Alphabet Soup Charity — Deep Learning Model Report

Overview of the Analysis

The goal of this analysis is to develop a binary classification model using deep learning techniques that can help the nonprofit foundation Alphabet Soup predict which applicants are more likely to be successful if funded. A neural network model was developed using TensorFlow and Keras, trained on over 34,000 past application records, and optimized for predictive accuracy.

Results

Data Preprocessing

- Target Variable: IS_SUCCESSFUL — A binary variable indicating if an applicant was successful (1) or not (0).

- Feature Variables: All other variables except EIN and NAME.

- Dropped Columns: EIN, NAME — Removed due to being non-predictive identifiers.

- Encoding: Used pd.get_dummies() for one-hot encoding of categorical variables.

- Rare Categories: Consolidated low-frequency application types and classifications into 'Other'.

- Data Splitting and Scaling: Used train_test_split (75/25) and StandardScaler for feature scaling.

Compiling, Training, and Evaluating the Model

- Initial Model Architecture:

- - 1st Hidden Layer: 80 neurons, ReLU activation
- - 2nd Hidden Layer: 30 neurons, ReLU activation
- - Output Layer: 1 neuron, sigmoid activation
- - Loss Function: Binary Crossentropy

- - Optimizer: Adam

- Initial Accuracy: ~72.8% on the test set

Optimizing the Model

- Tuning Method: Used Keras Tuner with Hyperband to search for optimal hyperparameters.

- Optimization Techniques Used:

1. 1. Tuned number of neurons in both hidden layers
2. 2. Tuned activation functions (relu, tanh)
3. 3. Added Dropout layer (rates between 0.1 and 0.5)
4. 4. Tuned learning rate (in some tests)
5. 5. Tuned number of epochs using Hyperband

- Best Performing Model: ~72.75%–73% accuracy. Although below the 75% goal, multiple optimization strategies were implemented.

Summary

The final model reached ~73% accuracy, showing modest improvement from the baseline. Multiple architecture and tuning attempts were applied using Keras Tuner. Although it fell short of the 75% target, further tuning and model alternatives could help.

Recommendation: Try Alternative Models

Given the neural network's limited performance on this structured dataset, consider trying models such as:

- - Random Forest Classifier
- - Gradient Boosted Trees (XGBoost)

These are often better suited for tabular data and may require less manual tuning.