Ian Zimmer

0028719394

I have used 2 late days on this assignment

Note: My algorithms sometimes hit a low accuracy of around 69% accuracy because of the random aspect. This rarely happens, maybe 1 out of 20 times. I don't know if this is allowed but please run it again if this happens because this is very rare and does not represent the work I have put into this project.

2: Theory

2.1: In the gradient descent algorithm, all samples in the data set are iterated through in order to update the parameter w. In the stochastic gradient descent, only a subset (usually 1) of samples in the data set are iterated through in order to update the parameter w. While using a large training set, it is better to use SGD because the gradient descent may take too long since the whole data set has to be traversed for every update. If the data set is smaller, it is better to use GD since the SGD algorithm doesn't minimize the error function as well as GD.

2.2: The model has converged when the step size is very close to 0. The criteria would be when the step size is lower than .001, then stop the gradient descent.

2.3: The bias term allows us to move the intercept of the line that determines the loss values. Without the bias, we would only be able to move the slope of the line, but a bias allows us to find a more predictive line.

2.4: True. It only looks at one sample verse the whole sample set.

2.5: This is because SGD only looks at one sample at a time. If all examples are clustered together, then the line will be fit to only that cluster instead of the entire dataset.

2.6: $dL(w)/dw = sum(-y\ xi)$ if $y\ (x*w) < 1$ or 0 if $y\ (x*w) > 1$

2.7: Log loss:

$dL(w)/dw = -sum(-g(wxi) + yi)xi + \lambda\|w\|$

Hinge loss:

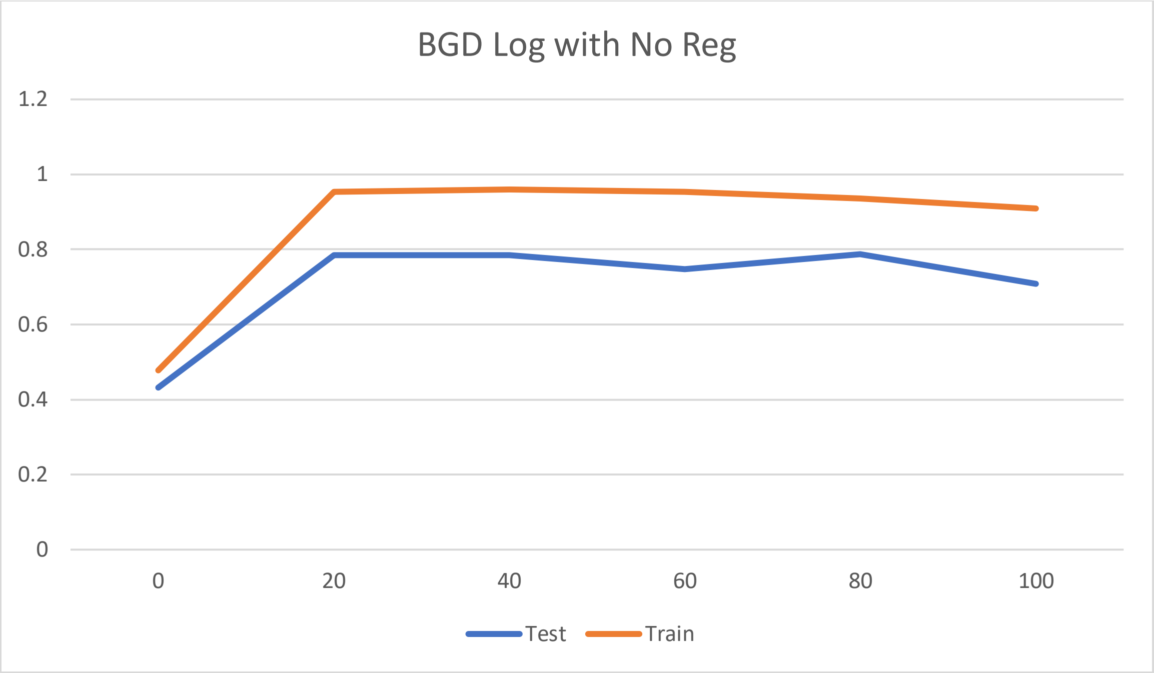
$dL(w)/dw = sum(\lambda\|w\| -y\ xi)$ if $y\ (x*w) < 1$ or 0 if $y\ (x*w) > 1$

