

Deep Learning Project

In order to determine whether Alphabet Soup, which has already financed over 30,000 organizations, will be successful in funding the applications, it was necessary to use deeper learning and neural nets.

Data Processing

No irrelevant information has been removed from the dataset; therefore, EIN and NAME have been removed from the model. The features of the model were considered to be the remaining columns. Training and testing sets of data have been separated. The model's target variable is "IS_SUCCESSFUL" and has been verified by the value, 1 was taken to be a yes while 0 was not. The application data have been analyzed and the CLASSIFICATION value has been used to bin. For each unique value, a number of data points were used as a threshold for bin "rare" categorical variables, together with a new value, "other". After that, I checked to see if the binning had been successful. "pd.get_dummies()" encoded the category variables.

Compiling, Training, and Evaluation

Multiple layers, three in total, of the neural network have been applied to each model. The number of hidden nodes was determined by the number of features. Parameters were generated by a three layer 505 training model.

Compile, Train and Evaluate the Model

```
In [56]: # Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.
# YOUR CODE GOES HERE
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_7"

Layer (type)	Output Shape	Param #
dense_10 (Dense)	(None, 7)	378
dense_11 (Dense)	(None, 14)	112
dense_12 (Dense)	(None, 1)	15

```
=====
Total params: 505 (1.97 KB)
Trainable params: 505 (1.97 KB)
Non-trainable params: 0 (0.00 Byte)
```