Performance Analysis of Deep Learning Model for Alphabet Soup

Results

Data Preprocessing

- **Target Variable:** The target variable for the model is `IS_SUCCESSFUL`, which indicates whether a funding application was successful.
- **Feature Variables:** The features include `APPLICATION_TYPE`, `AFFILIATION`, `CLASSIFICATION`, `USE_CASE`, `ORGANIZATION`, `STATUS`, `INCOME_AMT`, `SPECIAL_CONSIDERATIONS`, and `ASK_AMT`.
- **Removed Variables:** The `EIN` and `NAME` columns were removed from the dataset as they are unique identifiers and do not contribute to the prediction. I included the `NAME` column for my improved models and replaced any name with fewer than 100 counts with “other” as in the `CLASSIFICATION` and `APPLICATION_TYPE` columns.
- Before data cleanup:

```
# Determine the number of unique values in each column.
for col in df.columns:
    if df[col].dtype == 'object':  # Check if column is string-based
        print(col)
        print(df[col].nunique())
```

```
NAME
19568
APPLICATION_TYPE
17
AFFILIATION
6
CLASSIFICATION
71
USE_CASE
5
ORGANIZATION
4
INCOME_AMT
9
SPECIAL_CONSIDERATIONS
2
```

- After data cleanup:

```
# Determine the number of unique values in each column.
for col in df.columns:
    if df[col].dtype == 'object':  # Check if column is string-based
        print(col)
        print(df[col].nunique())
```

```
NAME
31
APPLICATION_TYPE
9
AFFILIATION
6
CLASSIFICATION
12
USE_CASE
5
ORGANIZATION
4
INCOME_AMT
9
SPECIAL_CONSIDERATIONS
2
```

Neural Network Architecture:

In my initial model, I used an input layer with 5 neurons and a ReLU activation function and an output layer with a sigmoid function. The model usually reached a loss of ~0.6 quickly and would not improve more, and based on the classification report, the model was guessing everyone was either approved or not approved. For my first improvement, I changed the activation function to a Leaky ReLU function to prevent nodes from dying off. This meant the model did not quickly land on guessing the same thing for every situation and instead explored other patterns in the data. Adding another hidden layer with 3 neurons seemed to give the model more consistent accuracy across epochs, but after examining the confusion matrix it was obvious that the model was guessing the same thing for almost every datapoint. My third improvement was to remove the second hidden layer and train the Leaky ReLU model for 25 epochs instead of 10. This led to the most diverse and varied guesses of any model and a 65% overall accuracy. For groups that successfully obtained funding, the models accuracy was closer to 50%.

Summary

Due to the size of the dataset and the number of features, a simpler model like a tree or gradient boosted one might perform better. They are less sensitive to the parameters and would be able to give us more information about which columns of data are the most informative.