Overview

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.

Results

Data Preprocessing

In the data preprocessing phase, the target variable for this model is defined as 'IS_SUCCESSFUL.' As for the features used in the model, all columns from the dataset, except 'EIN' and 'IS_SUCCESSFUL,' have been incorporated. These features encompass 'APPLICATION_TYPE,' 'AFFILIATION,' 'CLASSIFICATION,' 'USE_CASE,' 'ORGANIZATION,' 'STATUS,' 'INCOME_AMT,' and 'SPECIAL_CONSIDERATIONS.' It's important to note that 'EIN' and 'NAME' were excluded from the input data as they do not serve as either targets or features.

For my neural network model, I selected two hidden layers. The first hidden layer has 80 neurons, and the second hidden layer has 30 neurons. Both hidden layers use the ReLU (Rectified Linear Unit) activation function, which is commonly used in neural networks for its ability to handle non-linearity effectively.

In terms of the output layer, I have one neuron with a sigmoid activation function, which is suitable for binary classification problems like the one at hand where 'IS_SUCCESSFUL' is either 0 or 1.

In the first attempt the model achieved an accuracy of 60% which is below the desired accuracy which was 75%.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 80)                4080

 dense_1 (Dense)             (None, 30)                2430

 dense_2 (Dense)             (None, 1)                 31

=================================================================
Total params: 6541 (25.55 KB)
Trainable params: 6541 (25.55 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In an effort to enhance the model's performance, several adjustments were made to the neural network architecture:

Increased Hidden Layers and Neurons: The model now incorporates four hidden layers with varying numbers of neurons. The first hidden layer has 200 neurons, the second has 100, the third has 50, and the fourth has 30. This increase in depth and width of the network allows for the capture of more complex patterns in the data.

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_21 (Dense)            (None, 200)               8200

 dense_22 (Dense)            (None, 100)               20100

 dense_23 (Dense)            (None, 50)                5050

 dense_24 (Dense)            (None, 30)                1530

 dense_25 (Dense)            (None, 1)                 31

=================================================================
Total params: 34911 (136.37 KB)
Trainable params: 34911 (136.37 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Activation Function: The ReLU (Rectified Linear Unit) activation function, known for its ability to introduce non-linearity and prevent the vanishing gradient problem, continues to be employed in all hidden layers. This choice promotes better convergence and feature representation.

Output Layer: The output layer retains one neuron with a sigmoid activation function, suitable for binary classification tasks like the one in question, where 'IS_SUCCESSFUL' can take on values of 0 or 1.

These adjustments were made with the intention of constructing a more expressive model that can better learn from the data. It's important to note that achieving improved model performance often involves fine-tuning hyperparameters, data preprocessing, and rigorous evaluation, which should be carried out to assess the actual impact of these architectural changes.

Unfortunately, the second model was also not able to reach the accuracy that was initially sought, which was 75%.