**Introduction**

The nonprofit foundation Alphabet Soup wants a tool that can help it select the applicants for funding with the best chance of success in their ventures. With your knowledge of machine learning and neural networks, you’ll use the features in the provided dataset to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup.

**Findings**

**Data Preprocessing**

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

* The 'IS\_SUCCESSFUL' column from application\_df is the target variable, this is what we are trying to predict. This shows if the money was used effectively.

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

* The feature variables we used are:
  1. AFFILIATION—Affiliated sector of industry
  2. CLASSIFICATION—Government organization classification
  3. USE\_CASE—Use case for funding
  4. ORGANIZATION—Organization type
  5. STATUS—Active status
  6. INCOME\_AMT—Income classification
  7. SPECIAL\_CONSIDERATIONS—Special considerations for application
  8. ASK\_AMT—Funding amount requested

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

* Identification columns: The "EIN" and "NAME" columns are identification columns that typically provide unique identifiers for each organization. These columns usually have no direct impact on the target variable and can be dropped without affecting the model's accuracy.

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

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

* In my first neural network model, I used a two-layer architecture with a specific choice for the number of neurons, layers, and activation functions.
* Neural Network was applied on each model multiple layers, three in total. The number of features dictated the number of hidden nodes.
* A three-layer training model generated 477 parameters. The first attempt came close at 72% which was under the desired 75%.
* The model architecture with the chosen number of neurons, layers, and activation functions aimed to strike a balance between complexity and simplicity, allowing the model to learn and generalize well on the given classification task.

**Were you able to achieve the target model performance?**

* Yes. I was able to accede 75%

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

1. Increasing the number of neurons and epochs:

* By increasing the number of neurons in a layer, the model becomes more expressive and can capture complex patterns in the data. This allows for better representation of the underlying relationships between the features and the target variable, potentially leading to higher accuracy.
* Increasing the number of epochs gives the model more opportunities to learn from the data and adjust the weights. It allows the model to refine its predictions and find better parameter values, which can lead to improved accuracy. However, it's important to find a balance as increasing epochs excessively can lead to overfitting.

1. Adding more layers to the model:

* Adding more layers can provide the model with additional capacity to capture and represent intricate relationships within the data. Each layer can learn different levels of abstraction, enabling the model to extract more meaningful features and potentially improving accuracy. Deep models with multiple layers have the ability to learn hierarchical representations of the data, which can be advantageous for complex problems.

1. Using a different activation function

* Introducing a different activation function, such as sigmoid, can affect how the model interprets and transforms the inputs. Different activation functions have different properties and can capture different types of non-linearities. By using tanh, it introduces a different non-linearity that may better suit the problem at hand, potentially leading to increased accuracy.