Module 21 Challenge
Deep Learning

Analysis Report for the Alphabet Soup Charity

Overview:

The purpose of this project is to develop a model which will help select the applicants for funding that will prove to be the most successful in reaching their goals by using Machine Learning and Neural Networks.

Data Preprocessing:

- **EIN** and **NAME**—Identification columns
- **APPLICATION_TYPE**—Alphabet Soup application type
- **AFFILIATION**—Affiliated sector of industry
- **CLASSIFICATION**—Government organization classification
- **USE_CASE**—Use case for funding
- **ORGANIZATION**—Organization type
- **STATUS**—Active status
- **INCOME_AMT**—Income classification
- **SPECIAL_CONSIDERATIONS**—Special considerations for application
- **ASK_AMT**—Funding amount requested
- **IS_SUCCESSFUL**—Was the money used effectively

To initiate the data processing, we begin with the removal of any unnecessary information to optimize our findings. The EIN and NAME columns were dropped initially. However, NAME was re-instated for the following attempts for the purpose of accurately binning. The data was then split into a training and testing set. “IS_SUCCESSFUL” was assigned as our Target Variable while APPLICATION_TYPE and CLASSIFICATION were used as our Features.

Compiling, Training and Evaluating the Model:

Our initial model was comprised of 3 Hidden Layers which resulted with an accuracy of 72%

```
# Define the model - deep neural net, i.e., the number of input features and
hidden nodes for each layer.
number_input_features = len( X_train_scaled[0])
hidden_nodes_layer1=7
hidden_nodes_layer2=14
hidden_nodes_layer3=21
nn = tf.keras.models.Sequential()

nn = tf.keras.models.Sequential()

# Evaluate the model using the test data
model_loss, model_accuracy = nn.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
268/268 - 1s - loss: 0.5530 - accuracy: 0.7266 - 909ms/epoch - 3ms/step
Loss: 0.5529555082321167, Accuracy: 0.7266472578048706
```

In order to improve our accuracy for our second attempt we only removed the “EIN” column

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

Which increased our Trainable Parameters to 974 vs the 757 of our first attempt.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 7)                 525

 dense_1 (Dense)             (None, 14)                112

 dense_2 (Dense)             (None, 21)                315

 dense_3 (Dense)             (None, 1)                 22

=================================================================
Total params: 974 (3.80 KB)
Trainable params: 974 (3.80 KB)
```

For our final attempt, we kept the “EIN” field removed and increased the number of hidden layers to 5:

```
# Define the model - deep neural net, i.e., the number of input features and
hidden nodes for each layer.
number_input_features = len( X_train_scaled[0])
hidden_nodes_layer1=7
hidden_nodes_layer2=14
hidden_nodes_layer3=21
hidden_nodes_layer4=21
hidden_nodes_layer5=28



# First hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer1,
input_dim=number_input_features, activation='relu'))

# Second hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer2, activation='relu'))

# Third hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer3, activation='tanh'))

# Fourth hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer4, activation='linear'))

# Fifth hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer5, activation='relu'))



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

Which increased our accuracy to 75%

```
268/268 - 0s - loss: 0.4956 - accuracy: 0.7515 - 337ms/epoch - 1ms/step
Loss: 0.4956136643886566, Accuracy: 0.7514868974685669
```

Summary

As demonstrated, multiple hidden layers should be considered in approaching this task yet further modifications are required to improve the overall accuracy of the model.