Deep Learning Project

By Malek Kheirddin

Introduction

The provided dataset, sourced from Alphabet Soup's business team, includes over 34,000 records of previously funded organizations. Each record contains various columns capturing metadata about these organizations. By analyzing these features, our binary classifier will assist Alphabet Soup in identifying the most promising applicants, thereby maximizing the impact of their funding. And the steps are the following

Processing the data

The initial step involved uploading the dataset into Jupyter Notebook for analysis. During this phase, two non-beneficial ID columns, 'EIN' and 'NAME', were identified and removed to streamline the dataset and focus on more relevant features.

To gain a comprehensive understanding of the data, I assessed the number of unique values in each column. This step was crucial in identifying the variety of categorical variables present in the dataset, which would later require appropriate transformations for analysis.

An analysis of the 'APPLICATION_TYPE' column was conducted by examining its value counts. This analysis revealed a wide distribution of application types, prompting the need to consolidate less frequent types. A cutoff value was chosen to group application types with fewer occurrences into a single category labeled 'Other'. This approach helped simplify the dataset and reduce the complexity of the categorical variables.

Similarly, the 'CLASSIFICATION' column was analyzed by evaluating its value counts. Infrequent classifications were identified and grouped into a single category to streamline the dataset. This process, known as binning, involved selecting a cutoff value to determine which classifications to consolidate. The consolidated categories were then labeled appropriately to ensure consistency in the analysis.

With the categorical data consolidated, the next step was to convert these categories into numeric values. This transformation was essential for preparing the dataset for machine learning algorithms, which typically require numerical input. The conversion process ensured that all categorical variables were represented in a format suitable for modeling.

The final step in the preprocessing phase involved splitting the dataset into features and target variables. The features comprised the processed columns, while the target variable represented the success outcome. This step was critical in preparing the data for the subsequent modeling phases, enabling the creation of a binary classifier to predict the success of applicants funded by Alphabet Soup.

By following these steps, the dataset was effectively preprocessed, ensuring it was ready for building a predictive model to enhance the decision-making process of Alphabet Soup.

Compile, Train and Evaluate the Model

After the data set was ready it was time to train the model as the following:

```
# Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.
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()

# 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_4"

Layer (type)	Output Shape	Param #
dense_6 (Dense)	(None, 7)	350
dense_7 (Dense)	(None, 14)	112
dense_8 (Dense)	(None, 1)	15

Total params: 477 (1.86 KB)
Trainable params: 477 (1.86 KB)
Non-trainable params: 0 (0.00 B)

```
# Compile the model
#  YOUR CODE GOES HERE
nn.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])
```

```
# Train the model
#  YOUR CODE GOES HERE

train_model = nn.fit(X_train_scaled,y_train,validation_split=0.15, epochs=100)
```

I received and accuracy score of 0.726 with loss of 0.55