**Neural Network Charity Analysis**

**Overview**

The purpose of this analysis was to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup. This was achieved by utilizing recently acquired knowledge about machine learning and neural networks. The neural network performed the analysis using Google Colab, the TensorFlow platform in available in Python, and the dataset provided containing more than 34,000 organizations that have already received funding from Alphabet Soup. Please see below for the analysis deliverables.

* Deliverable 1: Preprocessing Data for a Neural Network Model
* Deliverable 2: Compile, Train, and Evaluate the Model
* Deliverable 3: Optimize the Model
* Deliverable 4: A Written Report on the Neural Network Model

**Results**

Data Preprocessing

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

The variable considered the target is "IS\_SUCCESSFUL" since Alphabet Soup wants to know which organizations that receive funding will be successful or not.

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

The variables considered the features were: "APPLICATION\_TYPE", "AFFILIATION", "CLASSIFICATION", "USE\_CASE", "ORGANIZATION", "STATUS", "INCOME\_AMT", "SPECIAL\_CONSIDERATIONS", "ASK AMT"

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

The variables that are neither targets nor features are "EIN" & "NAME" and were removed accordingly from the input data.

Compiling, Training, and Evaluating the Model

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

I used a total of 145 neurons across 3 hidden layers with a total of 3 different activation functions: "relu", "tanh", and "sigmoid." The number of neurons chosen was based of the rule of thumb of having 2-3 times the number of neurons as the number of inputs. As for the number of hidden layers, I thought spacing the neurons out across multiple hidden layers would improve the predictive abilities of the neural network. Lastly, I chose the activation functions "relu", "tanh", and "sigmoid" because the point of the neural network model was to build a binary classifier, and these were in my opinion the best options to go with. However, the use of "tanh" was simply an attempt to achieve a higher accuracy towards the end.

* Were you able to achieve the target model performance?

I was not able to achieve the desired target model performance.

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

I added more neurons to each of the existing hidden layers, added an additional hidden layer with a different type of activation function ("tanh"), and added more epochs in an attempt to increase the predictive accuracy and reduce the losses of the neural network.

**Summary**

Overall, the neural network performs decently well but not for its intended purposes. The level of risk associated with the model doesn't provide sufficient confidence to use in predicting which investments into applicants will be successful. A recommendation of mine will be to investigate SVM machine learning models. This recommendation is based on the fact that the dataset doesn't seem too large that implementation of the SVM model will be quicker to build & train and quite possibly achieve a better binary classification accuracy than the utilized deep learning model created in this challenge.