**Deep Learning Challenge: Alphabet Soup Charity Funding Success Prediction**

**Overview of the Analysis**

The goal of this analysis is to develop a binary classification model predicting the success of organizations funded by Alphabet Soup. The model uses several features, such as application type, organization classification, income, funding amount requested, etc., to predict whether an organization will be successful (IS\_SUCCESSFUL). The objective is to apply machine learning techniques to fine-tune the model and enhance its performance.

**Results**

**Data Preprocessing:**

* **Target Variable(s):** The target variable for this model is IS\_SUCCESSFUL, which indicates whether the funding was used effectively (1 for success, 0 for failure).
* **Feature Variables:** We dropped EIN and NAME columns for being identification columns and used the remaining relevant features for prediction:
  + APPLICATION\_TYPE
  + AFFILIATION
  + CLASSIFICATION
  + USE\_CASE
  + ORGANIZATION
  + STATUS
  + INCOME\_AMT
  + SPECIAL\_CONSIDERATIONS
  + ASK\_AMT
* **Unique Values in Columns:** Categorical variables with rare categories were grouped under the value Other, reducing complexity.
* **Data Encoding:** Categorical variables were encoded using pd.get\_dummies() to make the data suitable for model training.
* **Data Splitting:** The data was split into features (X) and target (y), and further divided into training and testing datasets using train\_test\_split().
* **Scaling Data:** StandardScaler was used to normalize the features.

**Model Architecture and Evaluation**

**First Attempt:**

* **Initial Model Configuration:** We began with a simple architecture consisting of two hidden layers with ReLU activation functions.
* A screenshot of a computer

  AI-generated content may be incorrect.
* **Evaluation Result:**
  + Loss: 0.5583
  + Accuracy: 0.7191
  + The initial model achieved an accuracy of 71.91%, below the target of 75%.

**Second Attempt:**

* **Model Configuration:** Increased the number of units in each layer.

A screenshot of a computer

AI-generated content may be incorrect.

* **Evaluation Result:**
  + Loss: 0.5586
  + Accuracy: 0.7238
  + Accuracy increased slightly to 72.38%.

**Third Attempt:**

* **Model Configuration:** Increased units further and changed activation function to tanh to experiment with improved performance.

A screenshot of a computer

AI-generated content may be incorrect.

* **Evaluation Result:**
  + Loss: 0.5608
  + Accuracy: 0.7245
  + The model's performance improved slightly to 72.45%, but we sought more optimization.

**Fourth Attempt:**

* **Model Configuration:** Added a third hidden layer with increased units.

A screenshot of a computer

AI-generated content may be incorrect.

* **Evaluation Result:**
  + Loss: 0.5750
  + Accuracy: 0.7278
  + This configuration resulted in 72.78% accuracy, an improvement.

**Keras Tuner Experiment:**

* **Keras Tuner Model:** Despite extensive hyperparameter tuning with Keras Tuner, the best model achieved only 72% accuracy. This suggested that further manual tuning was necessary. The result is saved in keras\_tuner folder.

**Optimization Attempts**

**Optimization Attempt 1:**

* **Model Configuration:** After reviewing the feature set and incorporating the NAME column by using its count as a feature, we achieved significant improvement in performance.
* A screenshot of a computer

  AI-generated content may be incorrect.
* **Evaluation Result:**
  + Loss: 0.4930
  + Accuracy: 0.7552
  + We achieved an accuracy of 75.52%, exceeding the target of 75%.

**Optimization Attempt 2:**

* **Model Configuration:** We experimented by increasing the threshold value for the NAME count feature to 100, but this led to a slight reduction in accuracy.
  + **Evaluation Result:**
    - Loss: 0.5057
    - Accuracy: 0.7488
  + We reverted the threshold value to 50, as it performed better.

**Optimization Attempt 3:**

* **Model Configuration:** With the NAME count threshold set at 50, we increased the number of units in the hidden layers to see if further optimization would yield better results.

A screenshot of a computer

AI-generated content may be incorrect.

* **Evaluation Result:**
  + Loss: 0.4995
  + Accuracy: 0.7554
  + The model's performance remained consistent at 75.54%, confirming that the NAME count feature contributed positively.

**Summary and Recommendations**

The final optimized model achieved an accuracy of **75.54%**, surpassing the target. The inclusion of the NAME count feature significantly contributed to the improved performance. While several configurations of neural network architectures were tested, the best results came from a model with two hidden layers and increased units, combined with a refined NAME feature.

Further optimization could focus on:

* Trying additional feature engineering or data transformations.
* Experiment with **alternative models** for comparison. Random Forest is a strong alternative to deep learning for this problem, offering better interpretability and strong performance on structured data. It should be considered for further analysis, along with XGBoost, to determine the best model for predicting Alphabet Soup’s funding success.
* Hyperparameter tuning for further improvements using Keras Tuner.

The deep learning model successfully predicts the success of Alphabet Soup-funded organizations and meets the desired accuracy goal.