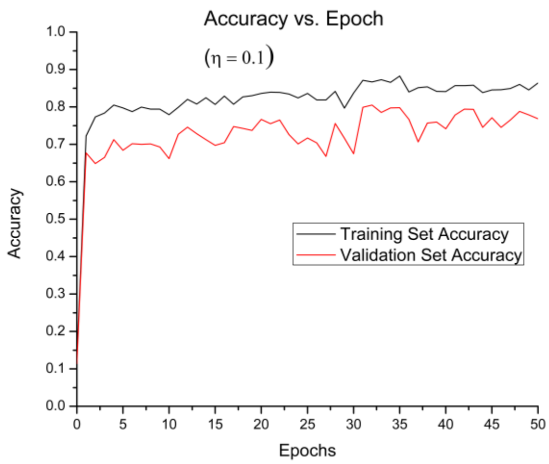
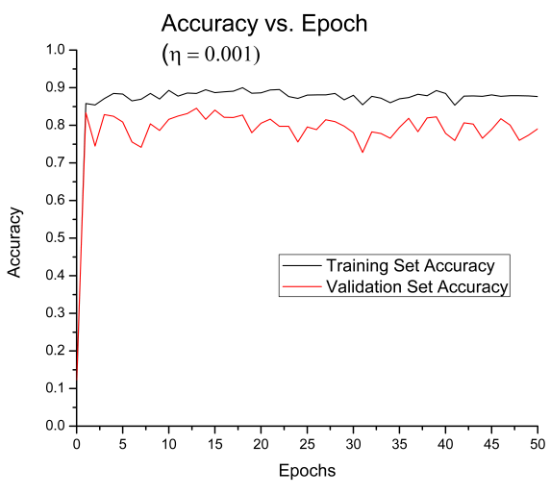
Austen Hsiao ID: 985647212
CS545 Machine Learning
Assignment 1

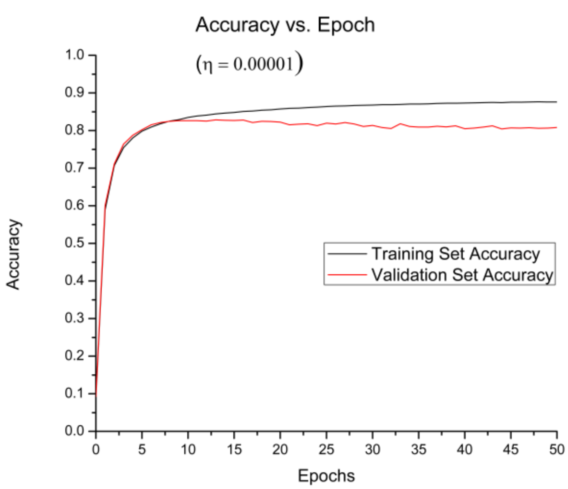
1. Description

The goal of this experiment is to showcase the perceptron learning algorithm by applying it to MNIST digit data with variable learning rates. With 60,000 training data, 10 perceptrons were trained over the course of 50 epochs. It is evident from the results that the perceptron learning algorithm is a facile and effective method for single-layer neural networks. Beginning with randomly assigned weights, this algorithm improved accuracy from ~10% to ~85%.

2. Results







Using learning rate (η) 0.1, 0.001, and 0.00001, the perceptrons were trained.

With $\eta = 0.1$, the accuracy vs. epoch graph shows the highest amount of variability between epochs. The accuracy oscillates between increasing and decreasing which may be due to the larger learning rate. A larger learning rate reduces damping in the calculation of Δw. The accuracy for the training and validation set end at 86% and ~77%, respectively.

With $\eta = 0.001$, the graph shows a reduced amount of variability when compared to $\eta = 0.1$. The "bumpiness" of both learning rates may be attributed to overfitting in which we lose overall trend in favor of experimental data point fidelity. The accuracy for the training and validation sets with a learning rate of 0.001 end at ~88% and ~80%, respectively.

