James Bao (PSU ID: 934097519)

Machine Learning Assign #1

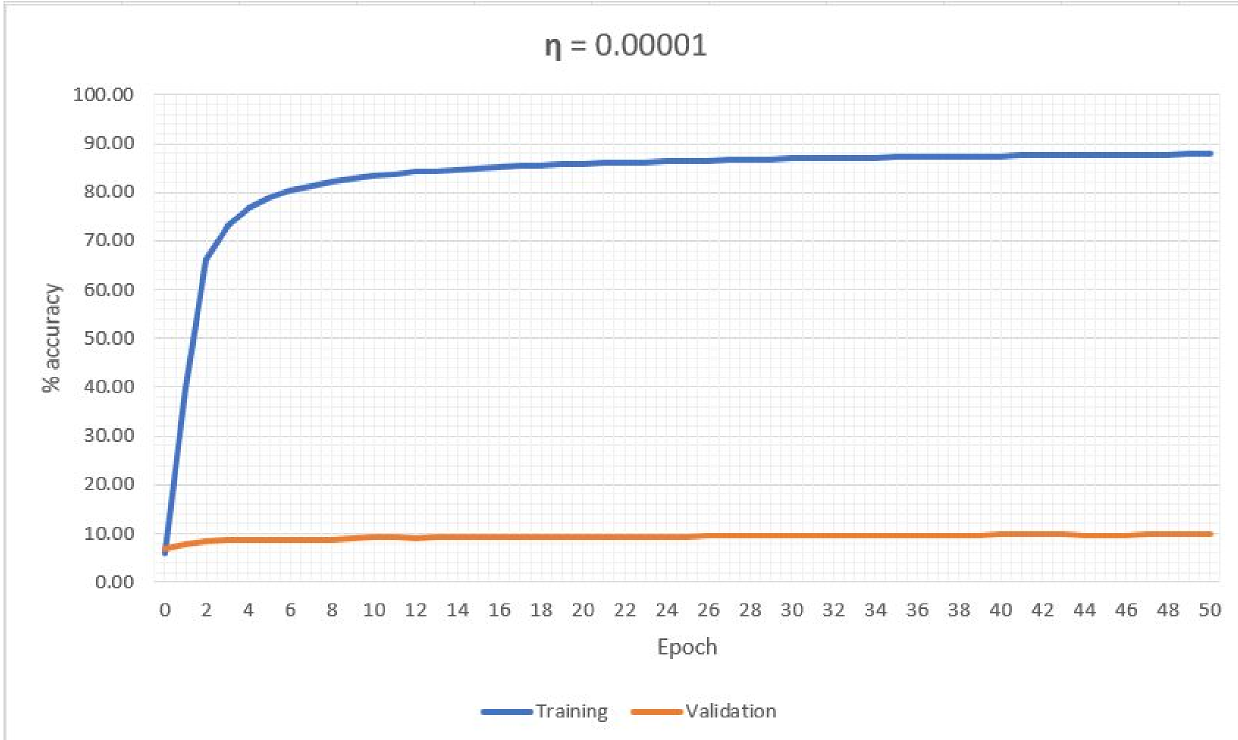
Using a 10-perceptron single layer network to classify handwritten digits

