**Deep Learning Challenge Analysis**

**Overview**

The Purpose of this analysis was to create a  tool that can help, The nonprofit foundation Alphabet Soup, select viable applicants for funding, with the best chance of success in their ventures.

**Results**:

* The target variable(y) also referred to as the label, was the column “IS\_SUCCESSFUL” which identifies if the money was used effectively or not based on a binary classification model of 1=yes 0=no
* The features included the remaining columns:
  + APPLICATION\_TYPE—Alphabet Soup application type
  + AFFILIATION—Affiliated sector of industry
  + CLASSIFICATION—Government organization classification
  + USE\_CASE—Use case for funding
  + ORGANIZATION—Organization type
  + STATUS—Active status
  + INCOME\_AMT—Income classification
  + SPECIAL\_CONSIDERATIONS—Special considerations for application
  + ASK\_AMT—Funding amount requested
* The columns that were removed were “EIN and NAME” which were identification columns

The data frame values were then converted from categorical data to numeric data which created a new data frame, using 1 and 0 to indicates whether or not the record holds that attribute.

The images below show the original data frame and then the converted data frame using get\_dummies

A screenshot of a computer

Description automatically generated

* The model was then Scaled and split into test and training data.

To compile, train and evaluate the model 3 layers were used: First, second and output

In the First layer 75 neurons were selected, 35 in the second and 1 in the output layer

The first 2 layers used ReLU activation function as this is a good starting point for modelling positive non linear data. the final layer used Sigmoid Function and as the data can be normalized to a binary classification to produce a result which decides whether or not the model is accurate or not

A screenshot of a computer

Description automatically generated

The model was optimised using a tensor flow model and the aim was to achieve a target of 75%

The model was trained 3 times, firstly using 20 epochs, then 50 and finally 150. The epochs were increased to train the model to see if the increase would increase the accuracy.

The target performance was not achieved as accuracy for all 3 models achieved around 72%

A screenshot of a computer code

Description automatically generated

**Summary**:

Initially the objective was to build a model to help it select the applicants for funding however the model created was not able to reach the level of accuracy to be viable.

To achieve an accuracy score closer to the required goal of 75% I would maybe change the activation function, add more layer to test or increase the amount of epoch the model is trained on.