**Alphabet Soup Neural Network Report**

**Overview**

The purpose of this report is to analyze the given data and to demonstrate how effectively it can be used to predict whether an applicant would be successful if funded by Alphabet Soup.

**Results**

**Data Preprocessing:**

* Target Variable: IS\_SUCCESSFUL
* Feature Variables: APPLICATION\_TYPE, AFFILIATION, CLASSIFICATION, USE\_CASE, ORGANIZATION, STATUS, INCOME\_AMT, SPECIAL\_CONSIDERATIONS, ASK\_AMT
* Removed Variables (neither features nor a target variable): EIN and NAME

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

* My initial model started with two hidden layers, 25 total nodes (10, 8, 6, 1) and “Relu” activation functions for the first three layers and a “sigmoid” function for the output layer. It seemed to me that this should be the minimum number of both nodes and layers to get a reasonable prediction and left plenty of room for optimization. I also concluded the model with the “sigmoid” activation function, because it would give us the required binary (“yes or no”) answer we required to make our prediction.
* Despite making a number of significant changes to the model, I was not able to achieve a significantly improved model to predict the success of an applicant. I started by increasing both the number of APPLICATION\_TYPES (Image 1) and CLASSIFICATIONS (Image 2) to give the model more data to work with instead of relegating so many data points to “Other.” Also, considering there were so many input dimensions (47), and it is good practice to have a multiples more nodes than there are initial inputs, I started with 200 nodes and gradually decreased town to the final done in the output layer (Image 3). Finally I increased the number of epochs, noticing after several tests that 50 was too few and 100 was more than necessary i.e. the improvement leveled out at around 80 epochs(Image 4). (NOTE: I also attempted a “LeakyRelu” activation function and increased the training data percentage from 75% to 80%, but both of those changes seemed to worsen the results and therefore didn’t end up in the final model.

**Summary**

Unfortunately, I would not recommend this model for predicting applicant success. A 72.5% success rate is insufficient for accurate predicting in a professional setting. Despite multiple attempts at optimizing the model a robust combination of data and deep learning could not be found. I would recommend finding more and unique data and perhaps attempt a supervised learning model instead of the neural network used here since it is great at binary categorizing and you are not only able to see it’s level of success, but also see how it’s making mistakes (False Positives vs False Negatives) and can judge the nature of its success that way instead.

Image 1

A screenshot of a computer program

Description automatically generated

Image 2

A screen shot of a computer program

Description automatically generated

Image 3

A screenshot of a computer program

Description automatically generated

Image 4

A screenshot of a computer

Description automatically generated