Assignment2 - CS6304 Deep Learning. Due Date: 11:59 PM of 09/27/2021

Purpose: The purpose of this assignment is to test concepts on how a multilayer network peforms prediction.
What to submit: You can write the code in Python (preferred) or R. If you are using Python, you are not required to use numpy tensors.

- If you are using Python, submit jupyter notebook that contains code and proper comments.
- If you are using R, submit RMD file that contains code and proper comments.
- Please include your name in your jupyter notebook or RMD file.

1. Multilayer neural network [100 points]

A fully connected neural network with one input layer (two nodes), one hidden layer (two nodes), and one output layer (two nodes- first node for class 0 and the second node for class 1) is shown below. There are 8 weights in the network, all **the biases are zero**. The activation function in the hidden layer is 'ReLU', and the activation function in the output layer is 'softmax'. This network model works for a data set with 2 predictors and 2 classes.

Let y_i = output signal from i_{th} neuron in the output layer. Since there are two neurons in output layer, $i = 1, 2$. Let I_i be the input signal to neuron i in output layer. Then

$$y_1 = softmax(I_1) = \frac{exp(I_1)}{\sum_{i=1}^{2} exp(I_i)} \tag{1}$$

$$y_2 = softmax(I_2) = \frac{exp(I_2)}{\sum_{i=1}^{2} exp(I_i)} \tag{2}$$

Let H_i = output signal from i_{th} neuron in the hidden layer. Since there are two neurons in hidden layer, $i = 1, 2$. Let J_i be the input signal to neuron i in hidden layer. Then

$$H_1 = ReLU(J_1) = max(0, J_1) \tag{3}$$

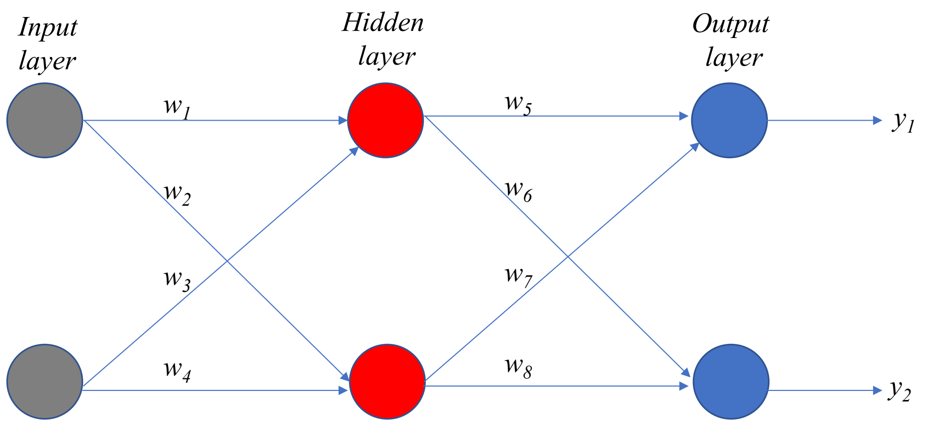

Figure 1: Network

Below, you are given two different sets of weights. These two sets of weights represent two sets of trained weights of the model using two different optimization algorithms. You are also given a test data set in 'testdata.txt' with class labels in 'test_class.txt'. The test data is already scaled. Your task is to figure out which of the two sets of weights represents a better trained model based on the prediction accuracy on the test data. With the knowledge of the weights and activation functions, write an **R or python code** to compute the probabilities of classifying each point in the test data as class 0 or class 1. Based on the prediction accuracy on the test data, conclude which set of weights is a better trained model. For this, you cannot use keras or any deep learning packages in R or Python. You have to code to compute output signals for the nodes in the model.

weights	**set1**	**set2**
w_1	1.3438	0.8061
w_2	-0.6225	0.2354
w_3	0.3509	-0.4092
w_4	-1.7072	-0.8999
w_5	-1.1398	-0.5538
w_6	0.3944	-0.1916
w_7	1.3882	0.0288
w_8	-0.8676	0.4918