Neural Networks lab (The LMS algorithm)

Nicole Beckage

March 8, 2017

Lab adapted from Mike Mozer's hw 1 from Neural Networks and Deep Learning (7222)

Goal
The goal of this assignment is to introduce neural networks in terms of ideas you are already familiar with: linear regression and linear-threshold classification.

Data Set
The data set is synthetic data generated such that X and Y have a linear relationship. X has 10 features and Y is a valued output. Also present is z, a binary classification value based on Y. The data files are called LMSalgtest.csv and LMSalgtrain.csv and are csv files where each line contains the data vector, output value and output class.

Part 1
Write code to read in a file – either train or test – and build a data structure containing the input-output examples. Find the least squares solution to $y = \boldsymbol{\beta} X$

You need a function *readInData* that takes in a file path/name and returns X (input) and Y(output) X is either a dataframe or matrix and Y is a vector. You need to include a function *OLSData* that takes in training data and training labels and returns the OLS solution in terms of $\boldsymbol{\beta}$. You should write another function *computeError* that takes in training/testing data and labels as well as the $\boldsymbol{\beta}$ vector and returns the MSE error.

Report the OLS solution as well as the testing and training MSE for the OLS solution.

Part 2
Write an LMS algorithm, a program that determines the coefficients w1,w2,...,w10, b via incremental updating and multiple passes through the training data. You will need to experiment with updating rules (online, batch, minibatch), step sizes (i.e., learning rates), stopping criteria, etc.

Report the experiments and what you've learned from these experiments in terms of online/batch/minibatch) as well as step size (learning rate). When should you stop training? Consider the train and test performance differences. Do not forget to use cross-validation.

Part 3

Turn this data set from a regression problem into a classification problem simply by using z (a rough thresholding of y) as representing one of two classes. In the data set you download, you'll see a variable z that represents this binary (0 or 1) class. Use the perceptron learning rule to solve for the coefficients w1, w2,...,b of this classification problem.

Note: if you do this right, your solution to Part 3 will require only a few lines of code added to the code you wrote for Part 2.

Using the test set, construct a confusion matrix. The 2x2 confusion matrix will specify the frequency by which each input is assigned to each output class.