Ryan C. Adams
CS 487

Project 1: Single-layer Linear Neural Networks

Foreword:

This report covers the use of single layer linear neural networks. Specifically we implement perceptron, adaline, and stochastic gradient descent. Moreover we use a classic iris data set from the UNCI website, and the red-wine data set from the kaggle/unci website. Credit for those datasets go to Kaggle, UNCI and the original authors of the papers that the datasets were derived from.

Overall Ideas:

This project allows the us to put forth hypothesis about the data, and see if the data supports that hypothesis given these single-layer linear neural networks. It should be noted that these algorithms do not guarantee a correct answer or necessarily the same answer.

Hypothesis:

In the iris dataset we explore the hypothesis that sepal length and pedal length can predict class, but specifically the Setosa class of Iris.

In the wine data set we explore the hypothesis that fixed acidity and volatile acidity can predict whether someone will rate a wine of a quality of 8.

The idea of linear separability, or simply being able to draw a line through the data points such that one category from another is the distinguishing characteristic. If a line can be drawn for our binary classifier such that such that we correctly predict (our positive class) from other classes (negative class) the we will have a high if not perfect accuracy. If a line cannot be draw or can only distinguish some of the positive class then our accuracy will decrease and the classes are not linearly separable.

Feature Scaling:

A quick note on feature scaling, for me on both data sets, feature scaling did not impact the rate of convergence. A couple possible reasons for this, my implementation has some error. The datasets are not large enough for it to dramatically impact the convergence rate. The attributes chosen may not be varied enough to drastically impact the convergence rate.

Variation Parameters:

Varying parameter greatly impacts results.

Eta - if eta is too large the data set can bound toward infiniti and not converge. In the middle ground and this was observed especially in adaline, the cost or average cost, can bounce up and down, but never settle into a minimum. Generally, we found that using a smaller eta was preferable.

Iterations - Iterations also play a large role in these algorithms. Increasing the number of iterations helps to expose patterns, and see if the algorithm appears to be trending towards a convergence point. However, there is a trade off, sometimes the algorithm will converge quickly, making extra iteration unnecessary and slow down processing. While other times the algorithm will bounce around before settling towards convergence after many iterations.

Findings:

In the iris data set we can conclude that that data is linearly separable using the sepal length and pedal length. The three algorithms, perceptron, adaline, and stochastic gradient descent produce accuracy of 100% and have convergence.

In the wine data set we can conclude that the data is not linearly separable and thus fixed acidity and volatile acidity cannot demark a wine quality of 8.   We can say this because perceptron never converges, and produces a 97% + accuracy. And while 97% is impressive, a wine quality rating of 8 on 18 of the 1600 wines observed. This is roughly 1%. For perfect linearly separability we would expect accuracy of 100%, somewhere between above 99% for real separability to occur. For both adaline and stochastic gradient descent, and adaline, we observe that cost and average cost decrease significantly, and move toward convergence, but we do not see accuracy change. This evidence supports our conclusion that the data is not linearly separable because accuracy stays at around 98.8 percent and we would expect much higher if the data were sepeable.

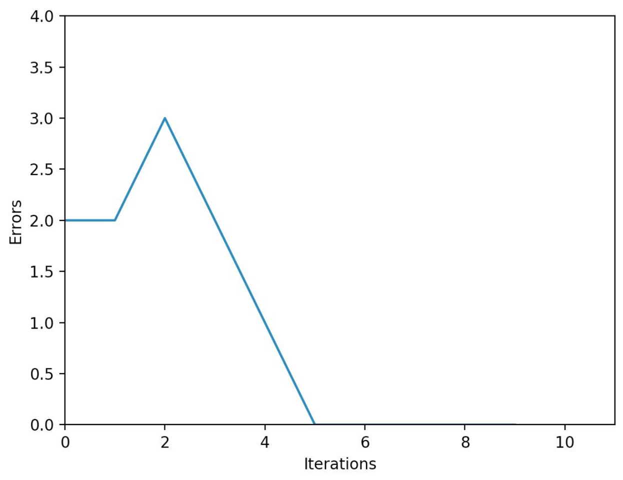
Classifier Reports:

