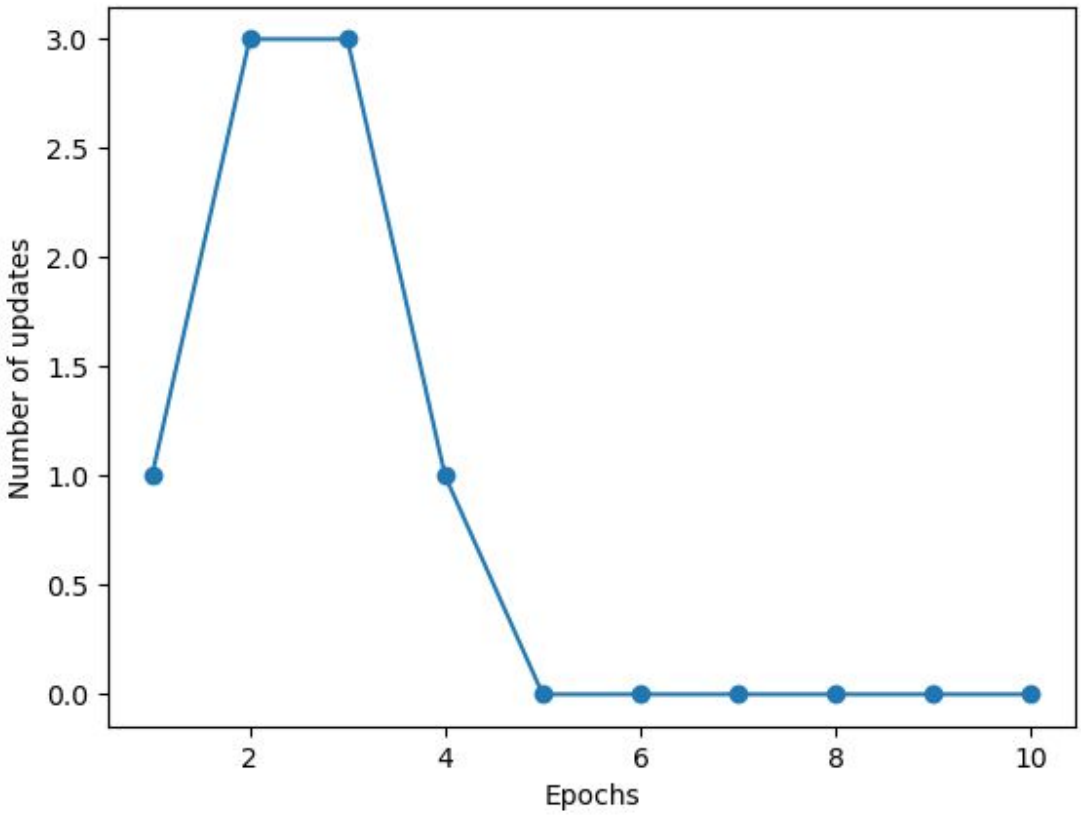
Collin Gros
02-20-2021
HW2
CS-487

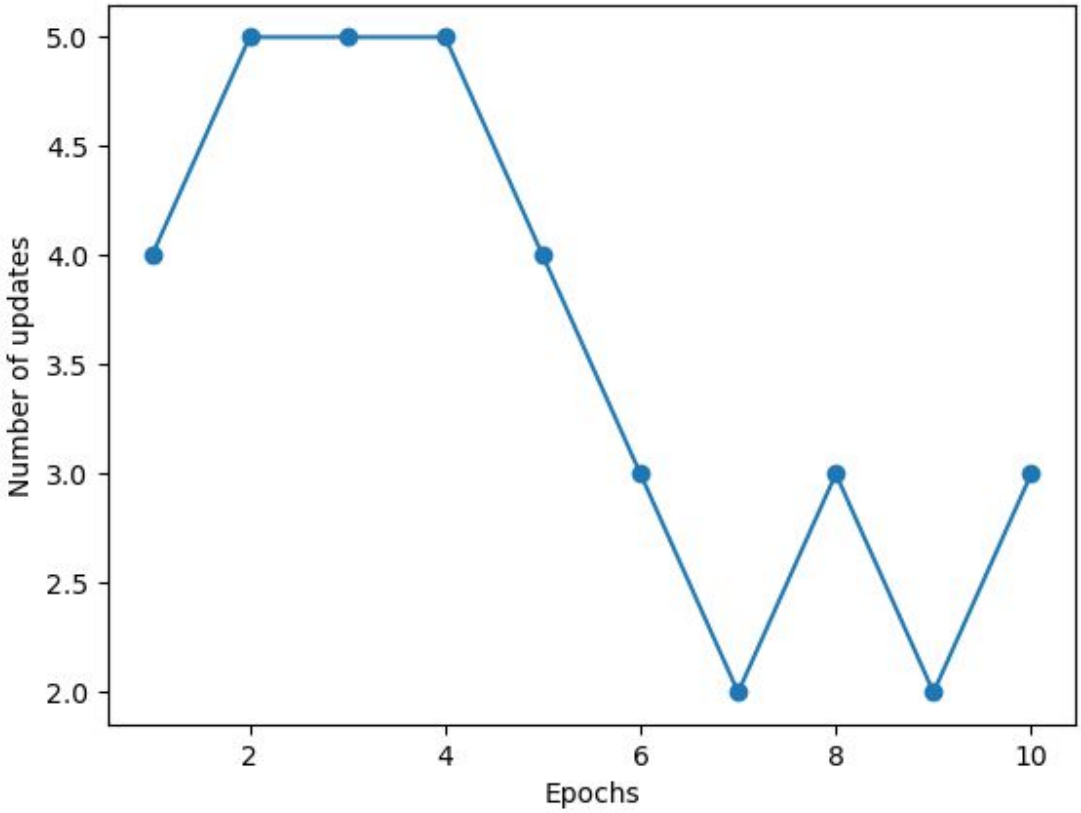
HW2

PERCEPTRON

When trained with eta=0.01, and 10 iterations, perceptron was 100% accurate. Here is a graph of errors for all 10 iterations:



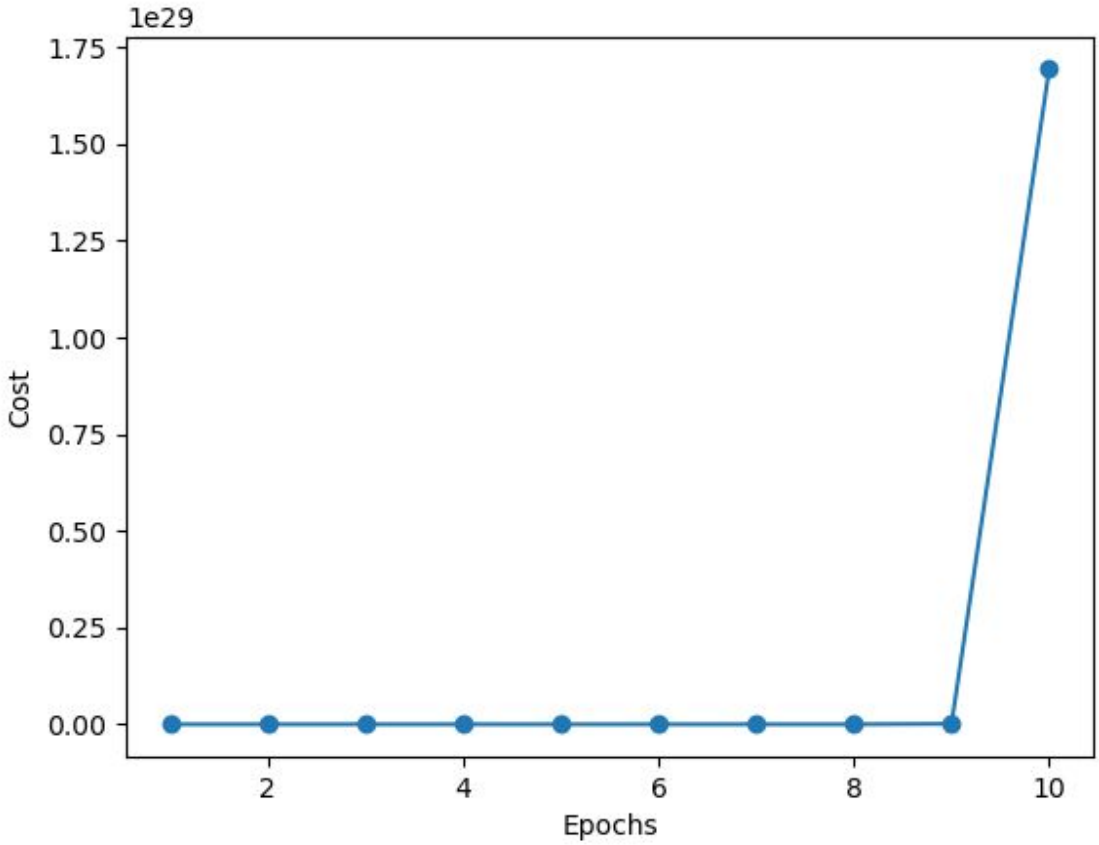
This graph shows that perceptron does a pretty effective job of classifying the two different labels of data. By iteration 5, there were no errors in any iteration - meaning the weights that were calculated worked for every value. With a lower learning rate, (eta=0.0001), perceptron encountered many more errors than before:



