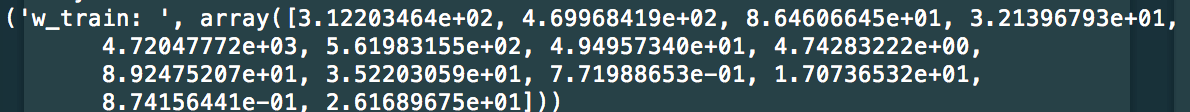
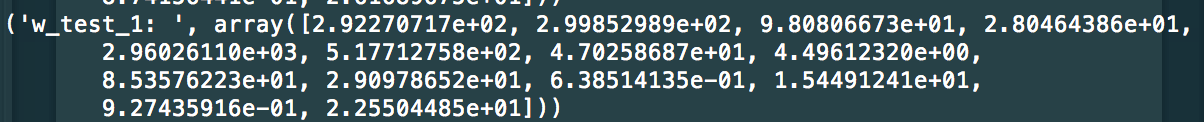
**Part 1**

1. Train Weight Vector:



Test Weight Vector:



2.

Train ASE = 62,249,111.49057

Test ASE = 3,098,592,132,898.01

3.

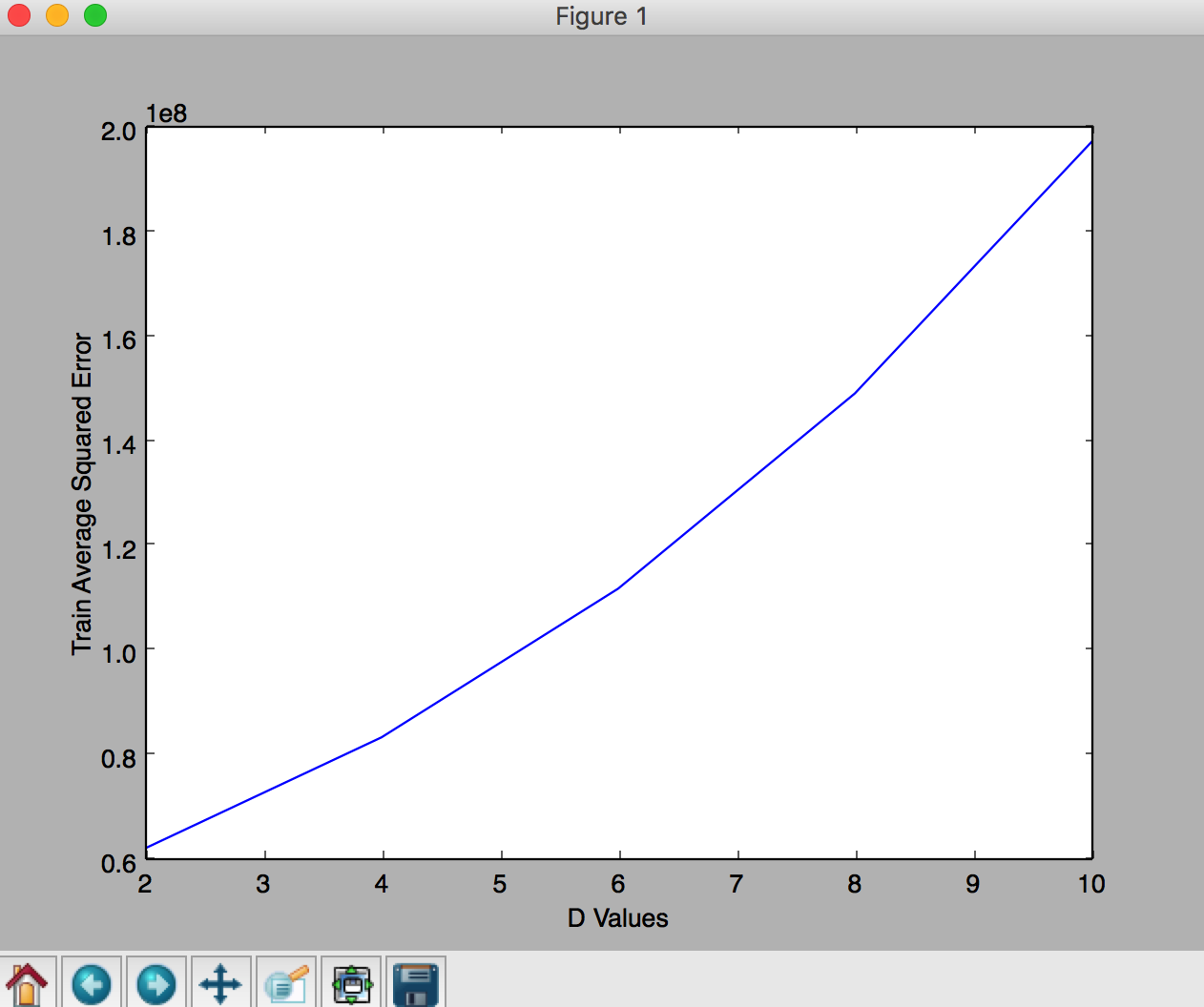
Getting rid of the dummy column of ones decreases the train average sum error slightly and dramatically decreases the test average sum error. This likely means that the column of one’s negatively correlates with the other features. Having the bias causes less weight to be given to the important features when predicting outcome. This causes an increase in error.

