Neural Network Report

**Overview:**

The purpose of this analysis is to create and train a Neural Network that can take in venture applicants and make predictions on their success using attributes like Government Organization Classification, Income classification, Funding Amount Requested, Sector Affiliation, as well as several others.

**Results:**

Q – What variable is the target of the model?

A – IS\_SUCCESSFUL. In this analysis, we want to assess whether the Alphabet Soup-funded companies used their money effectively.

Q – What variables are the features for the model?

A – The CSV came with several features that will help us to assess the success of the Alphabet Soup-funded company, including 'APPLICATION\_TYPE', 'AFFILIATION', 'CLASSIFICATION', 'USE\_CASE', 'ORGANIZATION', 'STATUS', 'INCOME\_AMT', 'SPECIAL\_CONSIDERATIONS', and 'ASK\_AMT'. Because many of these features were string types, it was necessary to one-hot-encode many of these features to help the ML algorithm.

Q – What variable should be removed from the input data because they are neither targets nor features?

A – ‘EIN’ and ‘NAME’ were removed from the input data. They help to identify the Alphabet-Soup-funded companies, but they don’t provide any quantifiable differentiation between the companies, and therefore, are not very helpful when training the model.

Q – How many neurons, layers, and activation functions were selected for the optimized model?

A – A screen shot of a computer

Description automatically generated

Q – Does the model achieve target model performance (75% accuracy)?

A – Unfortunately, no. The highest model accuracy achieved was 73%.

Q – What steps did you take in your attempts to increase model performance?

A – “AlphabetSoupCharity-Optimized.ipynb” attempted to improve the model accuracy from “StarterCode.ipynb” in many ways, including

* Created more bins for the “Application\_Types” and “Classification” features.
* Only used the Standard Scaler on the Ask\_AMT feature.
* Used the Keras tuner to determine optimal activation functions for each layer, the number of neurons for each layer, and number of layers.
  + Landed on having 1 input layer, 3 hidden layers, and 1 output layer after using the Keras tuner, and then tried running additional keras tuner runs with the number of input layers set.

Summary:

The model was only able to achieve 73% accuracy in its predictions of success, even after the optimization steps taken above. However, further optimization steps could have taken. For example, we only one-hot-encoded two features, but there were several other features that could have been one-hot encoded such as ‘Use-Case’ and ‘Organization’. One-hot-encoding all of the features would have made the input dimensions for the neural network quite large, but doing this could have helped the deep learning model to focus more on other features, rather than placing the higher emphasis on Applcation Types and Classification.