{Continued on next page for continuity}

Classifier: Perceptron
Data set: iris.csv
Eta: .01 | Iterations: 10

Iteration	Error	Accuracy
0	2	98.667
1	2	98.667
2	3	98.
3	2	98.667
4	1	99.333
5	0	100
6	0	100
7	0	100
8	0	100
9	0	100

Errors vs. Iterations: Perceptron - Iris.csv

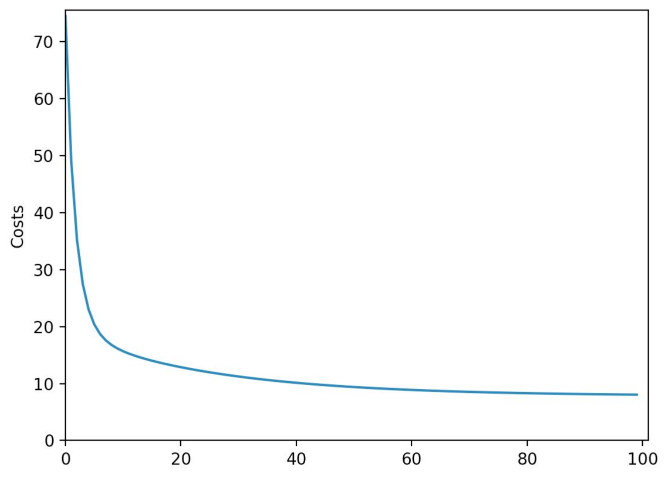


Classifier: Adaline
Data set: Iris.csv
Eta: .001 | Iterations: 100

For the sake of brevity I only include the first 10 entries. A more comprehensive list is available in command line.

Iteration	Cost	Accuracy
0	74.531	96
1	21.477	97.333
2	17.321	97.333
3	15.739	97.333
4	14.702	98
5	13.884	98
6	13.190	98
7	12.583	99.333
8	12.046	99.333
9	11.571	99.333

Costs vs. Iterations: Adaline - Iris.csv

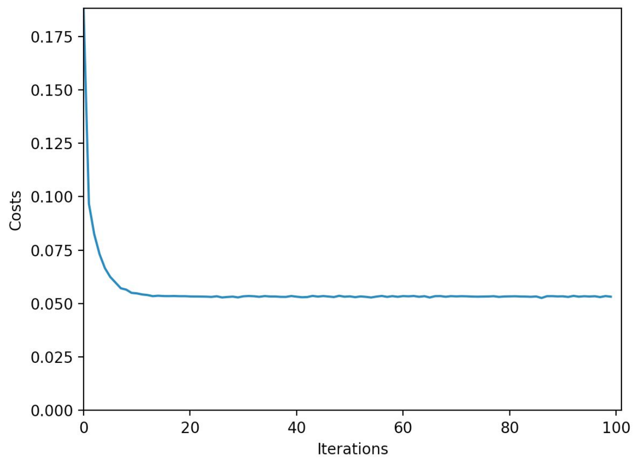


Classifier: stochastic Gradient Descent
Data set: iris.csv
Eta: .01 | Iterations: 100

For the sake of brevity I only include the first 10 entries. A more comprehensive list is available in command line.

Iteration	Avg_Cost	Accuracy
0	0.188	97.333
1	0.098	98
2	0.082	99.333
3	0.073	99.333
4	0.066	99.333
5	0.062	100
6	0.059	100
7	0.057	100
8	0.056	100
9	0.055	100

Avg_Costs vs. Iterations: SGD - Iris.csv

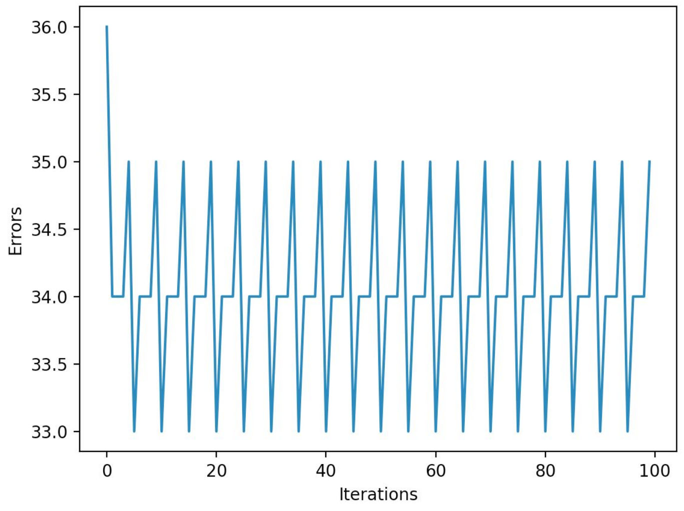


Classifier: Perceptron
Data set: winequality-red.csv
Eta: .1 | Iterations: 100

For the sake of brevity I only include the first 10 entries. A more comprehensive list is available in command line.