Train ASE Without Column of Ones = 53,850,369.889

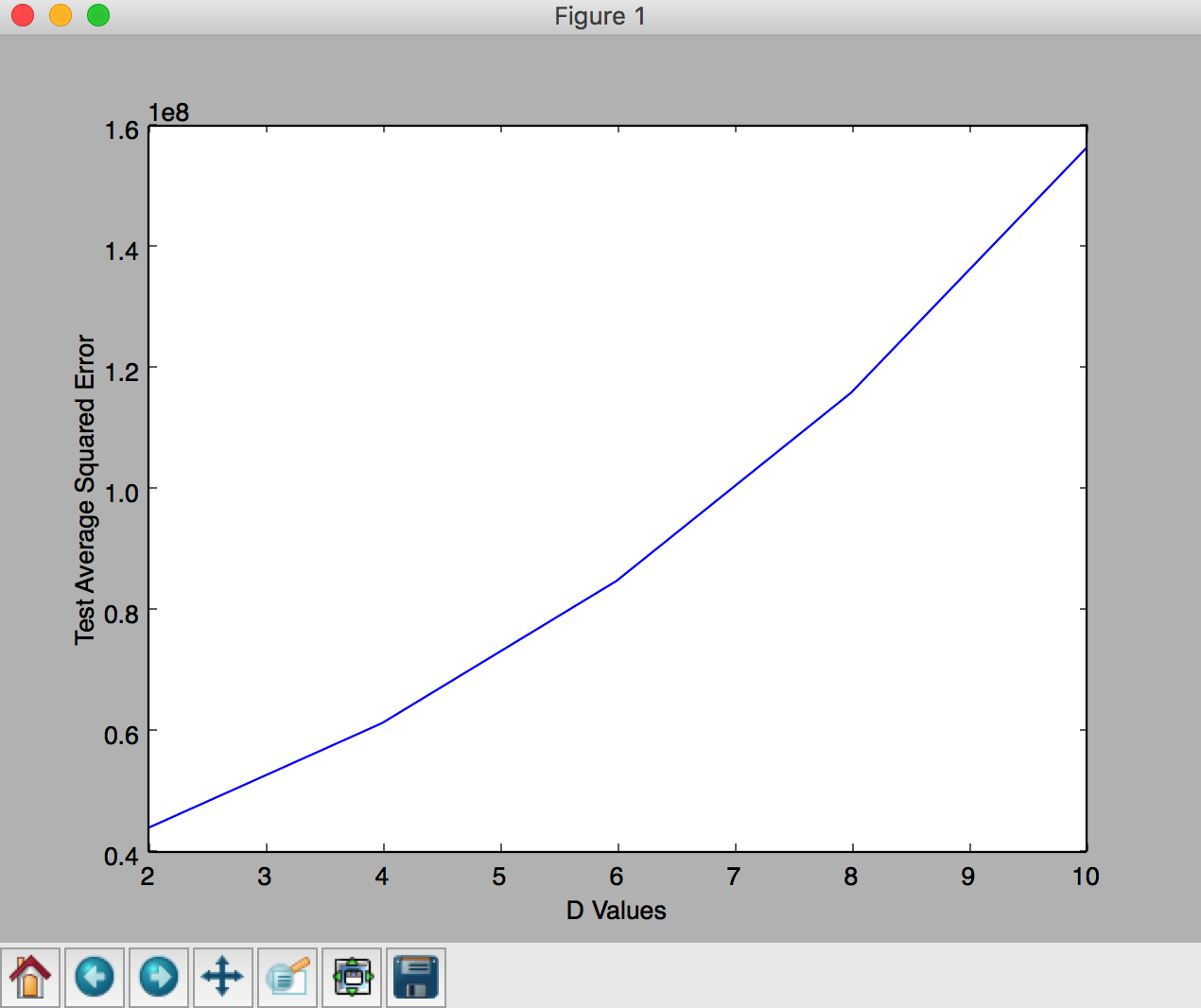
Test ASE Without Column of Ones = 37,292,977.9029

4.

