Alphabet Soup Charity Fundraising Analysis

1. Objective:

The Alphabet Soup Charity has helped numerous applicants with funding projects and ventures. To date, over 34,000 organizations have received funding through Alphabet Soup, with some organizations utilizing the funding efficiently and others missing the mark. With data compiled from previous funding applications and the subsequent success and utilization of money for the project, we will be building a machine learning model to help analyze if a new application will be able to use funding effectively if granted by the organization.

2. Preprocessing of the Data:

The target variable for our model was the 'IS_SUCCESSFUL' variable which determines if the money allocated to the organizations project was used effectively. In order to analyze this variable, we first had to preprocess the data provided by Alphabet Soup's business team. This included removing several identifying columns, such as 'NAME' and 'EIN' as well as several columns that will likely not contribute towards the target and should be removed as features such as 'SPECIAL_CONSIDERATIONS' and 'STATUS' (performed to improve optimization).

After these columns were removed, we noticed that several features had too many unique values, therefore the 'CLASSIFICATION' and 'APPLICATION_TYPE' column data was organized into appropriate bins to manage the data.
The data was then converted to numeric data with pd.get_dummies and split into the target array and features. We then split the data into a training and testing dataset and standardize the data with StandardScalar.

3. Building and Training the Model:

A model was initially created using two hidden layers, with 50 and 30 neurons respectively and using a relu activation function. This model was compiled and trained over 100 epochs and resulted in an accuracy of 73.8%.
In order to optimize the performance of the model, several changes with attempted to include:

- Removing the 'SPECIAL_CONSIDERATIONS' and 'STATUS' as they most likely did not affect the target variable (improved model)

- Increasing the number of bins during preprocessing of the data. (improved model)
- Increasing the number of hidden layers and number of neurons for each layer (improved model)
- Changing the activation function to Sigmoid (improved model)
- Increasing the number of epochs for the model to train (did not affect model)
- Changing the training and testing data split proportion (did not affect model)

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 80)                3920

 dense_1 (Dense)             (None, 65)                5265

 dense_2 (Dense)             (None, 48)                3168

 dense_3 (Dense)             (None, 38)                1862

 dense_4 (Dense)             (None, 25)                975

 dense_5 (Dense)             (None, 10)                260

 dense_6 (Dense)             (None, 1)                 11

=================================================================
Total params: 15,461
Trainable params: 15,461
Non-trainable params: 0
```

These parameters optimized the model to 74.1%, just shy of the 75% goal.

3. Summary:

While the model performed well and almost accomplished the 75% threshold, further modeling can include optimization toolkits to better analyze the data and use ideal factors to train the model. In addition, more insight into the data features can help identify which factors have the greatest impact on our target variable and can be used to train the model.