**Deep Learning Homework: Charity Funding Predictor**

**## Overview**

The non-profit foundation Alphabet Soup wants to create an algorithm to predict whether or not applicants for funding will be successful. My task was to create a binary classifier that is capable of predicting whether applicants will be successful if funded by Alphabet Soup.

From Alphabet Soup’s business team, I received a CSV containing more than 34,000 organizations that have received funding from Alphabet Soup over the years. Within this dataset are a number of columns that capture metadata about each organization, such as the following:

Here I will explain the methods and analysis of results.

**Results**

**Data Pre-processing**

\* What variable(s) are considered the target(s) for your model?

\* \*\*IS\_SUCCESSFUL\*\*—Was the money used effectively

\* What variable(s) are considered to be the features for your model?

\* \*\*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 consideration for application

\* \*\*ASK\_AMT\*\*—Funding amount requested

\* What variable(s) are neither targets nor features, and should be removed from the input data?

\* \*\*EIN\*\* and \*\*NAME\*\*—Identification columns

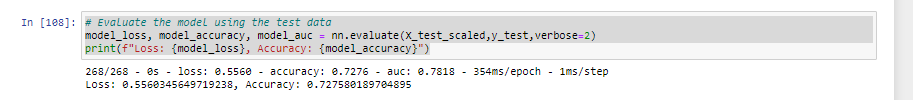
**Compiling, Training, and Evaluating the Model**

Q1) How many neurons, layers, and activation functions did you select for your neural network model, and why?

I initially chose the ReLu activation function for the hidden layers, and the sigmoid function for the output layer. I selected two hidden layers, which each contained five neurons.

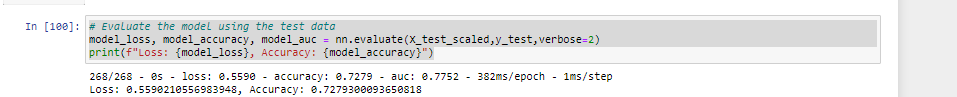
Q2) Were you able to achieve the target model performance?

No, the initial model returned an accuracy score of 73%, and any improvements did not bring it above this level.

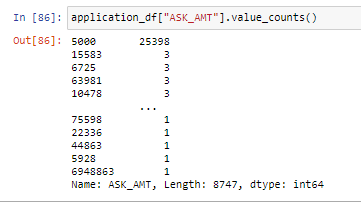


Q3) What steps did you take to try and increase model performance?

I first increased the number of binned application types from 9 to 12, which binned anything below 1000 value counts. This increased accuracy from 0.7279 to 0.7292 when training the model.



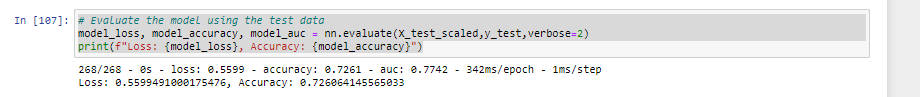
I was then intrigued in the high value count of the “5000” value for the ask amount – 25,398!



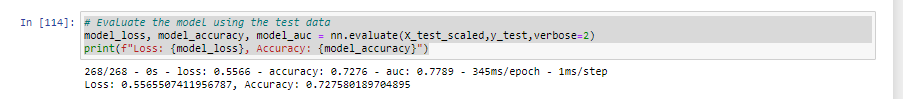
Analysis showed that the classification split was the same between the businesses asking for 5000 and the others, so I deemed this not to be an issue.



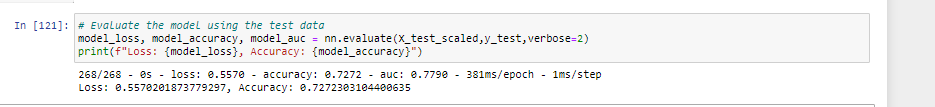
I then changed the activation function from ReLu to Tanh, which I had read was optimal for binary classification tasks. This decreased accuracy to 0.7261



I reverted to using the ReLu activation function and tried adding a third hidden layer to the model. This decreased accuracy to 0.7276.



I reverted back to using two hidden layers and increased the number of epochs from 100 to 200. This decreased the accuracy to 0.7272



**Summary**

The optimum score for my model was with the original specification and further binning of the “application type” feature to group all value counts below 1000 into the “other” category – this returned an accuracy score of 0.7292 (73%).