Alphabet Soup Charity Deep Learning Model Report

Overview of the Analysis

The purpose of this analysis is to design and evaluate a deep learning model to predict the success of funding applications submitted to the Alphabet Soup foundation. The model aims to classify whether an application is successful.The analysis involves preprocessing data, designing a neural network, and optimizing its performance through multiple attempts, with the goal of improving the predictive accuracy.

Results

Data Preprocessing

- **Target Variable**:
  - IS_SUCCESSFUL: (binary classification: 1 for success, 0 for failure).
- **Feature Variables**:
  - Categorical variables:
    - APPLICATION_TYPE
    - CLASSIFICATION
    - One-hot encoded categorical columns derived from the above variables.
  - Numerical variables:
    - Other numerical columns from the dataset after scaling.
- **Removed Variables**:
  - EIN and NAME: Do not provide meaningful information for predicting the success of an application.

Compiling, Training, and Evaluating the Model

- **Model**
  - **Neurons**: The model used 100 neurons in the first hidden layer and 50 neurons in the second hidden layer.
  - **Layers**:
    - Input layer explicitly defined with the number of features.
    - Two hidden layers with activation functions (relu and tanh in separate attempts).
    - An output layer with 1 neuron and a sigmoid activation function to provide probabilities.
  - **Activation Functions**:
    - Hidden layers: relu and tanh were used to explore their effects.
    - Output layer: sigmoid for binary classification.
- **Performance**:
  - Final evaluation results:
    - **Loss**: 0.5708
    - **Accuracy**: 72.77%
  - The model did not achieve the target accuracy of 75%.

Optimization Attempts

1. **Increasing Neurons**:
    - Added 100 neurons in the first hidden layer and 50 neurons in the second hidden layer to increase the model's capacity.
    - Result: Slight improvement in accuracy but still below the target.
2. **Adjusting Activation Functions**:
    - Experimented with relu and tanh activations in hidden layers to explore non-linear relationships.
    - Added a LeakyReLU activation for better handling of negative input values.
    - Result: Marginal changes in accuracy.
3. **Binning Rare Categories**:
    - Adjusted binning thresholds for APPLICATION_TYPE and CLASSIFICATION columns to reduce noise caused by rare categories.
    - Result: Improved generalization, but the target accuracy was not met.
4. **Longer Training**:
    - Increased the number of epochs to 100 in some runs to improve learning.
    - Result: Limited improvement, but overfitting began to occur on the validation set.

Summary

- **Overall Results**:
    - The deep learning model achieved an accuracy of **72.77%**, falling short of the target accuracy of 75%.
    - Preprocessing efforts, including scaling, one-hot encoding, and binning, contributed to a cleaner dataset for training.
    - Optimization attempts, such as increasing neurons, experimenting with activation functions, and adjusting training parameters caused incremental improvements.
- **Recommendations**:
    - **Alternative Model**:
        - A tree-based model
            - Tree-based models are better at capturing non-linear relationships in tabular data, especially when categorical variables dominate.
            - They require less tuning of architecture compared to deep learning models and are more interpretable for feature importance.