NEURAL NETWORK MODEL REPORT

Leslie Leiva

February 14, 2024

Purpose of analysis

Create a neural network model that has 75% accuracy in successfully matching applicants with funding opportunities. The features for this model will be a binary classifier that can predict whether applicants will be successful.

Data preprocessing

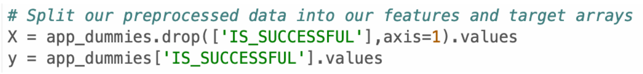
Dropping non-beneficial columns: 'EIN', 'SPECIAL_CONSIDERATIONS', & 'ASK_AMT'

Finding the number of data points for columns with more than 10 unique values and choosing the cut off points to bin rare categorical values:

- *'APPLICATION_TYPE' = 700.*
- *'CLASSIFICATION = 200.*
- *'NAME' = 50.*

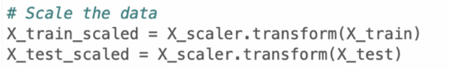
Compiling, Training, Evaluating the Model

Converting categorical data into numeric by utilizing 'pd.get_dummies()' and divide the data into a target and feature array



```
# Split our preprocessed data into our features and target arrays
X = app_dummies.drop(['IS_SUCCESSFUL'],axis=1).values
y = app_dummies['IS_SUCCESSFUL'].values
```

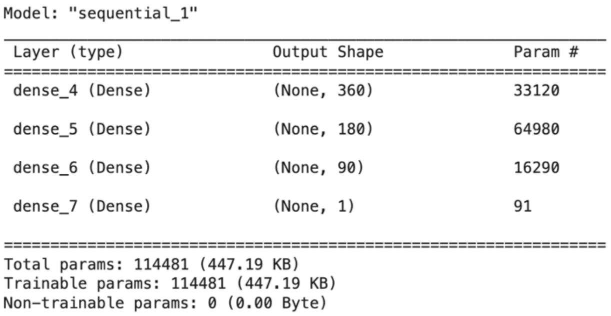
The 'train_test_split" method was applied to create a testing and training dataset. A Standard Scalar instance was created, fitted to the' X_training' variable and transformed.



```
# Scale the data
X_train_scaled = X_scaler.transform(X_train)
X_test_scaled = X_scaler.transform(X_test)
```

The model was defined by number_input_features, three hidden nodes layers and an outer nodes layer to provide a model summary.

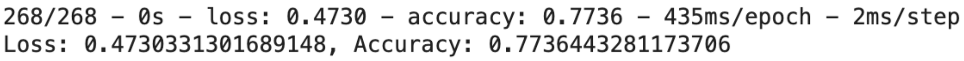
- *number_input_features = len(X_train[0])*
- *hidden_nodes_layer1 = 360*
- *hidden_nodes_layer2 = 180*
- *hidden_nodes_layer3 = 90*
- *outer_nodes_layer = 1*



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_4 (Dense)             (None, 360)               33120

 dense_5 (Dense)             (None, 180)               64980

 dense_6 (Dense)             (None, 90)                16290

 dense_7 (Dense)             (None, 1)                 91

=================================================================
Total params: 114481 (447.19 KB)
Trainable params: 114481 (447.19 KB)
Non-trainable params: 0 (0.00 Byte)
```

Model was then compiled using the `'binary_crossentropy'` *as the loss and* `'adamax'` *as the optimizer then the model was trained using test scaled and fit variables.*

Model was then evaluated using the evaluate method.



```
268/268 - 0s - loss: 0.4730 - accuracy: 0.7736 - 435ms/epoch - 2ms/step
Loss: 0.4730331301689148, Accuracy: 0.7736443281173706
```

💻 Summary

✓ *Initial model had a 71% accuracy. For optimization of the model to 77% the following was adjusted:*

- *Dropping 3 columns instead of 2. Attempts were made when majority of the columns were deleted resulting in 60% accuracy but when majority of the columns remained the accuracy increased by 1%.*

- *Another bin for ‘NAME’ column values was created for rare occurrences. By reducing the variation of value counts, the model will have an increased accuracy.*

- *Three hidden layers were added to the model with more neurons allowing the information to be filtered properly.*

- *For additional accuracy increase, the second hidden layer has the ‘leaky_relu’ activation function and the third layer had the “sigmoid” activation function.*

- *Model optimizer that was used for compiling was ‘adamax’ and the batch size was reduced to 10 to process smaller data sets increasing the accuracy by 1%.*