With $\eta = 0.00001$, we instantly notice a smoother profile, which more accurately portrays the general data trend. The accuracy for the training and validation sets with a learning rate of 0.00001 end ~88% and ~81%, respectively. While the ending accuracies for 0.001 and 0.00001 were almost identical, the accuracy for the lower learning rate grew much more steadily. The 0.001 learning rate grew sporadically.

In all three cases, the perceptron learning algorithm logarithmically increased the accuracy to ~85% and slightly increases thereafter in the training data set. The validation data follows the same overall profile as the training set but lags behind in accuracy. In $\eta = 0.1$ and $\eta = 0.001$ the difference between training set and validation set accuracy was mostly consistent after the initial increase for the remaining duration of 50 epochs. This is in contrast to the data for $\eta = 0.00001$ in which the difference between training set and validation set accuracy increased as each epoch passed.

It’s noteworthy to say that the data presented in this report is largely incomplete, since only one trial was conducted at each learning rate and I didn’t bother to quantify the accuracy difference between the two sets.

η = 0.1

Actual Classifier

Experimental Classifier

	0	1	2	3	4	5	6	7	8	9
0	797	3	62	114	35	211	8	108	57	140
1	0	817	4	2	2	5	1	1	4	3
2	6	3	812	5	3	1	1	19	1	1
3	8	2	13	806	3	45	0	13	8	11
4	2	1	6	0	811	8	2	8	4	20
5	0	0	0	0	0	292	2	0	1	1
6	72	11	32	8	34	45	926	1	9	2
7	1	0	6	4	0	7	0	787	2	7
8	94	297	95	66	76	270	18	21	884	74
9	0	1	2	5	18	8	0	70	4	750

η = 0.001

Actual Classifier

Experimental Classifier

	0	1	2	3	4	5	6	7	8	9
0	920	1	66	106	33	193	13	118	59	67
1	0	925	22	8	10	16	1	19	17	24
2	1	2	747	5	2	0	0	7	0	0
3	3	2	12	801	1	43	0	5	3	6
4	0	0	4	1	793	5	2	6	5	8
5	1	0	0	0	0	375	4	0	2	0
6	20	25	45	14	31	49	923	2	11	1
7	0	1	3	2	0	3	0	711	1	3
8	34	179	129	65	75	193	15	34	868	57
9	1	0	4	8	37	15	0	126	8	843

η =0.00001

Actual Classifier

Experimental Classifier

	0	1	2	3	4	5	6	7	8	9
0	952	0	10	8	5	28	16	9	8	13
1	0	947	9	1	2	3	1	5	2	6
2	0	3	762	9	5	3	5	19	4	1
3	2	4	39	872	9	76	1	25	15	14
4	1	0	9	0	798	7	7	7	8	32
5	1	1	0	8	1	387	2	0	2	1
6	10	5	26	9	11	31	890	2	14	0
7	0	2	9	5	1	7	1	811	1	11
8	12	172	161	94	99	335	32	67	918	185
9	2	1	7	4	51	15	3	83	2	746

3. Discussion

Each confusion matrix was generated after 50 epochs of training. The confusion matrix allows us to assess where the network had trouble evaluating digits. Across learning rates, the network had the most trouble deciphering the digit, ‘5’, confusing it with the digit, ‘8’. However, there are no patterns of difficulty across learning rates, that is, the ability to correctly predict, ‘5’, increased when the learning rate was tuned from 0.1 to 0.001 but decreased when the learning rate was changed from 0.001 to 0.00001. Again, there very well may be patterns of difficulty in determining the correct digit but I just didn’t run a sufficient number of trials (1 trial surely isn’t). The network correctly calculated, ‘0’, at a higher rate than the other digits. The most frequent incorrect output was, ‘8’. This may be rationalized as ‘8’ being more similar to the other digits than any other digit.

There are definite differences when changing the learning rate. The profile of the accuracy vs. epoch graph is much smoother at the lowest learning rate (0.00001). The final accuracies for the training- and validation-set improved between a learning rate of 0.1 and 0.001. The accuracies stagnated between 0.001 and 0.00001 but the growth in accuracy in the lower learning rate was more consistent, that is, the change in accuracy between epochs exhibited less variance. This suggests that the accuracies for learning rates of 0.1 and 0.001 are less reliable than 0.00001.