CS 4375
ASSIGNMENT 2

Names of students in your group:
James Hooper
Hritik Panchasara

Number of free late days used: 0

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

Please list clearly all the sources/references that you have used in this assignment.

For preprocessing data split
https://scikit-learn.org/stable/
For data manipulation
https://pandas.pydata.org/
https://numpy.org/

REPORT

Summarizing the Results

To clarify the logs we must understand what each part means. We initially list out the Iterations, Learning Rate, Test Size (so .1 equals a 90/10 split for Train/Test), Seed 1 which will give us a seed value to determine the state of the train-test-split tool, Seed 2 which will give us a seed value to determine the random seed that will be fed to the creation of the initial weight values, the amount of nodes/neurons for the first hidden layer, and the finally the amount of nodes for the second hidden layer. The idea of having all these parameters changeable is to allow repeatable, comparable results. There are then three print outs that will denote the train & test accuracy for all three activation functions: Sigmoid, ReLu, and Tanh. The accuracy is determined through two different values. The first value is the percent correct which is a generalized idea of out of how many samples did the network calculate the correct answer. For example, if the boundaries for rounding in this case would be 0 = 0 <-> .499999 & 1 = .5 <-> 1. The percent is directly the amount correct divided by the amount tested times 100 for the percentage. The second value is the mean squared error found for the given pass through the trained neural network model. This value should be minimized from the training.

From the logs we gave/found the consensus is that the best parameters are: Iterations = 9000, Learning Rate = .001, Test Size = .1, Hidden Layer 1 = 6 Nodes, and Hidden Layer 2 = 4 nodes. For iterations, we settled at about 9000 with the value of 10000 yielding good results as well. Anything over than that seemed to overfit or not lead to any significant change in accuracy. A learning rate of .0001 seemed to do fine but again wasn't anything too special that would stop us from utilizing a Learning Rate of .001 as the optimal choice. Any learning rate above that would severely hurt the results given. Again, keep in mind the seeds are simply for different randomized train/test data & initial weight values. For the activation functions the best overall seemed to be split between the Tanh and Sigmoid function, with the Tanh being the best choice between the two on most cases tested. The overall last place goes to the ReLu function from what we found. The ReLu struggled the most with finding a optimum minimum MSE value. This could be due to the learning rate being finer tuned for the other two activation functions and could be solved with further testing. Other notes besides the ReLu struggling to even get close to a desirable MSE value, is that on occasion the Tanh function enabled a 100% test result. This odd case had the parameters 50000 iterations, .001 LR, .1 test size, 6 hidden layer 1 nodes, 4 hidden layer 2 nodes, and again two specific seeds. In this odd case the Sigmoid and Tanh performed extremely well, while the ReLu function suffered immensely.