Cs 373 hw 4 Xiao Qin
Cr statement: Dan Luo， Aarushi Banerjee

2 theory:
1) difference between batch gradient descent and stochastic gradient descent.

When would you prefer one over the other

	Batch gradient descent	Stochastic gradient descent
Difference	We update the weight after going through all trainlng example	Shuffle the training data at the beginning of each iteration
When each is preferred	When I don't have too many data	We we have too many data

2)in gradient descent, stop training when the model converge, how do you know the model converge. If the gradient becomes 0 or approaching 0
State the stopping criteria and apply the criteria thought your implementation

I set the stop criteria as np.linalg.norm(self.gw)<0.0001

3)What will be the effect of bias term in BGD/SGD learning? (2 points)
True/False. Stochastic gradient descent performs less computation per update than batch gradient to adjust data to let the line passing through origin?

The bias term in the linear model is to shift the line to improve the confidence of separating the positive and negative group. In the BGD/SGD, there are bias associated with weight and bias associated with gradient. For the bias associated with weight we use it to better linearly classify the data points, for the bias term is it is being calculated to follow the likelihood function of weight.

Yes the stochastic gradient descent performs less computation per update than batch gradient the data set has more than 1 example.

Why do we randomly shuffle the training examples before using SGD optimization? (2 points)

Because we want to make sure the choice we make is random or stochastic

Hinge Loss which was introduced in class is not differentiable everywhere unlike log loss. Then show how we can use hinge loss function in the gradient descent optimization algorithm. What will be the gradient of hinge loss function without regularization term? (5 points)

Delta w max{0, 1-yn(w*xn+b)}

=delta w{ 0 if yn(wxn+b)>1

yn(w*xn+b) otherwise

=delta{ 0 if yn (w*xn+b)>1

ynxn otherwise

What will be the gradient of the log and hinge loss function if you add the L_2 regularization term $(1/2)\lambda||w^2||$? (5 points)

	Log loss	Hinge loss
Gradient + L2	Alpha*sigma d=1 to D (yd-wi*xd)xdi +lamda w	if if yn (w*xn+b)>1: lamda *w Else: yn xn+ lamda *w

Why is regularization used? What's a potential issue with L_2 regularization, if the λ hyper-parameter can take negative values? (5 points) Regularization is used to avoid overfitting. If it takes the negative value it causes noise. For the gradient function the lambda is reduced from exponent of 2 to 1. So the negative will show the direction in the gradient function instead of just absolute value in the log likely hood function.

3 batch gradient descent

3.1Algorithm (5 points)

Write down the batch gradient descent algorithm for log loss in appropriate algorithmic format.

```
#BGD
#initialize w0 with (0…0)
#b0=0
#For i=0...max_Iter:
```

```
#      delta w=(0,0,...,0)
#.     delta b=0
#.     For every training example in x: 699 j from 0 to 699
#              delta w += -(y-sigmoid(xj))*xj
#              delta b +=sigmoid(xj)-y
#       if |delta w| approach to 0
#.          break
#      //update the weight and weight's bias at the end of each iteration and regularize
#      delta w+=lambda*weight
#       w+=-learningrate/m delta w. // m is the number of examples
#       b+=- learningrate/m delta b
```

3.2implementation

3.3 BGD Analysis

One graph for hinge loss, one graph for log loss without regularization

Run the same experiments with regularization and plot the training and test accuracy for each epoch again for each type of loss function

4.stochastic gradient descent

4.1 write the stochastic gradient descent algorithm for hinge loss in appropriate algorithmic format

```
#initialize w0 as (0,…,0)
#b0=0

#For i=0…max_Iter:
#delta w=(0,0,...,0)
#delta b=0

#    shuffle the training data looping through all examples
#     if y(w*x+b)<=1 then
#         delta w += -y*xj
```

```
#         delta b +=-y #
#      End if
#.      delta w+=lamda *w.       //add in regularization term
#       w+=-learningrate* delta w
#       b+=-learningrate* delta b
#      if |delta w| approching 0
#.          break
#return w,b
```

4.2 implementation

4.3 SGD Analysis:
1.Plot graph without regularization
2. Plot graph with regularization