DEEP LEARNING CHALLENGE: CHARITY FUNDING PREDICTOR

A homework assignment from the University of Birmingham Data Analytics Bootcamp (April 2022)

James Fairgrieve

Overview

The non-profit foundation Alphabet Soup wants to create an algorithm to predict whether applicants for funding will be successful. The organisation has provided a CSV file containing more than 34,000 organisations that have received funding from Alphabet Soup over the years. Within this dataset are a number of columns that capture metadata about each organisation, such as the following:

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

The foundation wants the final model to achieve a final model accuracy of over 75%.

Results

Data Pre-processing

The obvious target in the model is the **IS_SUCCESSFUL** column, as this is the indicator for whether an applicant is successful or not. The variables that could be considered to be potential features of the final model are:

- **NAME**
- **APPLICATION_TYPE**
- **AFFILIATION**
- **CLASSIFICATION**
- **USE_CASE**
- **ORGANIZATION**
- **INCOME_AMT**
- **ASK_AMT**
- **IS_SUCCESSFUL**

At this stage, this leaves **EIN**, **STATUS** & **SPECIAL_CONSIDERATIONS** as the remaining variables that can be removed.

For the model, there were two hidden layers used with 5 & 10 neurons. The final output layer had one unit as this model looked at the binary yes or no for whether the funding was successful.

```
# Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.
length = len(X_train_scaled[0])
layer1 = 5
layer2 = 10

nn = tf.keras.models.Sequential()

# First hidden layer
nn.add(tf.keras.layers.Dense(units = layer1, input_dim = length, activation = "relu"))

# Second hidden layer
nn.add(tf.keras.layers.Dense(units = layer2, input_dim = length, activation = "relu"))

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

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

```
Model: "sequential"
```

Including the **NAME** variable to create more parameters in the model resulted in a higher accuracy than in previous attempts. Including this column resulted in a model accuracy of 77.3%, surpassing the target set by Alphabet Soup.

```
Epoch 1/100
724/724 [==============================] - 2s 2ms/step - loss: 0.5587 - accuracy: 0.7293 - val_loss: 0.4818 - val_accuracy:
0.7688
Epoch 2/100
724/724 [==============================] - 1s 2ms/step - loss: 0.4667 - accuracy: 0.7759 - val_loss: 0.4552 - val_accuracy:
0.7769
Epoch 3/100
724/724 [==============================] - 1s 2ms/step - loss: 0.4569 - accuracy: 0.7786 - val_loss: 0.4513 - val_accuracy:
0.7765
Epoch 4/100
724/724 [==============================] - 1s 2ms/step - loss: 0.4532 - accuracy: 0.7797 - val_loss: 0.4506 - val_accuracy:
0.7777
Epoch 5/100
724/724 [==============================] - 1s 2ms/step - loss: 0.4516 - accuracy: 0.7817 - val_loss: 0.4496 - val_accuracy:
0.7792
Epoch 6/100
724/724 [==============================] - 1s 2ms/step - loss: 0.4507 - accuracy: 0.7817 - val_loss: 0.4497 - val_accuracy:
0.7820
Epoch 7/100
```

```
In [16]: # 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: 0.4665 - accuracy: 0.7727 - 272ms/epoch - 1ms/step
Loss: 0.4664676785469055, Accuracy: 0.7727113962173462
```

Summary

In this case, simply increasing the number of parameters in the model has increased the accuracy, but there are other ways that a more accurate model can be achieved:

- Adding more hidden layers
- Adding more neurons to the hidden layers