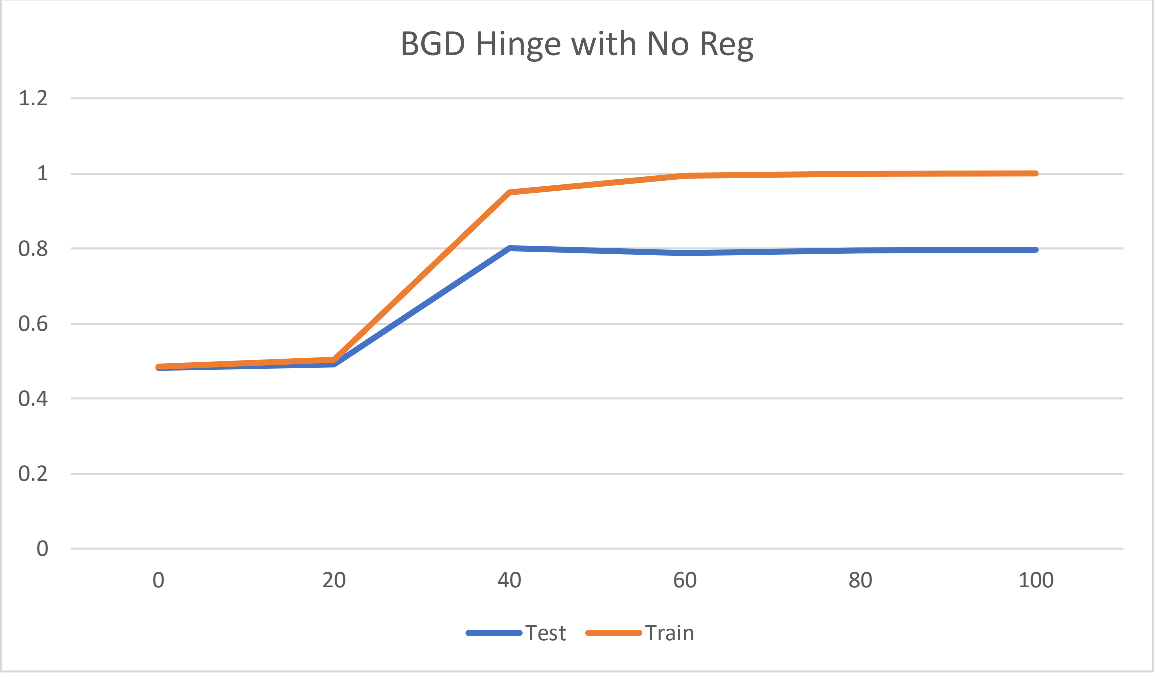
2.8: Regularization is used to reduce overfitting of the data. Since the goal of regularization is to penalize gradients with high coefficients, a negative lambda will decrease the effect of regularization.

3.1

```
While not converged
    Initialize an array called gradients (size vocab_size) to zero
    For each sample in train_data
        For each feature in each sample
            Get the gradient of each feature
            Do this with the formula gradients[feature] = gradients[feature] – (feature * (label – sigmoid(weighti, feature))
    Update the weights = weights – (gradients * learningrate)
Return weights
```

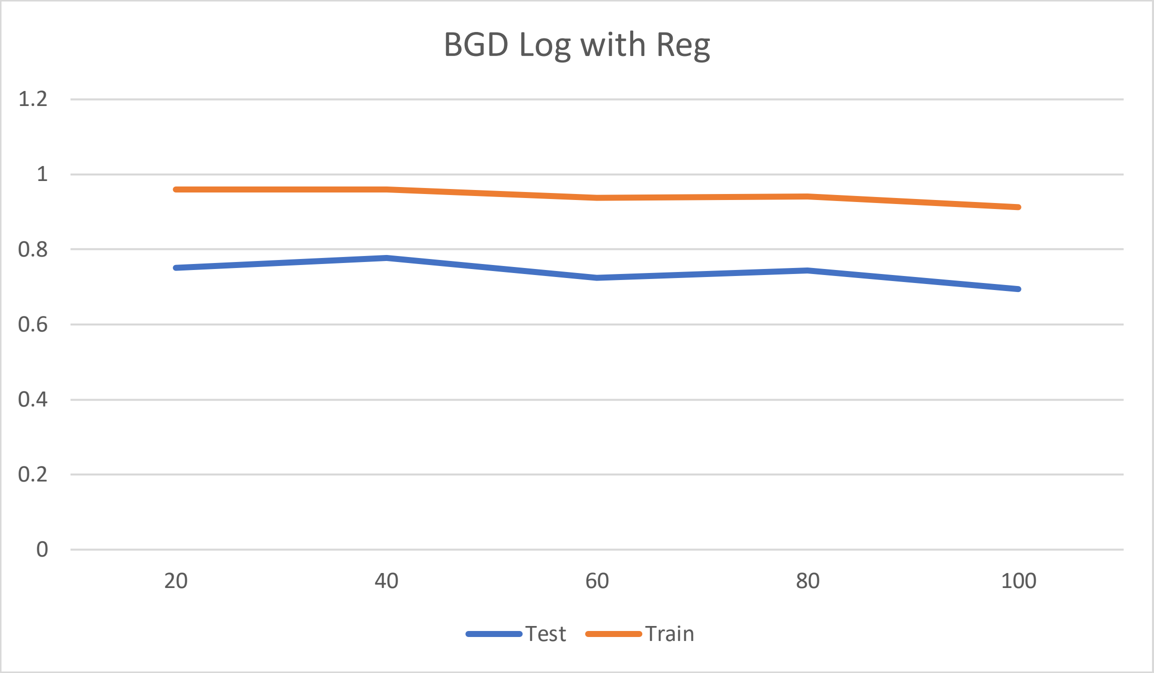
3.3.1

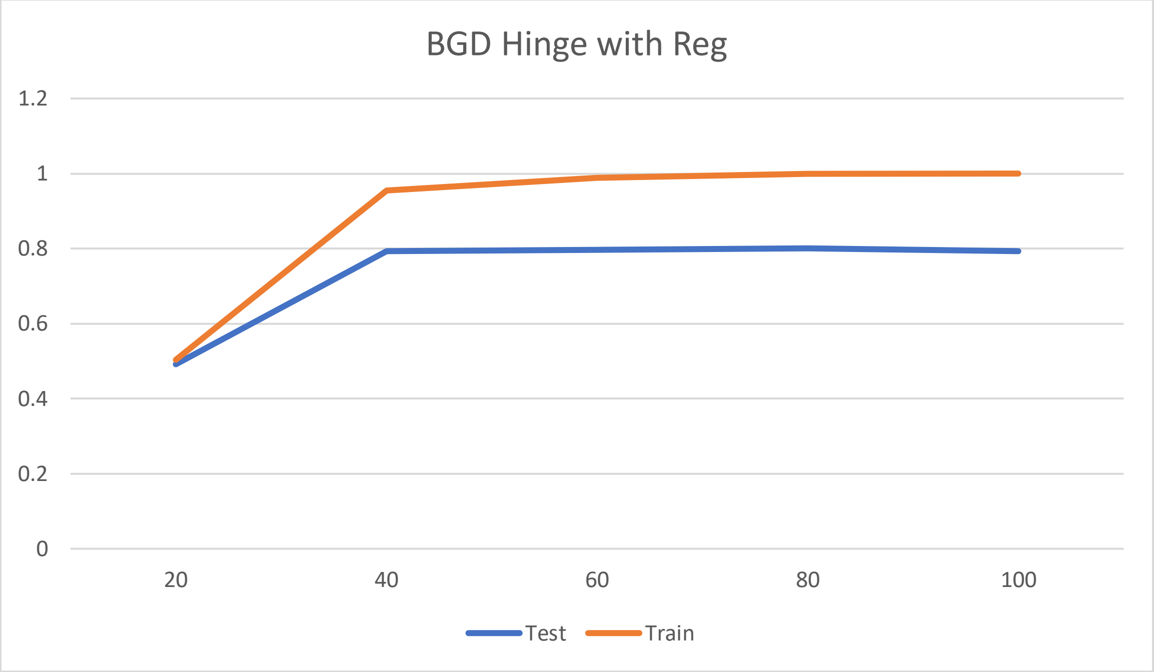




3.3.2

Cannot calculate accuracies at 0 because of a division by zero error

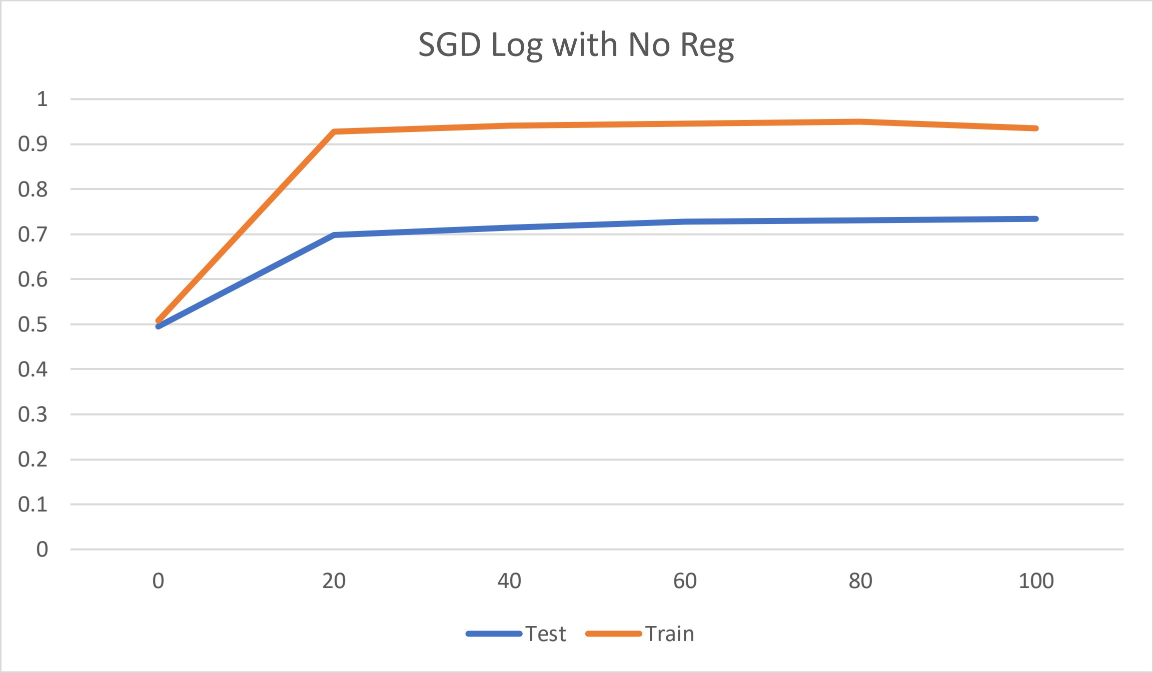


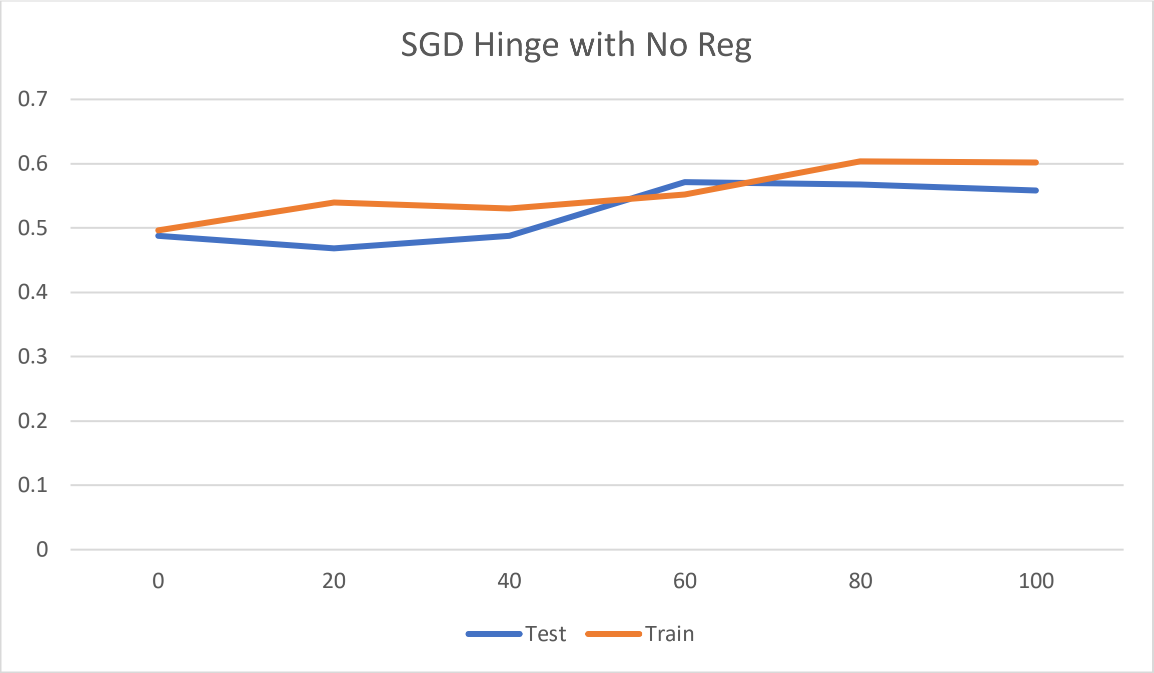


4.1

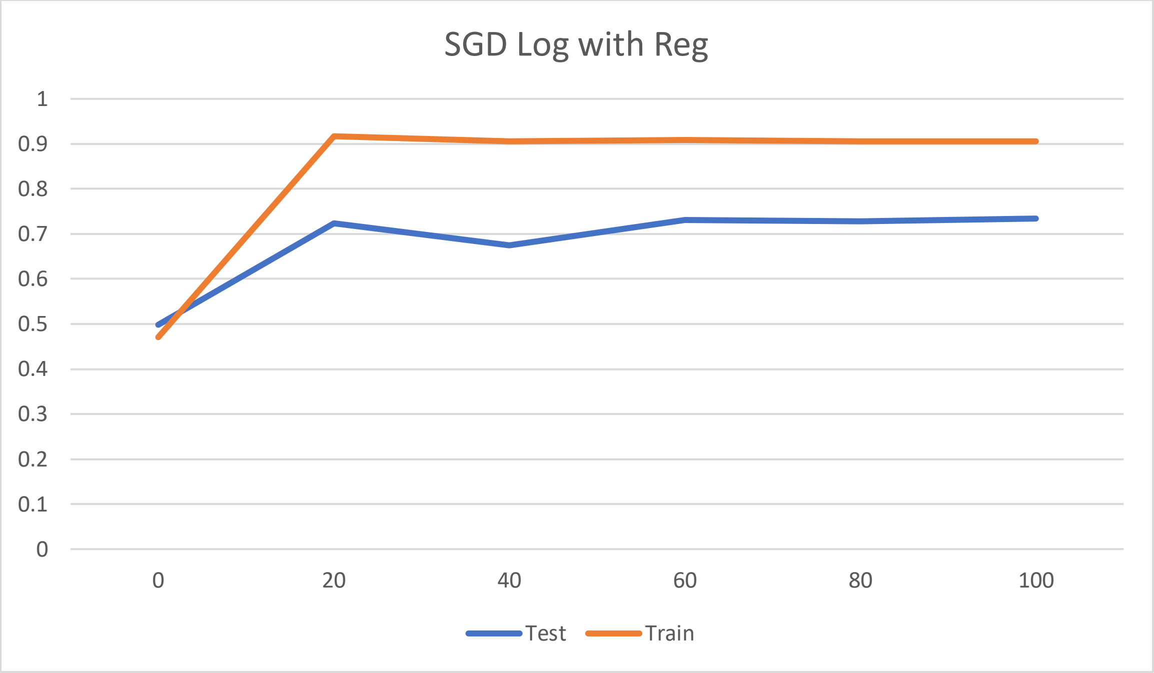
```
While not converged
    Initialize an array called gradients (size vocab_size) to zero
    For each sample in train_data
        If (sample * weights) X label < 1
            For each feature in each sample
                Get the gradient of each feature
                Do this with the formula gradients[feature] =
                gradients[feature] – (feature X label)
            Update the weights = weights – (gradients * learningrate)
Return weights
```

