Deep Learning Challenge Analysis

Overview

Deep learning and neural networks were used to determine if applicants would be successful if they were funded by Alphabet Soup, whom previously funded over 34,000 organizations.

Data Processing – We started by removing the EIN and NAME columns from the dataset. The remaining columns were considered features for the model. Although NAME was added back in the second test. CLASSIFICATION and APPLICATION_TYPE was replaced with 'Other due to high fluctuation. The data was split into training and testing sets of data. The target variable for the model is "IS_SUCCESSFUL" and is verified by the value, 1 was considered yes and 0 was no. APPLICATION data was analyzed, and CLASSIFICATION's value was used for binning. Each unique value used several data point as a cutoff point to bin rare categorical variables together in a new value, 'Other'. Afterwards checked to see if binning was successful. Categorical variables were encoded by 'pd.get_dummies().

Model - Neural Network was applied on each model multiple layers. The number of features dictates the number of hidden nodes.

A three-layer training model generated 477 parameters. The first attempt came close at 72% which was under the desired 75%.

```
# 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()

# 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'))

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

# Check the structure of the model
nn.summary()
```

✓ 0.0s

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 7)                 350

 dense_1 (Dense)             (None, 14)                112

 dense_2 (Dense)             (None, 1)                 15

=================================================================
Total params: 477 (1.86 KB)
Trainable params: 477 (1.86 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
# 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}")
```

✓ 0.1s

```
268/268 - 0s - loss: 0.5497 - accuracy: 0.7296 - 90ms/epoch - 335us/step
Loss: 0.5496866106987, Accuracy: 0.7295626997947693
```

Deep learning models should have multiple layers, since it is machined based it teaches a computer to filter inputs through the layers to learn how to predict and classify information.