Neural Network Performance Analysis

Overview of the Analysis

The purpose of this analysis is to build and evaluate a deep learning model to predict the success of applications using a neural network. By analyzing key features and target variables, the model aims to identify patterns in the data that influence the likelihood of success. This analysis includes data preprocessing, model evaluation, and recommendations for improvement.

Results

Data Preprocessing

Target Variable(s):

-IS\_SUCCESSFUL is the target variable, indicating whether an application was successful.

-Feature Variable(s):

-Key features include:

-APPLICATION\_TYPE

-AFFILIATION

-CLASSIFICATION

-USE\_CASE

-ORGANIZATION

-STATUS

-INCOME\_AMT

-SPECIAL\_CONSIDERATIONS

-ASK\_AMT

Removed Variable(s):

EIN and NAME were removed as they are identifiers with no predictive value.

Compiling, Training, and Evaluating the Model

Neurons, Layers, and Activation Functions:

Input Layer: Number of features = 9 (based on the dataset after preprocessing).

Hidden Layers:

L-ayer 1: 12 neurons, ReLU activation.

-Layer 2: 8 neurons, Tanh activation.

-Output Layer: 1 neuron, Sigmoid activation (suitable for binary classification).

Rationale: ReLU was chosen for its efficiency in avoiding vanishing gradients. Tanh was included to experiment with activation function variation.

Model Performance:

-Accuracy: 72.76%

-Loss: 0.5547

-The model did not meet the desired threshold for high performance (e.g., >75% accuracy).

Steps to Improve Performance:

Tested different numbers of neurons in each hidden layer.

-Experimented with additional hidden layers to increase model capacity.

-Attempted different activation functions (e.g., Tanh vs. ReLU) to improve non-linearity.

-Adjusted the number of training epochs and batch size.

Summary

The deep learning model achieved a moderate accuracy of 72.79%, which provides some predictive power but falls short of optimal performance for this classification task. The loss value indicates room for improvement in the model's optimization.

Recommendation

For better performance, consider using a different model such as a Random Forest Classifier:

Why Random Forest?

Handles categorical and numerical features effectively without extensive preprocessing.

-Provides feature importance insights, which can guide further feature selection and engineering.

-Less prone to overfitting compared to individual decision trees, offering better generalization.

-By implementing a Random Forest, you can achieve higher interpretability and potentially better accuracy while simplifying the workflow for handling categorical variables and feature interactions.