Train ASE vs. D value plot:

****

Test ASE vs. D value plot:

****

The trend that I noticed for the training and test data is that they show a very similar graph that looks like a linear function. More features provide worse prediction performance at the testing stage. Randomly adding features that don’t positively correlate, the more variables that we add which negatively correlate with performance, the larger the ASE value. This won’t always be the case when randomly choosing additional features, but with our data, they negatively affect the performance.

**Part 2**

1. Gradient Descent:

With 10 epochs

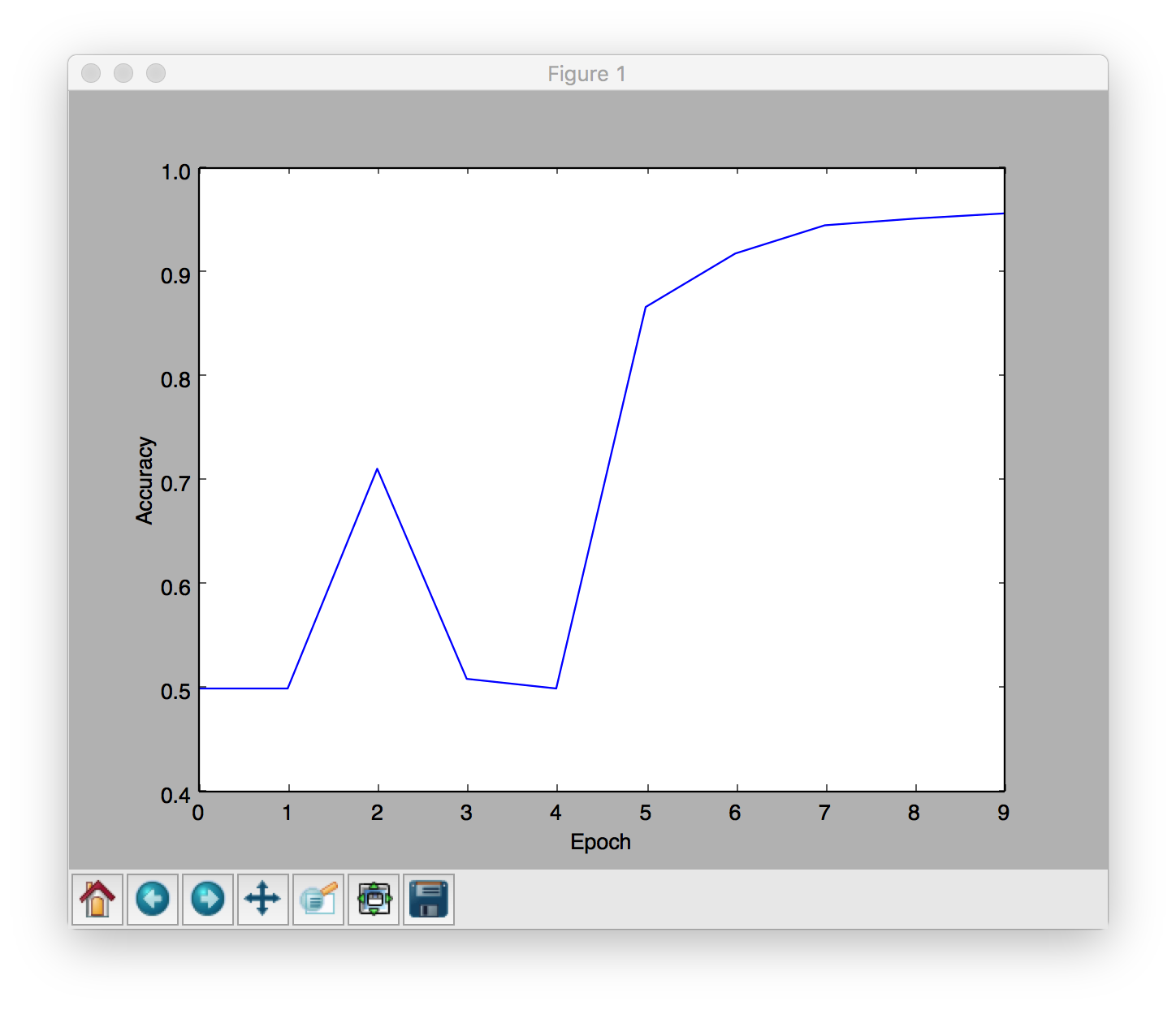
.0000001: Accuracy 47.857%

.0001: Accuracy 88.64%

.01: Accuracy 95.64%

1: Accuracy 95.7%

The overall trend looks like this:



3. Gradient Descent pseudocode for batch learning with L2 Regularization:

For inputs X[1, … ,n] and labels Y[1, … , n]: