Rohith Ravindranath
PUID: 0028822977
Dan Goldwasser
CS 37300
23th April 2019

Homework 4 (***USING ONE DAY LATE PASS***)

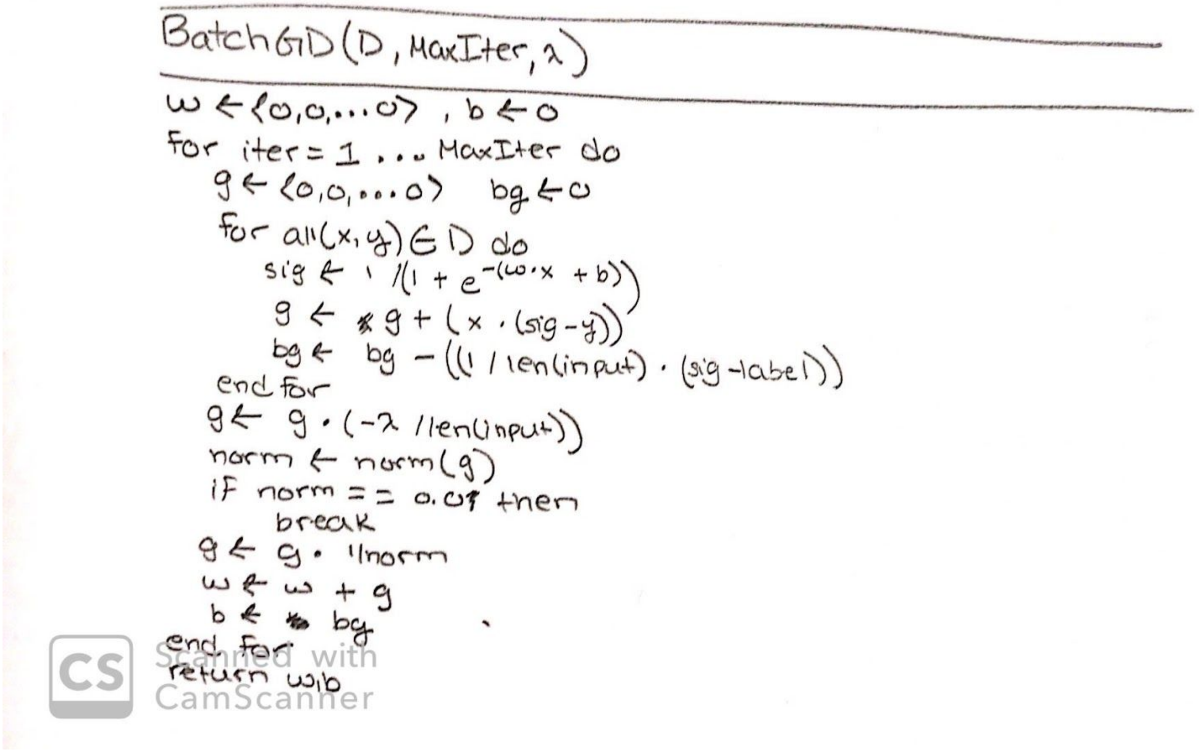
2. Theory

1. When it comes to Batch Gradient Descent, the update on the weight vector and bias occurs after one full iteration over the entire dataset. For Stochastic Gradient Descent, the update on the weight vector and bias occurs after we look at each individual example in the dataset. We would prefer Batch Gradient Descent when the size of the dataset is reasonable that traversing the entire dataset multiple times is feasible. We would prefer Stochastic Gradient Descent when the dataset is too large or we have are constantly receiving data where we cannot iterate through the entire dataset in one iteration.
2. One stopping criteria is when the norm of the vector becomes very small or zero. This equates to our gradient not making a significant change or we are as close to the minimum to the loss function as we can be. The other stopping criteria is when we have reached the number of iterations that we manual set.
3. The bias term is used to shift the model based on the sigmoid function.
4. True. Batch Gradient Descent will compute the gradient over the entire dataset and then update the weights. Where in Stochastic Gradient Descent, will look at one example and then update the weights. Due to this, the Batch Gradient Descent does more computation per update.
5. We prefer to randomize our data since we don't want to have a stream of positive class labels and then negative class labels. This would induce a bias in our model towards negative class labels. This is how the dataset we are given is. The first half is all positive labels while the second half is negative labels
6. When it comes to hinge function, the function is not differentiable a x=1. Due to this problem we will need to use the subgradient to find the gradient of the hinge loss function. The gradient is:
   $\vartheta_w max(0, 1 - y_n(w * x_n + b))$
   $\vartheta_w \; if \; (y_n(w * x_n + b)) \; > \; 1 \; is \; \vartheta_w 0$
   $\vartheta_w \; if \; (y_n(w * x_n + b)) \; <= \; 1 \; \vartheta_w(y_n(w * x_n + b))$
   $\vartheta_w 0 = 0$
   $\vartheta_w \; (y_n(w * x_n + b)) \; = \; y_n * x_n$
   Final Gradient
   $If \; (y_n(w * x_n + b)) \; > \; 1 : 0$
   $Else : y_n * x_n$
7. Log Loss Gradient: $\Sigma((\sigma(z) - y_i) * x - \lambda * w)$
   Hinge Loss Gradient: $if \; if \; (y_n(w * x_n + b)) \; <= \; 1 : \; (y_i x - \lambda * w)$

8. We use the regularization line of our code to make sure that the gradient vector doesn’t grow too much when an error is found on the data points that are not as impactful as other data points. This way our model doesn’t get too skewed towards on side. If our lambda was negative, then instead of subtracting from the gradient, we would be increasing the gradient. This would greatly hurt our model.

3. Batch Gradient Descent

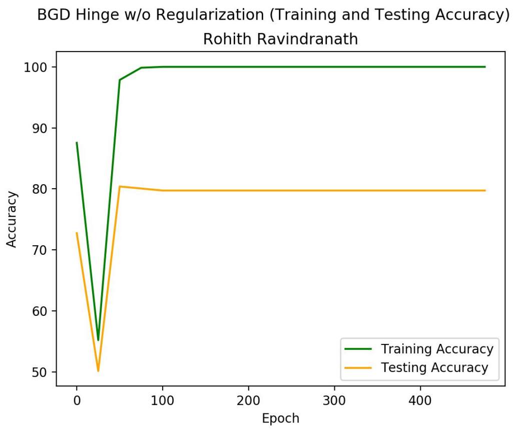
3.1 Algorithm

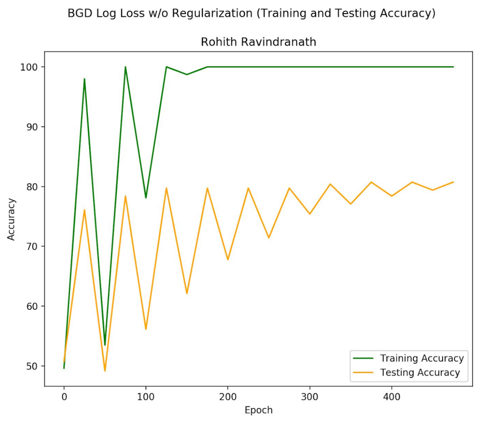


```
BatchGD(D, MaxIter, $\lambda$)
$w \leftarrow \langle 0,0,\ldots 0 \rangle$ , $b \leftarrow 0$
for iter = 1 ... MaxIter do
    $g \leftarrow \langle 0,0,\ldots 0 \rangle$   $bg \leftarrow 0$
    for all$(x, y) \in D$ do
        $sig \leftarrow 1/(1 + e^{-(w \cdot x + b)})$
        $g \leftarrow g + (x \cdot (sig - y))$
        $bg \leftarrow bg - ((1/\text{len(input)}) \cdot (sig - label))$
    end for
    $g \leftarrow g \cdot (-\lambda/\text{len(input)})$
    $norm \leftarrow \text{norm}(g)$
    if $norm == 0.01$ then
        break
    $g \leftarrow g \cdot 1/norm$
    $w \leftarrow w + g$
    $b \leftarrow bg$
end for
return w,b
```

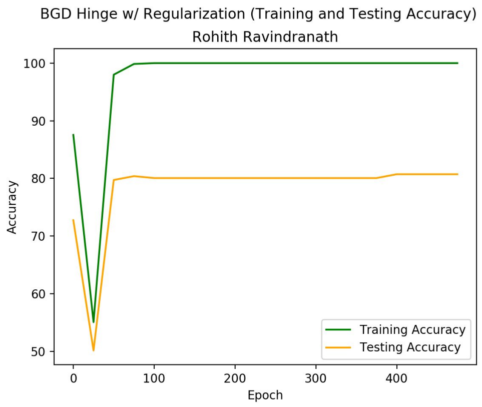
3.3 BGD Analysis

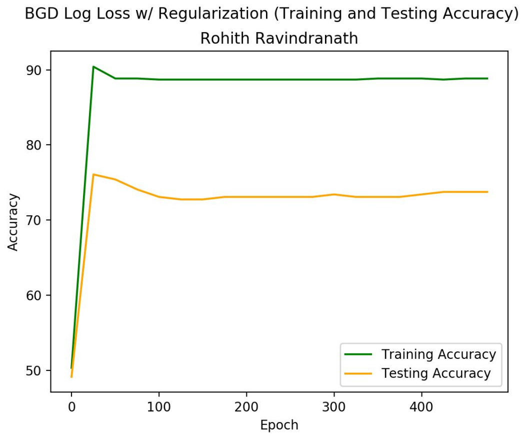
1.





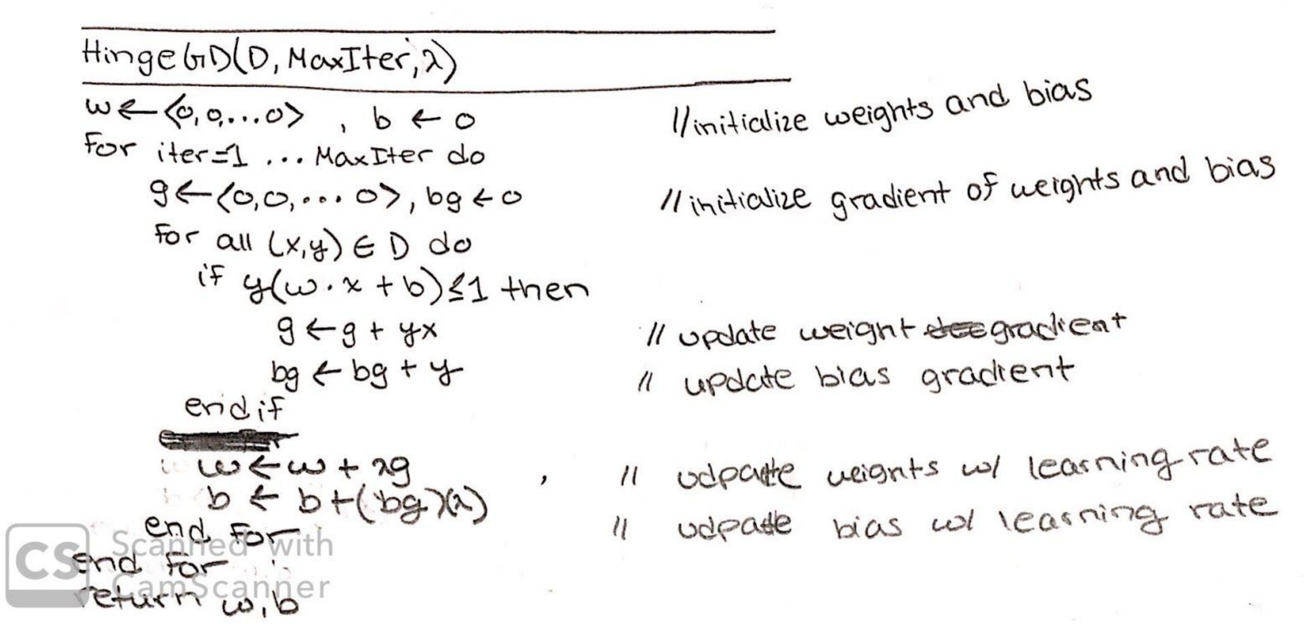
2.





4. Stochastic Gradient Descent

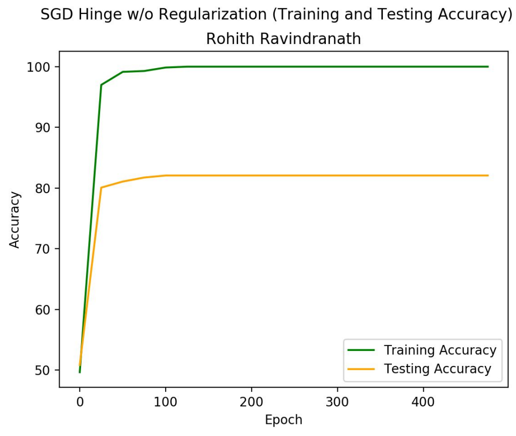
4.1 Algorithm

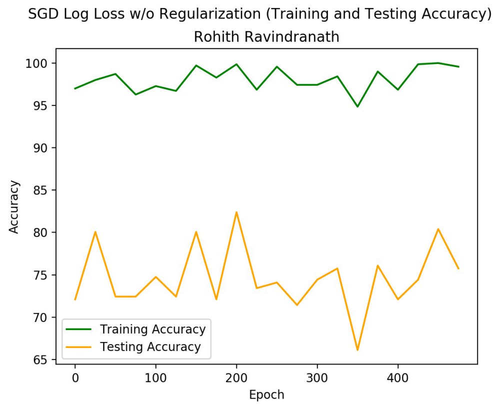


```
HingeGD(D, MaxIter, λ)
w ← ⟨0, 0, ... 0⟩ , b ← 0                         //initialize weights and bias
For iter=1 ... MaxIter do
    g ← ⟨0, 0, ... 0⟩, bg ← 0                     //initialize gradient of weights and bias
    for all (x,y) ∈ D do
        if y(w · x + b) ≤ 1 then
            g ← g + yx                            // update weight ~~dee~~gradient
            bg ← bg + y                           // update bias gradient
        endif
    w ← w + λg ,                                  // udpate weights w/ learning rate
    b ← b + (bg)(λ)                               // udpade bias w/ learning rate
    end for
end for
return w, b
```

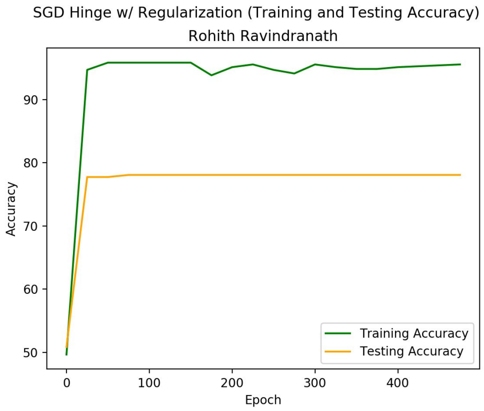
4.3 SGD Analysis

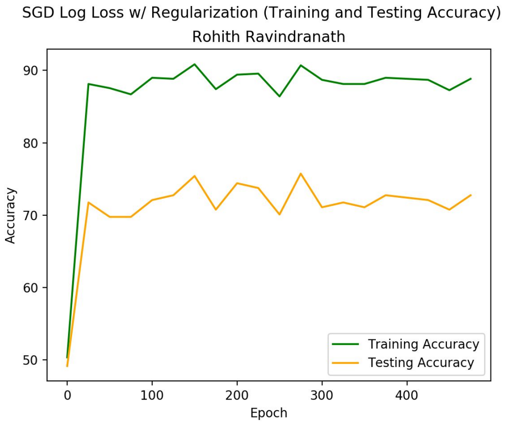
1.





2.


SGD Hinge w/ Regularization (Training and Testing Accuracy)
Rohith Ravindranath
Accuracy
90
80
70
60
50
0
100
200
300
400
Epoch
Training Accuracy
Testing Accuracy


SGD Log Loss w/ Regularization (Training and Testing Accuracy)
Rohith Ravindranath
Accuracy
90
80
70
60
50
0
100
200
300
400
Epoch
Training Accuracy
Testing Accuracy