10	0	134	0	10	19	785	0	0	0
7	6	2	224	0	51	11	1	662	15	56
8	3	0	174	0	15	11	8	0	760	3
9	4	4	66	0	318	22	0	0	54	541

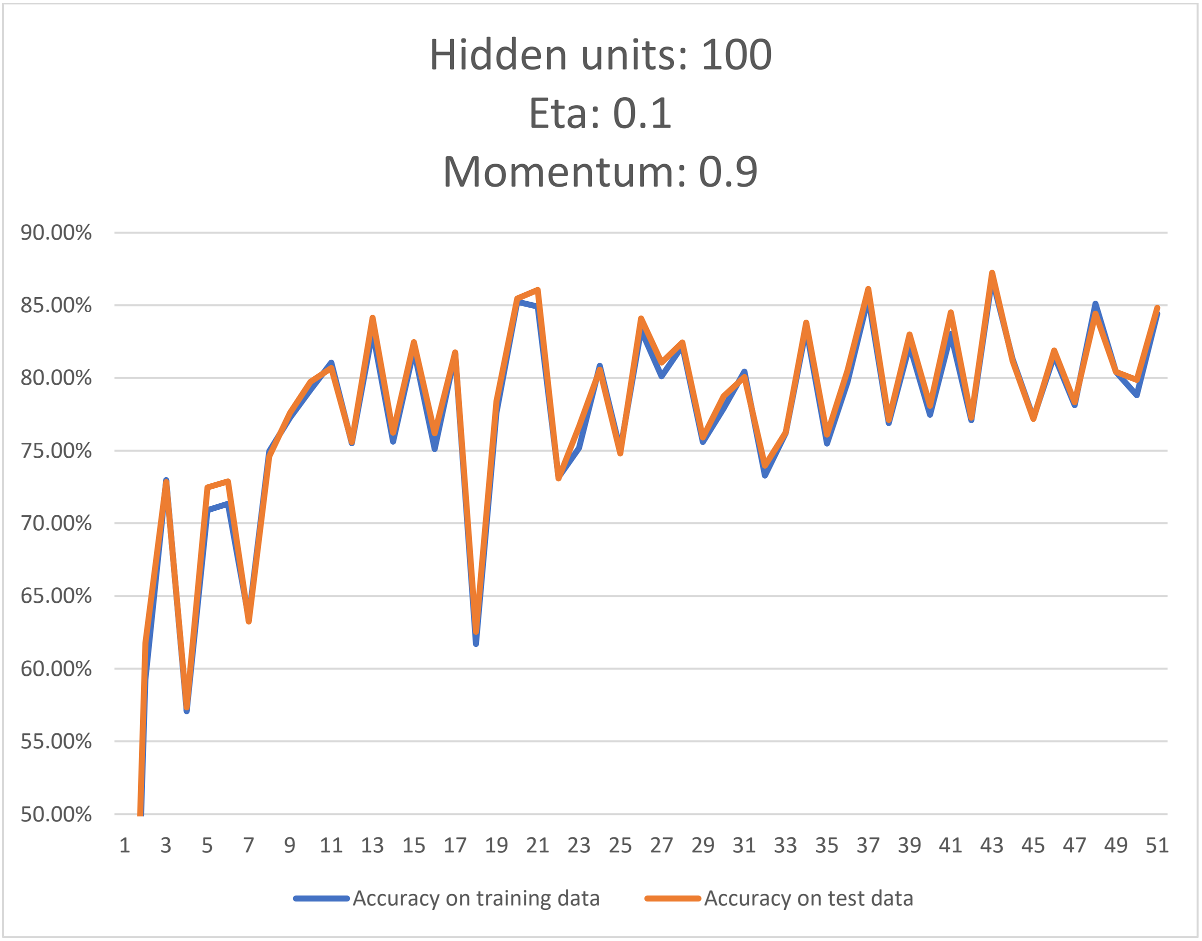


Thomas Pollard
Programming 1 Report

3) 1000 training examples

Confusion Matrix:

	0	1	2	3	4	5	6	7	8	9
0	947	0	0	2	1	7	16	3	4	0
1	0	1085	0	10	1	2	5	1	31	0
2	41	22	753	28	14	2	77	26	69	0
3	13	3	14	895	2	28	3	19	32	1
4	4	11	1	5	825	4	101	5	15	11
5	39	12	1	63	11	662	36	14	53	1
6	24	4	0	1	3	8	915	1	2	0
7	7	21	11	5	12	2	6	946	9	9
8	18	12	3	27	12	34	32	16	819	1
9	20	21	3	26	124	38	37	74	30	636



Thomas Pollard
Programming 1 Report

Experiment 3 summary:

The size of the training data should increase accuracy on the test data up to a certain epoch. My lowest training data size had similar accuracy to my highest data size example. I believe these results are because of the high amount of variance in the possible combinations of training subsets. Overall, I am a little disappointed in the ability and results of my program.