I'm not sure why this is the case, perhaps it couldn't get enough data to make an accurate model?

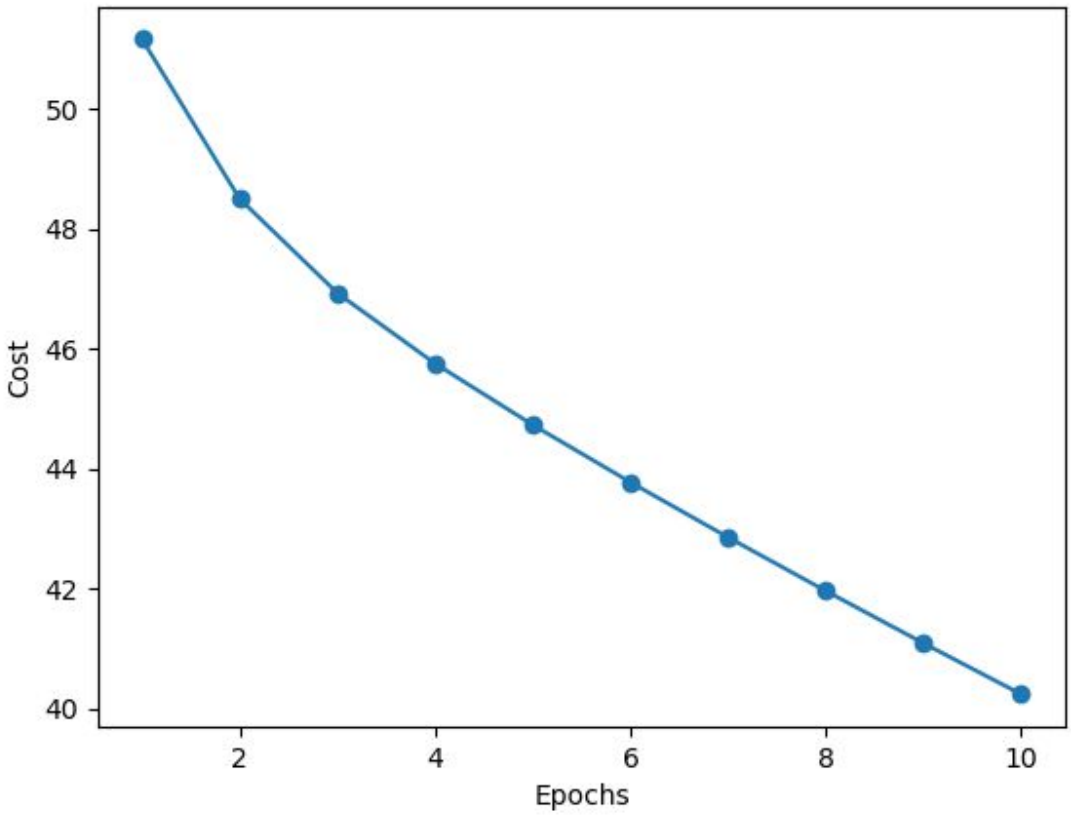
ADALINE

When trained with eta=0.01, and 10 iterations, adaline was 0% accurate. Here is a graph of cost for all 10 iterations:



