Analysis Report: Alphabet Soup Funding Prediction

Purpose of the Analysis

The nonprofit foundation Alphabet Soup seeks to develop a machine learning tool to predict the success of applicants for funding. Using a binary classification approach, this tool identifies candidates with the best potential for success. The analysis involves creating and comparing two models:

Manual Model: Manually configured parameters.

Optimized Model: Created using Keras Tuner to automate hyperparameter selection.

This report evaluates the models' performance, describes data preprocessing, and explores alternative approaches to solving this problem.

Data Preprocessing

Target Variable

IS_SUCCESSFUL: The binary variable indicating whether an applicant was successful (1) or not (0).

Feature Variables

The following variables were selected as features for the model:

APPLICATION_TYPE
AFFILIATION
CLASSIFICATION
USE_CASE
ORGANIZATION
STATUS
INCOME_AMT
SPECIAL_CONSIDERATIONS
ASK_AMT

Variables Removed

EIN: A unique identifier that does not contribute to the prediction.

NAME: A non-informative identifier irrelevant to the prediction task.

Model Development and Performance

Compiling, Training, and Evaluating the Models

Manual Model

Architecture:

Loss: 0.5558692812919617

Accuracy: 72.65%

Parameters were manually tuned to achieve satisfactory performance, though some limitations remain.

Optimized Model

Architecture:

Activation Function: tanh

First Layer Units: 9

Number of Layers: 3

Units per Layer: [3, 9, 1]

Loss: 0.5521643757820129

Accuracy: 73.50%

Reason for Parameters: Keras Tuner was used to optimize hyperparameters such as the number of layers, units per layer, and activation function to achieve the best validation accuracy.

Steps Taken to Improve Performance

Adjusted the number of layers and units, focusing on minimizing overfitting and improving validation accuracy.

Set the maximum number of epochs to 20 during tuning.

Changed the activation function to tanh for better performance with the given dataset.

Results and Discussion

Answering Key Questions

1. What variable(s) are the target(s) for your model?

IS_SUCCESSFUL

2. What variable(s) are the features for your model?

APPLICATION_TYPE, AFFILIATION, CLASSIFICATION, USE_CASE, ORGANIZATION, STATUS, INCOME_AMT, SPECIAL_CONSIDERATIONS, ASK_AMT

3. What variable(s) should be removed from the input data because they are neither targets nor features?

EIN and NAME

4. How many neurons, layers, and activation functions did you select for your neural network model, and why?

Parameters for the optimized model were:

Activation Function: Tanh

First Layer Units: 9

Number of Layers: 3

Units per Layer: [3, 9, 1]

These values were chosen using Keras Tuner, which identified combinations yielding the best validation accuracy.

5. Were you able to achieve the target model performance?

The optimized model achieved a validation accuracy of 73.50%, which is a significant improvement over the manual model's accuracy of 72.65%.

6. What steps did you take in your attempts to increase model performance?

Used Keras Tuner to automate the optimization of hyperparameters.

Experimented with different activation functions and reduced the number of layers to prevent overfitting.

Model Summary

While the optimized model improved slightly upon the manual model in terms of accuracy (73.50% compared to 72.65%), the overall performance remains suboptimal. Neither model reached a sufficiently high level of accuracy (e.g., 75% or higher), which raises concerns about the model's ability to make reliable predictions.

The relatively low accuracy suggests that the data might not be well-suited to the current deep learning approach or that additional preprocessing, feature engineering, or alternative modeling techniques may be required.