Charity Funding Predictor - Deep Learning Homework Report

By Alex Mull-Dreyer
February 19, 2022

Overview

The purpose of this analysis is to help the non-profit foundation **Alphabet Soup** create an algorithm to predict whether or not applicants for funding will be successful.

Overview: My Actions

I used the features in the provided dataset of 34,000 organizations receiving funding previously to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup

Data Preprocessing

The first task was removing the non-beneficial ID columns, EIN and NAME, from the model.

Variables that were considered features + some taken out

The variables that were considered features were APPLICATION_TYPE, AFFILIATION, CLASSIFICATION, USE_CASE, ORGANIZATION, STATUS, INCOME_AMT, SPECiAL_CONSIDERATION, and ASK_AMT. “Classification” and Application Type are neither targets nor features and should be removed from the input data.

The data was split into training and testing sets; I used the CLASSIFICATION value counts for binning.

Variables that are considered targets for the model

The target variable for the model was “IS_SUCCESSFUL.”

Compiling, Training, and Evaluating the Model

```
# 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_nodes_layer1 = 5
hidden_nodes_layer2 = 10
hidden_nodes_layer3 = 20

nn = tf.keras.models.Sequential()

# First hidden layer
nn.add(tf.keras.layers.Dense(units = hidden_nodes_layer1, input_dim = 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()
```

In terms of the number of neurons, layers, and activation functions selected for the neural network model, please refer to the following subsections for results.

Layers

There were three layers total applied using the Neural Network model.

Activation functions

There were three activation functions selected.

Neurons

There were 30 neurons used.

This model was able to generate 316 parameters with a 72% accuracy score below the 75%

threshold. No further steps were taken besides analysis to try to increase model performance.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 5)                 245

 dense_1 (Dense)             (None, 10)                60

 dense_2 (Dense)             (None, 1)                 11

=================================================================
Total params: 316
Trainable params: 316
Non-trainable params: 0
_________________________________________________________________
```

Summary

Recommendation on solving the classification problem