I have been unable to figure out why cost is at such a high number, and I assume it's probably due to a bug in my code somewhere in the **fit** function. And with a lower learning rate

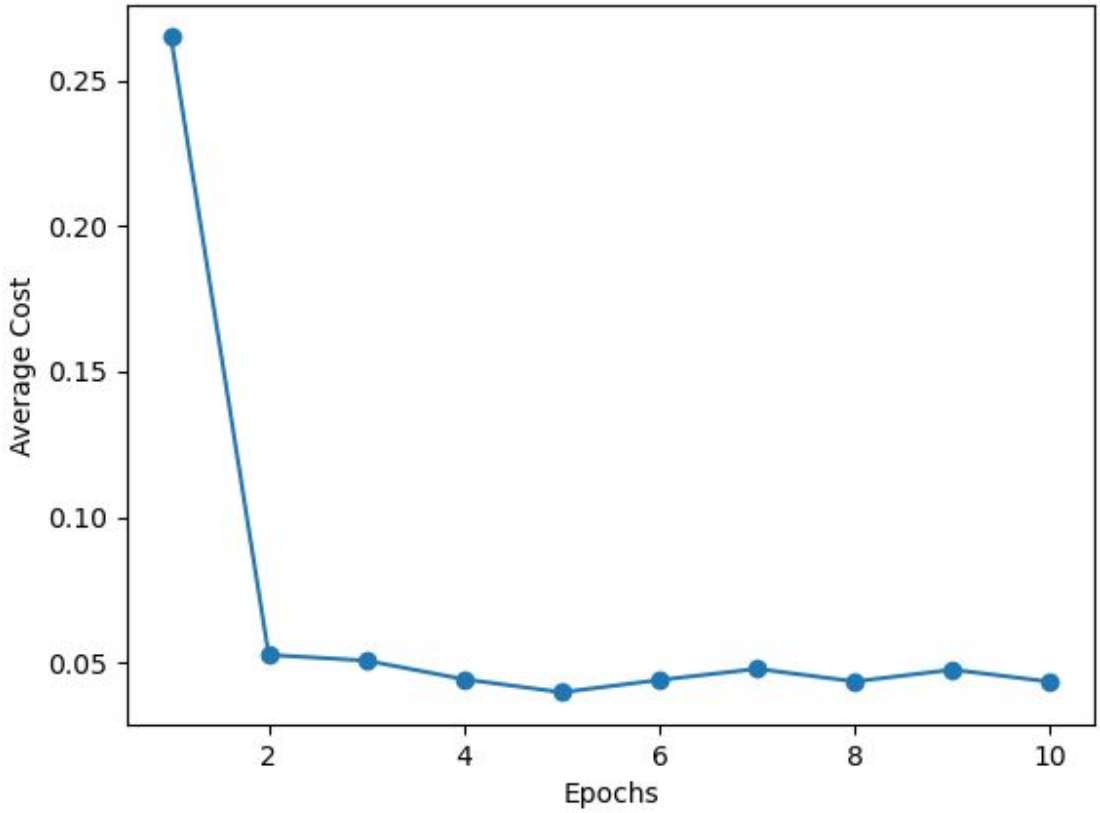
(eta=0.0001), Adaline seems to do significantly better than before:



