Programming Assignment 3:

Implementing a Linear Regression Model and a Neural Network Model

This assignment has two problems.

1. Write a program to implement a linear regression model following the lecture. No need to consider regularization.

   Input: The input data will be given in a file in the following format, where the first row gives the attribute/variable names with the last variable as the output/predictive variable, and the rest are the data instances. Please use the same loss function as in the lecture and the batch gradient descent method for optimization.

   A1, A2, Y
   1.5, 2.2, 4.3
   20, 1.6, 3.5
   …

   Outputs:

   (1). Given a sequence of test instances in a file in the same format as the input (without the output variable), predict the value of the output variable and write the value in a text file in the same order as the input with one instance per line. In this case, we can have any number of input variables.

   (2). Please find a plot package to plot the training input data points, your output linear function, and the predicted value for each test instance. In this case, the data has only one input variable.

2. Implement a neural network classifier. Your network should have three fully connected layers, i.e., an input layer, a hidden layer and an output layer. Use the same activation function, loss function, and stochastic gradient descent (SGD) for optimization as described in the lecture slides.

   Input data: You will be given two kinds of inputs.

   (1). The input data will be given in a file in the following format, where the first row gives the attribute names, and the rest are the data instances. The input data may have any number of attributes/variables. The last attribute is the class attribute, which can have any number of classes. The attribute values are all numerical except the class attribute.

   A1, A2, A3, C
   1.5, 2.2, yes
   20, 1.6, no
   …

   (2). The number of neurons for each layer and the minibatch size for SGD. They will be given to you right before demo.

   Output: Given a sequence of test instances in a file in the same format as the input, predict the class for each of them and write the output class in a text file in the same order as the input with one instance per line as follows, where 1 and 0 are class labels,

yes
no
yes
..

Group project: This is a group project. Each group can have no more than 2 students.

Code submission requirements: (1) A **readme file** must accompany the **code** for each program to describe (i) what each function is for, (ii) which functions are for gradient descent for each program, and (iii) which functions are for forward pass and backward pass (backpropagation) for the neural network program. (2) Your code should also have clear comments. If we cannot find your functions for those functions, points will be deducted.

Programming language: You may use any language of your choice.

Demo date: April 28, 2022, from 9:30am – 12:30pm.

Deadline for code and readme file submission on Blackboard: April 27, 2022, 11:59pm.