Deep Learning - Charity Funding Predictor

Nicole Van Driss

Overview

The purpose of this assignment is to use deep learning and neural networks to determine if applicants would be successfully funded by Alphabet Soup.

Results:

- Data Preprocessing
    - In the first test, EIN and NAME were dropped from the dataset because of their irrelevance. However, NAME was re-added in the second test to increase accuracy. CLASSIFICATION and APPLICATION_TYPE was replaced due to high fluctuation. The data was then split into training and testing datasets. The target for the model is “IS_SUCCESSFUL”. The value of CLASSIFICATION was used for binning. Each value used several data points as a cutoff to bin categorical variables together in a new value. Categorical variables were encoded by pd.get_dummies().

- Compiling, Training, and Evaluating the Mode
    - Three layers were applied on each model. The number of features determined the number of hidden nodes.

Compile, Train and Evaluate the Model

```
# 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()
```

```
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
Trainable params: 477
Non-trainable params: 0
_________________________________________________________________
```

```
In [36]: # 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 - 0s - loss: 0.5532 - accuracy: 0.7307 - 273ms/epoch - 1ms/step
         Loss: 0.5532045364379883, Accuracy: 0.7307288646697998
```

click to expand output; double click to hide output

A three-layer training model generated 477 parameters. The accuracy came to 73%, lower than the expected 75%.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 7)                 3171

 dense_1 (Dense)             (None, 14)                112

 dense_2 (Dense)             (None, 1)                 15

=================================================================
Total params: 3,298
Trainable params: 3,298
Non-trainable params: 0
_________________________________________________________________
```

click to scroll output; double click to hide

```
In [30]: # 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 - 0s - loss: 0.4720 - accuracy: 0.7848 - 264ms/epoch - 984us/step
         Loss: 0.4720495343208313, Accuracy: 0.7848396301269531
```

Summary: A second attempt was completed for optimization. ‘NAME’ was added back into the dataset. The accuracy increased to 78% which is 3% over target. The model generated a total of 3,298 params. It is necessary for deep learning models to have multiple layers in order to filter inputs and predict or classify information more accurately.