Iteration	Error	Accuracy
0	36	97.749
1	34	97.874
2	34	97.874
3	34	78.874
4	35	97.8111
5	33	97.936
6	34	97.874
7	34	97.874
8	34	97.874
9	35	97.8111

Error vs. Iterations: Perceptron - winequality-red.csv

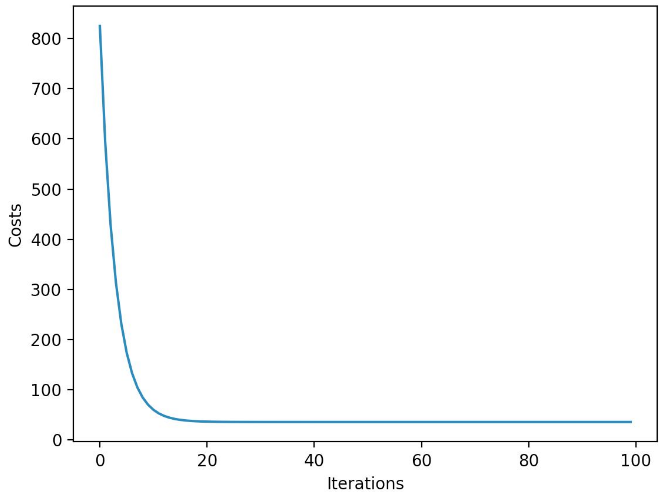


Classifier: Adaline
Data set: winequality-red.csv
Eta: .0001 | Iterations: 100

For the sake of brevity I only include the first 10 entries. A more comprehensive list is available at the command line.

Iteration	Cost	Accuracy
0	825.334	98.874
1	592.879	97.874
2	428.841	97.874
3	313.079	98.874
4	231.383	98.874
5	173.726	98.874
6	133.034	98.874
7	104.315	98.874
8	84.045	98.874
9	69.738	98.874

Costs vs. Iterations: Adaline - winequality-red.csv

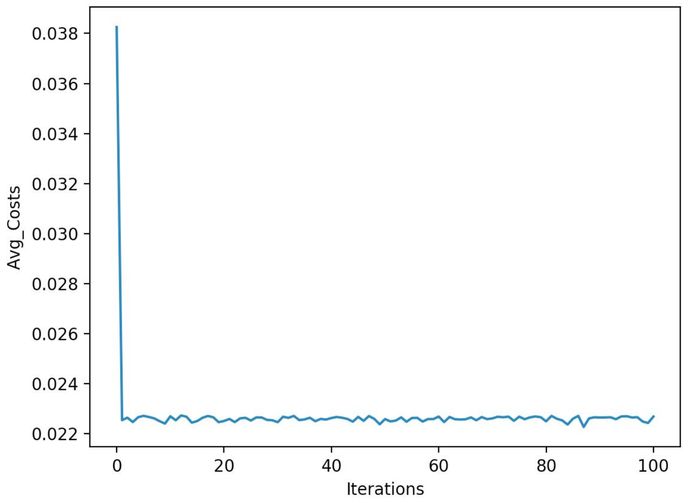


Classifier: Stochastic Gradient Descent
Data set: winequality-red.csv
Eta: .01 | Iterations: 100

For the sake of brevity I only include the first 10 entries. A more comprehensive list is available at the command line.

Iteration	Cost	Accuracy
0	0.0382534960785	98.874
1	0.0225375948171	97.874
2	0.0226353084542	97.874
3	0.0224603289267	98.874
4	0.0226542375021	98.874
5	0.0227099513997	98.874
6	0.0226632894508	98.874
7	0.02260627948	98.874
8	0.022495233706	98.874
9	0.0223993293844	98.874

Avg_Costs vs. Iterations: SGD - winequality-red.csv


Avg_Costs
Iterations