

# Report on the Neural Network Model

## 1. **Overview** of the analysis: Explain the purpose of this analysis.

Ans: This analysis is to help Alphabet Soup, a nonprofit organization, determine which applicants will have the best success rates in their ventures.

## 2. **Results:** Using bulleted lists and images to support your answers, address the following questions:

### • Data Preprocessing

- What variable(s) are the target(s) for your model? 0 for no.
- What variable(s) are the features for your model?
- What variable(s) should be removed from the input data because they are neither targets nor features?

ANS: Data processing for this dataset involved the removal of all unnecessary columns. In this dataset, the columns EIN and NAME were removed. APPLICATION\_Data and CLASSIFICATION were used for binning and also replaced with Other for each unique variable. For this model, the target (s) variables(s) are "IS\_SUCCESSFUL". This has a value of 1 for yes and 0 for no.

### • Compiling, Training, and Evaluating the Model

- How many neurons, layers, and activation functions did you select for your neural network model, and why?
- Were you able to achieve the target model performance?
- What steps did you take in your attempts to increase model performance?

## 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.
# YOUR CODE GOES HERE
number_of_input_features = len(X_train_scaled[0])
hidden_nodes_layer1=80
hidden_nodes_layer2=30
hidden_nodes_layer3=1
nn = tf.keras.models.Sequential()

# First hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer1, activation="relu", input_dim=number_of_input_features))

# 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\_2"

Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 80)	9,040
dense_4 (Dense)	(None, 30)	2,430
dense_5 (Dense)	(None, 1)	31

Total params: 11,501 (44.93 KB)

Trainable params: 11,501 (44.93 KB)

Non-trainable params: 0 (0.00 B)

```
# 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 - 493us/step - accuracy: 0.7304 - loss: 0.5951
Loss: 0.5950543284416199, Accuracy: 0.7303789854049683
```

ANS: There were three layers and activation functions selected because three features which influence the number of hidden nodes. The model generated 11,501 parameters which represents 73.04% which is lower than the desired 75%

3. **Summary:** Summarize the overall results of the deep learning model. Include a recommendation for how a different model could solve this classification problem, and then explain your recommendation.

ANS: Deep Learning Models need to have several layers since the model needs to be filtered to classify and predict information.