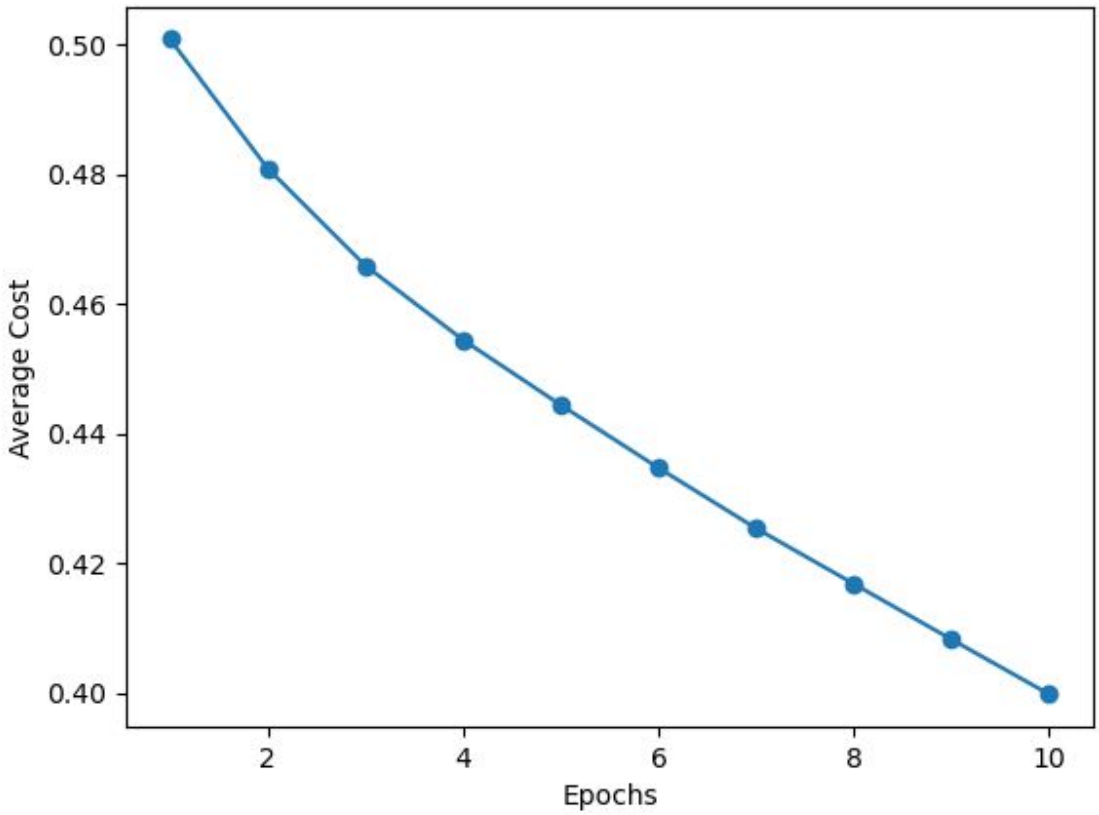
And has an accuracy of 100% (when the guess is off-by-one). This makes me think that Adaline is much better using extremely low learning rates than Perceptron.

SGD

When trained with eta=0.01, and 10 iterations, SGD was 99% accurate. Here is a graph of cost for all 10 iterations:



It would seem that the most dramatic effect happens from the first to the second epoch, where cost is dramatically reduced and fluctuates until epoch 10. And with a lower learning rate (eta=0.0001), SGD does much worse than before:



Though average cost is decreasing, the previous learning rate was better as the average cost was actually kept much lower than the average cost in Epoch 10 (when eta=0.0001).