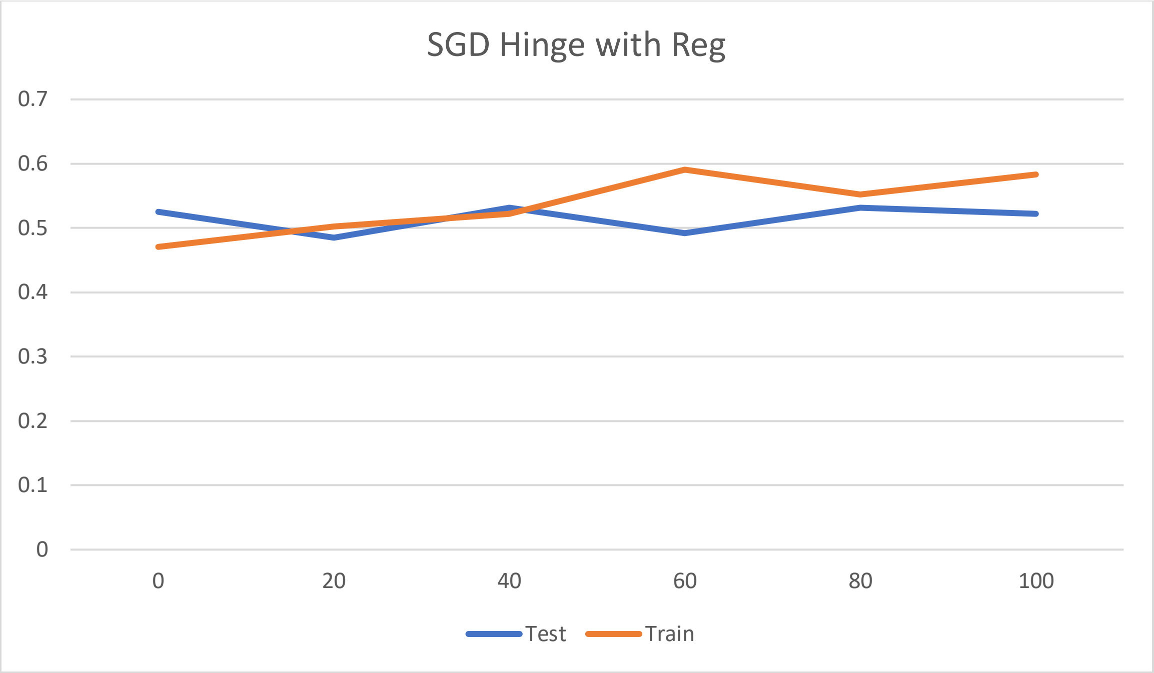
4.3.1




SGD Hinge with No Reg
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
Test
Train

4.3.2


SGD Log with Reg
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
Test
Train


SGD Hinge with Reg
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
Test
Train