**Neural Network Report**

1. In the given excel sheet named “results” we have provided the summary of the results we obtained with various parameters.
   1. It has 3 sections – one for each data set used.

Rows 3-11: summarize results for [census data set](https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data)

Rows 15-23: summarize results for [iris data set](https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data)

Rows 26-34: summarize results for [car evaluation data set](https://archive.ics.uci.edu/ml/machine-learning-databases/car/car.data)

1. For Census Data set:
   1. It was found that the data had missing values which were denoted by “?”.
   2. We tested 2 scenarios for this data set:
      1. One with removing rows containing missing values and
      2. Another by replacing missing values with the most frequent value of that attribute.
   3. For us, the scenario where rows containing missing values were removed, gave less error.
2. For iris data set:
   1. It was found that this data set had high correlation between 2 of the columns (correlation = 0.96) and did not have any missing values.
   2. We tested 2 scenarios for this data set:
      1. One without removing correlated attributes
      2. Second removing one of the correlated attributes
   3. No significant increase or decrease in error was observed in the above scenarios.
3. For car evaluation data set:
   1. This data set did not have missing values or any high correlated attributes. Thus, we did not consider any special scenarios for this data set.
4. Assumptions and Simplifications:
   1. We have ignored the bias neurons
   2. We have assumed that the last column will be the class label column
   3. We have assumed that the training and test datasets will have the same format i.e. same number and placement of columns
   4. We haven’t implement regularization, adaptive learning rate, or momentum factors.
5. Preprocessing:
   1. Replaced ‘?’ from the data with NaN values.
   2. Removed all the duplicate rows.
   3. Converted all the categorical valued attribute to numerical value using labelEncoder
   4. Scaled the data using MinMaxScaler
   5. Normalized the data
   6. Converted numpy array to dataframe
   7. Data is split into test and train. 25% test and 75% train
6. We also tried different number of hidden layers for each data set and have provided separate python files for each hidden layer configuration:
   1. Number of hidden layer = 4. Here we took 4 neurons in the first three hidden layers and 2 neurons in fourth hidden layer, this file is named as “<ML_4H.py>”.
   2. Number of hidden layers = 2. Here we took 4 neurons in the first layer and 2 neurons in second layer, this file is named as “<ML_2H.py>”.
   3. Number of hidden layers = 1. Here we took 4 neurons in the first layer, this file is named as “<ML_1H.py>”.
7. Best parameter for: The learning rate, No. of iterations and No. of Neurons in each layer has been chosen based on the experience.
8. Observations:
   1. Good results were obtained with 4 hidden layers in case of Census Income Data Set and Car Evaluation Data Set and with 1 hidden layer in case of Iris Data Set, as evident from the results.
   2. Among all the activation function, sigmoid function performed best. ReLU tends to blow up activation as there is no mechanism to constrain the output of the neuron, as "X" itself is the output. In case of tanh it is a rescaled of logistic sigmoid. So, with large iteration the magnitude is greater compared to sigmoid.