Deep Learning Challenge Analysis

Overview

The purpose of this data analysis is to create an algorithm that can predict if applicants will use Alphabet Soup's funding successfully. The algorithm is trained on a data set that contains information on over 34,000 organizations.

Results

Data Preprocessing:

- Target variable = 'IS_SUCCESSFUL'
- Feature variables:
  - 'APPLICATION_TYPE'
  - 'AFFILIATION'
  - 'CLASSIFICATION'
  - 'USE_CASE'
  - 'ORGANIZATION'
  - 'STATUS'
  - 'INCOME_AMT'
  - 'SPECIAL_CONSIDERATIONS'
  - 'ASK_AMOUNT'
- Removed variables:
  - 'EIN'
  - 'NAME'

```
# Drop the non-beneficial ID columns, 'EIN' and 'NAME'.
application_df = application_df.drop(columns=['EIN','NAME'])
```

```
# Determine the number of unique values in each column.
for x in application_df.columns:
  print(x, len(application_df[x].unique()))
```

```
APPLICATION_TYPE 17
AFFILIATION 6
CLASSIFICATION 71
USE_CASE 5
ORGANIZATION 4
STATUS 2
INCOME_AMT 9
SPECIAL_CONSIDERATIONS 2
ASK_AMT 8747
IS_SUCCESSFUL 2
```

Compiling, Training, and Evaluating the Model:

- First hidden layer: 7 neurons, activation = ‘relu’
- Second hidden layer : 14 neurons, activation = ‘relu’
- Output layer: 1 neuron, activation = ‘sigmoid’

```
# Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.
input_features = len(X_train_scaled[0])
hidden_layer1=7
hidden_layer2=14

nn = tf.keras.models.Sequential()

# First hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_layer1, input_dim=input_features, activation='relu'))

# Second hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_layer2, activation='relu'))

# Output layer
nn.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
```

- Unable to achieve target of 75% accuracy for model performance after 3 attempts
- My first attempt yielded 74%. Then I doubled the amount of epochs but lost accuracy. Then I went back to the original amount of epochs, but doubled the amount of neurons in each layer. That did not increase model performance either.

Summary

The best results from my deep learning model yielded a 74% accuracy score. Improving the model might include more hidden layers, neurons, and/or activation functions. One might consider creating a heat map of the features to see which columns can be dropped so that we are only training with important data.