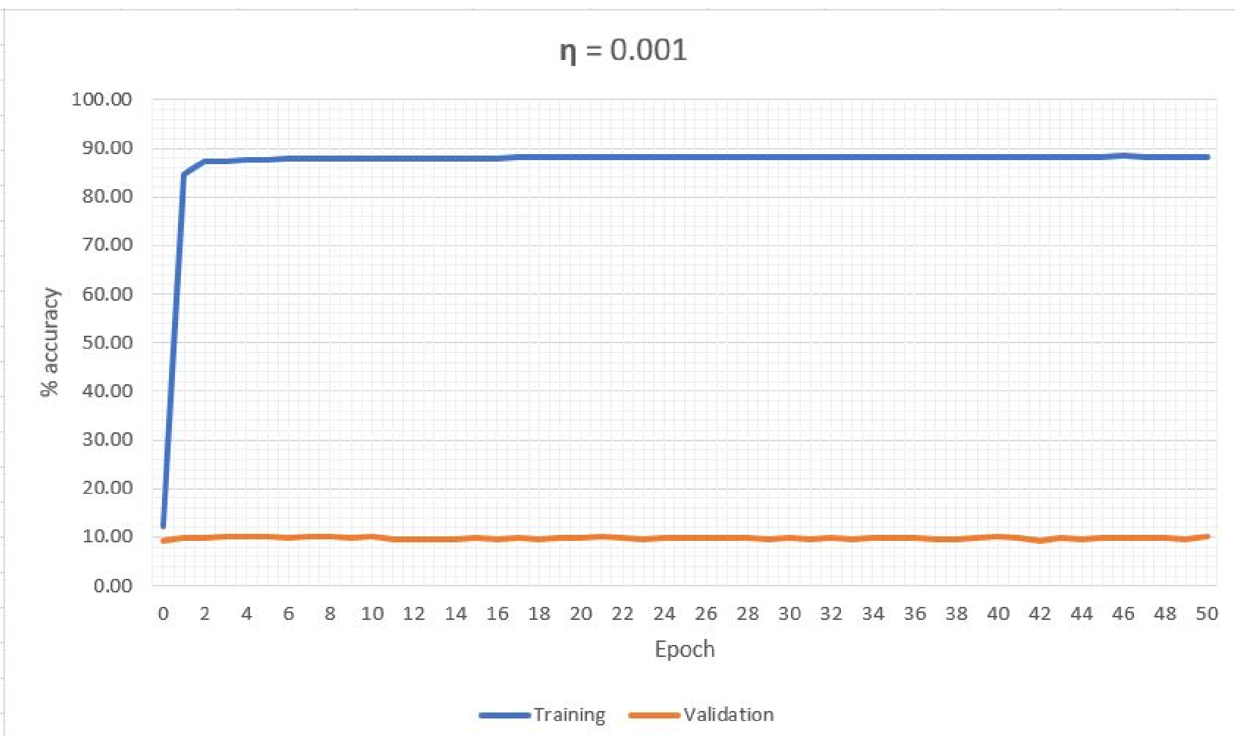
Introduction

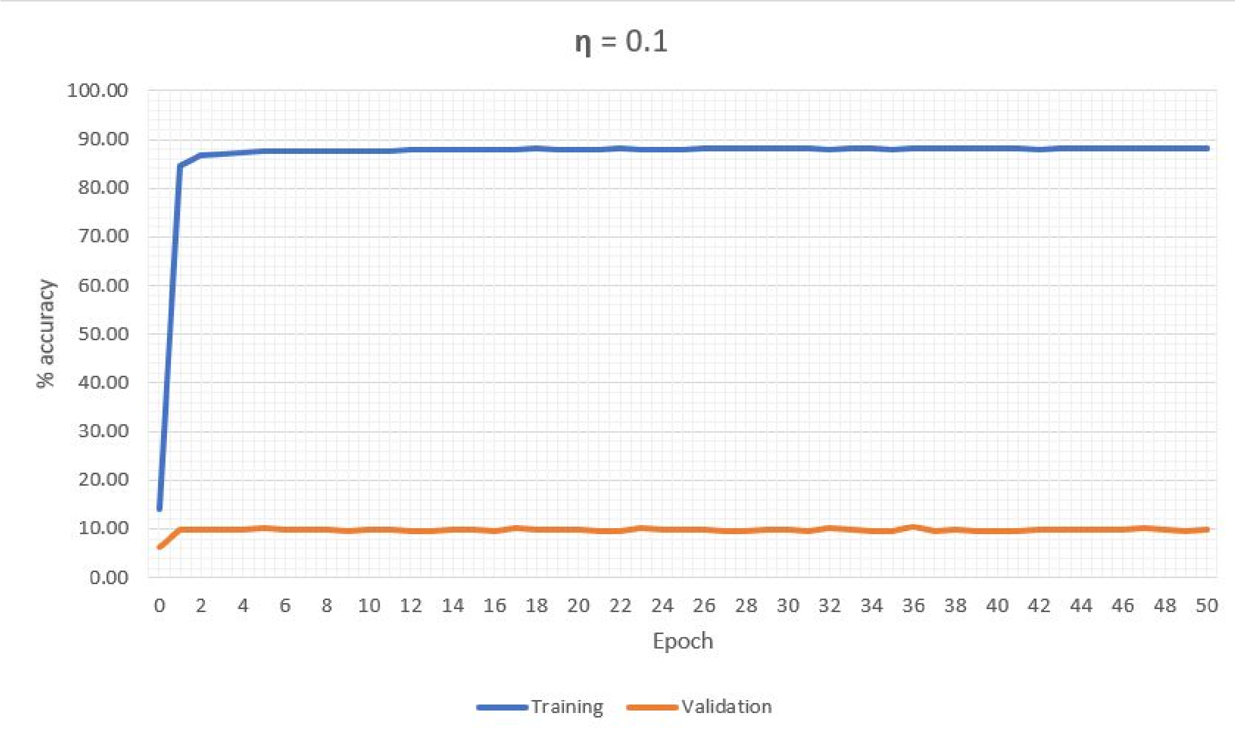
The purpose of this exercise was to design and implement a single-layer perceptron network of 10 perceptrons to classify handwritten digits pixel data stored in .csv format. The datasets used were a 60,000-sample training set and a 10,000-sample validation set; however smaller datasets were also used to test and debug the perceptron code. The perceptrons were trained at 3 different learning rates- 0.00001, 0.001, and 0.1. Weights were initialized at values between -0.05 and +0.05, and the training periods occurred over 50 epochs.

