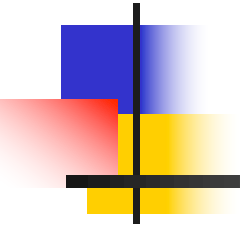


CMSC 510 – Fall 2020



Homework Assignment 3

Announced: 10/6

Due: Tuesday, 10/27, noon



The problem

- Implement and test:
 - Logistic regression (LR) with L_1 regularization
 - LR is differentiable
 - But L_1 norm is not
 - Use proximal gradient descent
 - For L_1 norm, that's soft-thresholding
 - Use tensorflow library
- Dataset – the same as in HW2:
 - Classify two digits from MNIST dataset



Hints about tensorflow

- See: `tensorflow_minimizeF.py`
 - Performs projected gradient descent on a simple function
 - The function has global minimum at
 - $w_1 = -0.25, w_2 = 2$
 - But the feasible set Q is: $w_1 \geq 0, w_2 \geq 0$
 - For this function, the best solution is $w_1 = 0, w_2 = 2$
 - The code does the following, in a loop:
 - Gradient step on the function, followed up by proximal step
 - Here, the proximal step is just “make w nonnegative” by replacing negative values with 0, the closest non-negative value
 - Feasible set Q is set of all vectors with nonnegative coordinates, i.e., for 2D, $w_1 \geq 0, w_2 \geq 0$
 - In your actual code, you should use soft-thresholding instead



Hints about tensorflow

- See: `tensorflow_leastSquares.py`
 - Performs gradient descent on a function based on data
 - We have some fake data x, y , where $y = w * x + b + \text{small_gaussian_noise}$
 - The code tries to find best w_{best} , b_{best} that predict y
 - It uses the loss: $(y - y_{\text{predicted}})^2$
 - $y_{\text{predicted}} = w_{\text{best}} * x + b_{\text{best}}$
 - In your code:
 - x, y will be taken from the MNIST dataset
 - the loss should be *logistic loss*
 - you need to add the proximal step / soft-thresholding
 - Constant L is unknown, you should try several gradient step sizes
 - Constant in front of L1 penalty is unknown, you should try several values



Returning the Assignment

- Solution code should be written by you and you only (no web/book/friend/etc. code)
 - You can freely use the code provided on BB as your starting point
- Upload through Blackboard
 - A report in PDF
 - Results of tests of the method on MNIST dataset, for decreasing training set sizes (include you V#, and what are your two digits defining the two-class problem).
 - Code in python for solving the MNIST classification problem (for full size of the training set):
 - The file should have your name in a comment at the top