Neural Network Model Report

Overview of the Analysis

The purpose of this analysis was to develop a deep learning model to predict the success of funding applications submitted to Alphabet Soup, a non-profit organization. The goal was to design and optimize a neural network model to classify whether an application would be successful based on several input features. The target performance goal was to achieve an accuracy of **75%** or higher.

Results

Data Preprocessing

- **Target Variable:**
  - `IS_SUCCESSFUL` – This variable indicates whether the funding application was successful (1) or not (0).
- **Feature Variables:**
  - All other columns in the dataset after removing non-beneficial ID columns.
  - Features include categorical and numerical data such as `APPLICATION_TYPE`, `CLASSIFICATION`, `ASK_AMT`, and `USE_CASE`.
- **Removed Variables:**
  - `EIN` and `NAME` were removed because they are identifiers and provide no predictive value.

Compiling, Training, and Evaluating the Model

Three attempts were made to improve the performance of the neural network model:

1. **First Attempt:**
   - **Layers:** 2 hidden layers
   - **Neurons:** 64 in the first layer, 32 in the second layer

    - **Activation Functions:** ReLU for hidden layers, sigmoid for output layer
    - **Epochs:** 25
    - **Batch Size:** 32
    - **Result:** Accuracy = **72.1%**

2. **Second Attempt:**

    - **Layers:** 3 hidden layers
    - **Neurons:** 64 → 32 → 16
    - **Activation Functions:** ReLU for hidden layers, sigmoid for output layer
    - **Epochs:** 50
    - **Batch Size:** 32
    - **Result:** Accuracy = **72.6%**
    - **Reason for Adjustment:** Adding a third hidden layer was intended to increase model complexity and pattern recognition.

3. **Third Attempt:**

    - **Layers:** 3 hidden layers
    - **Neurons:** 64 → 32 → 24
    - **Activation Functions:** ReLU for hidden layers, sigmoid for output layer
    - **Epochs:** 75
    - **Batch Size:** 32
    - **Improvements:**
        - Added **Batch Normalization** after each hidden layer to stabilize learning.
        - Introduced **Dropout** (0.2) to prevent overfitting.
        - Increased neurons in the third hidden layer to improve complexity.
    - **Result:** Accuracy = **72.9%**

Were the Goals Met?

- The target accuracy of **75%** was **not met**, but incremental improvements were made with each adjustment.
- The highest accuracy achieved was **72.9%** after introducing batch normalization, dropout, and increasing the third hidden layer size.

Summary

The deep learning model showed consistent improvement through a series of adjustments, including increasing the number of hidden layers, tuning the number of neurons, and applying batch normalization and dropout to reduce overfitting. However, the target accuracy of **75%** was not reached, suggesting that the model may be limited by the dataset itself or the architecture used.

Recommendation:

To further improve performance, a different model could be tested:
1. **Random Forest or Gradient Boosting:** Tree-based models often perform well on structured data with categorical features.
2. **Hyperparameter Tuning:** Adjust the learning rate, batch size, and number of epochs using grid search or random search.
3. **Feature Engineering:** Additional feature selection and creation might help expose hidden patterns.

While the neural network showed promise, a more traditional classification algorithm or ensemble model might yield better results for this specific problem.