CIS526: Homework 4
Assigned: October 8th, 2021
Due: October 15th at 5pm

Homework Policy

All assignments are INDIVIDUAL! You may discuss the problems with your colleagues, but you must solve the homework by yourself. Please acknowledge all sources you use in the homework (papers, code or ideas from someone else). Assignments should be submitted in class on the day when they are due. No credit is given for assignments submitted at a later time, unless you have a medical problem.

Problems: hand it as pdf

Problem 1 (20 points). Let us suppose we want to learn a linear predictor $\sum_{j=0}^{M} w_j x_j$ on training data with *N* examples and *M* features that minimizes loss function defined as $Loss = MSE + \alpha \sum_{j=0}^{M} {w_j}^2$, where α is a regularization hyperparameter. Show that *Loss* can be written as a quadratic form (so, there is a closed form solution for **w**). Find the expression for **w** that minimizes *Loss*. Write this expression in a matrix form.

Problem 2 (15 points). Let us suppose we know that target variable can only be positive and decide to learn prediction function of type $relu(\sum_{j=0}^{M} w_j x_j)$. Remember $relu(x) = x$ if $x \geq 0$ and $relu(x) = 0$ if $x < 0$. Show that MSE of this predictor is not a quadratic form. We decided to use gradient descent algorithm to find **w** that minimizes MSE of this predictor. Derive the update formula in a vector form. EXTRA CREDIT: Show that MSE for this predictor has the global minimum (i.e, show that MSE is a convex function).

Problem 3 (15 points). Let us suppose we want to train on training data with *N* examples and *M* features a feedforward neural network with *H* hidden *relu* nodes in one hidden layer and one sigmoid output neuron. Assume that target *y* is a binary variable (*y* = {0,1}). The loss function is cross-entropy loss (practitioners like it as a loss function for binary classification) defined as $Loss = \sum_{i=1}^{N} y_i \log(nn(x_i, w)) + (1 - y_i)\log(1 - nn(x_i, w))$. Use backpropagation to derive gradient descent update for weights in the output neuron and for weights in the hidden layer (weight gradient in the hidden layer should be expressed using gradient of weights in the output neuron). You can derive an update for each separate weight using partial derivatives, but it is preferable if you use gradients to express the updates in the matrix form.

Programming Assignment: hand it as an ipynb file and a pdf file for the 1-page report

Problem 4 (50 points). Download “CIS5526 2020 – Homework4.ipynb” and solve 4 problems.