Accuracy plots

Each accuracy plot shows 2 curves, representing the training and validation datasets tested over 50 epochs. Due to time constraints and possible design flaws, there were unavoidable errors when running the 10k-sample validation sets after training on the 60k-sample sets. This may have been caused by using different sets of perceptrons for the training and validation sets (see code).


η = 0.00001
% accuracy
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50
Epoch
Training
Validation


η = 0.001
% accuracy
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50
Epoch
Training
Validation



Confusion Matrices (from 10k sample validation dataset)

η = 0.00001 (actual = rows, expected = columns)

	0	1	2	3	4	5	6	7	8	9
0	88	115	124	116	79	107	108	108	97	115
1	125	123	114	117	86	82	116	122	105	118
2	89	110	85	115	86	88	72	109	89	89
3	116	100	111	93	128	77	91	99	115	110
4	99	106	104	90	81	97	95	69	103	94
5	76	121	77	76	77	88	95	93	72	91
6	101	115	86	79	130	105	101	71	83	118
7	93	130	98	121	112	86	103	113	112	79
8	99	97	102	92	95	81	87	120	102	87
9	94	118	131	111	108	81	90	124	96	108

η= 0.001 (actual = rows, expected = columns)

	0	1	2	3	4	5	6	7	8	9
0	104	104	104	108	77	97	95	115	91	99
1	110	139	124	116	122	120	111	119	121	136
2	85	115	76	117	96	77	76	107	83	98
3	97	95	98	105	110	91	88	90	108	99
4	102	106	122	92	89	84	95	90	91	114
5	77	125	91	70	86	73	103	105	80	97
6	101	120	94	100	91	90	104	89	99	96
7	97	106	101	131	106	88	106	100	95	95
8	104	97	101	89	97	79	92	107	116	81
9	103	128	121	82	108	93	88	106	90	94

η = 0.1 (actual = rows, expected = columns)

	0	1	2	3	4	5	6	7	8	9
0	95	113	117	99	91	95	99	107	96	122
1	107	130	133	120	114	119	97	112	99	126
2	86	109	83	110	92	84	65	112	83	99
3	112	95	93	104	102	81	94	108	123	115
4	116	119	100	93	84	75	97	76	86	102
5	72	132	98	78	92	85	98	102	87	85
6	105	123	88	91	90	82	103	92	81	101
7	88	99	90	117	103	103	105	97	107	94
8	97	95	103	96	96	76	99	109	106	82

9	102	120	127	102	118	92	101	113	106	83

Discussion of Results

While not clearly visible in the graphs, there were mild oscillations for the training dataset at all 3 learning rates. These occurred after the accuracy rate reached the upper 80s%, with larger oscillation magnitude and duration at the 0.1 learning rate, compared to the 0.001 learning rate. (The 0.00001 learning rate exhibited minimal oscillation, possibly due to continuous gradual increases in the accuracy over time). The validation datasets also showed signs of oscillations after 10 epochs, although the accuracy rates plateaued at ~10% instead of 85-90%.

Due to the errors in replicating the high accuracy rates of the training dataset in the validation dataset, it is unreasonable to expect the confusion matrix to show accurate classification of digits. We hope future investigation and tweaking of this perceptron network will yield more accurate classification results.