# Deep Learning Project: Charity Funding Predictor

Alice Krauze

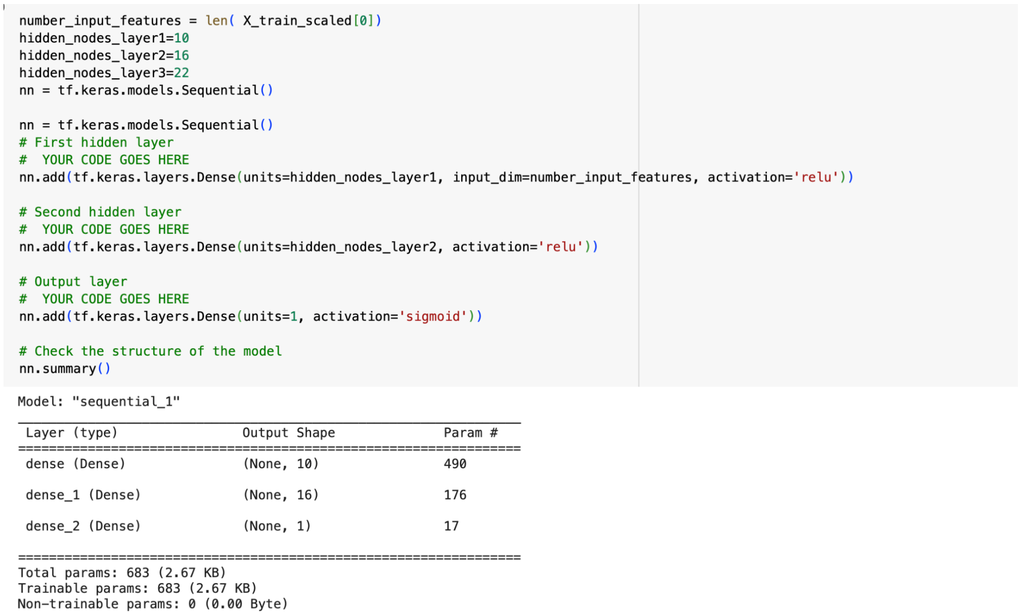
The Scope of this project was to determine whether applicants will be successfully funded by Alphabet Soup. The data set provided included 34,000 organizations that have received successful funding from Alphabet Soup over the years. The data tells us information about EIN and name, application type, affiliation, classification, use-case for funding, organization type, status, income, special considerations, amount requested, and if the money was used successfully.

## Data Preprocessing

The target for the model was IS_SUCCESSFUL, which was 1 indicating whether the money was used successfully or not, 1 indicating yes and 0 indicating no. The variables which were removed included EIN and Name because neither are targets or features.

## Compiling, Training, and Evaluating the Model

The target was 75% accuracy. The highest accuracy obtained was about 74%. However, the different models were altered to try to reach this accuracy. After model 1 did not reach this accuracy model 2,  I increased the amount of nodes. In model 3, I increased the number of epochs dramatically, while also increasing the number of values in each bin. In model 4, I decreased the number of epochs. All four attempts yielded very similar accuracy results ranging between 72-74%.



```
number_input_features = len( X_train_scaled[0])
hidden_nodes_layer1=10
hidden_nodes_layer2=16
hidden_nodes_layer3=22
nn = tf.keras.models.Sequential()

nn = tf.keras.models.Sequential()
# First hidden layer
#  YOUR CODE GOES HERE
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer1, input_dim=number_input_features, activation='relu'))

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

# Output layer
#  YOUR CODE GOES HERE
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, 10)                490

 dense_1 (Dense)             (None, 16)                176

 dense_2 (Dense)             (None, 1)                 17

=================================================================
Total params: 683 (2.67 KB)
Trainable params: 683 (2.67 KB)
Non-trainable params: 0 (0.00 Byte)
```

## Summary

The model's best result employing the different numbers of neurons and layers was about a 74% accuracy for the relu and sigmoid activations. Deep Learning models should have multiple layers because it is machine based and each layer further teaches the computer to filter inputs to learn how to classify and predict information. I would recommend reducing the number of epochs since it did not change the yield results and will save the data scientist some time. In future attempts, I would consider adding more layers to help yield more accurate predictions. The random forest model could also be used next to achieve desired results, it might be a better solution as it aggregates the data, reduces overfitting, and can handle high dimensional data.