**Neural Network Model Report for Alphabet Soup**

## Overview of the Analysis

The purpose of this analysis is to develop a binary classification model to predict whether an Alphabet Soup-funded organization will be successful. The analysis involves data preprocessing, building and training a deep learning model, and optimizing the model to achieve the desired accuracy (75%).

## Results

**Data Preprocessing**

* Target Variable(s)
  + The target variable for the model is IS\_SUCCESSFUL, which indicates whether the funded organization was successful.
* Feature Variable(s)
  + The features for the model are the remaining columns that provide information about each organization, such as APPLICATION\_TYPE, AFFILIATION, CLASSIFICATION, USE\_CASE, ORGANIZATION, STATUS, INCOME\_AMT, SPECIAL\_CONSIDERATIONS, and ASK\_AMT.
* Removed Variable(s)
  + The EIN and NAME columns were removed from the input data because they are identifiers and do not contribute to the predictive power of the model.

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

* Neurons, Layers, and Activation Functions
  + Neurons: The original model consists of two hidden layers with 80 and 30 neurons. The optimized model consists of three hidden layers with 80, 30, and 20 neurons.
  + Layers: The models have two to three hidden layers and one output layer.
  + Activation Functions: The ReLU (Rectified Linear Unit) activation function is used for the hidden layers, and the sigmoid activation function is used for the output layer. During optimization, the eLU and tanh activation functions were experimented with.
  + Why? These choices were made to provide the model with sufficient capacity to learn complex patterns while avoiding overfitting.
* Model Performance
  + The initial model did not achieve the target performance of over 75% accuracy. After optimization, the model was able to achieve over 75% accuracy.
  + Original Model Results



* Steps Taken to Increase Model Performance
  + Added additional hidden layers and neurons.
    - Adding additional layers proved to increase the accuracy but not enough to meet 75% and the loss increased.



* + Experimented with different activation functions.
    - tanh – Decreased accuracy and loss ultimately determined to not be a good fit for this model.



* + - ELU – Increased accuracy and decreased loss ultimately determined to be a good fit for the model.



* + Adjusted the number of epochs during training.
    - Raising the number of epochs from 50 to 100 increased the accuracy but increased the loss to over 1.0.



* + **None of these steps by themselves met the goal of increasing the accuracy to 75% or higher so began experimenting with a combination of the above methods to reach our goal. Examples of these Model Optimization Experimentation can be found in the Additional-Optimization-Attempts folder**

## Summary

The overall results of the deep learning model indicate that while the initial model did not meet the target performance, subsequent optimization steps improved the accuracy. The best-performing model achieved an accuracy of over 75%. This method included utilizing the ReLU as the activation function in the first hidden layer and eLU as the activation function in the second hidden layer to achieve the desired accuracy and decrease the loss.

Final Model Results:



## Recommendation for a Different Model

Further optimization can be achieved by experimenting with different combinations of activation functions, epoch numbers, and layers. Additionally, ensemble methods like Random Forest or Gradient Boosting could be considered for better performance and interpretability.