Deep Learning

Charity Analysis Report

Overview:

The purpose of our analysis is to determine which model or models could best predict which applicants have the best chance of success in their ventures. We will look at the past performance of applicants and use their features and outcomes to aide in determining if a future applicant is likely to experience success.

Results:

* Data Preprocessing
  + Drop columns that do not provide value as Features
    - EIN
    - NAME
  + Identified Features \*\*
    - APPLICATION\_TYPE
    - AFFILIATION
    - CLASSIFICATION
    - USE\_CASE
    - ORGANIZATION (type)
    - STATUS
    - INCOME\_AMT
    - SPECIAL\_CONSIDERATIONS
    - ASK\_AMT
  + Identified Target
    - IS\_SUCCESSFUL
  + Data Reduction
    - Classification - Binning
      * Reduced by combining all classifications with and occurrence of less than 100 into an instance of other.
    - Application Type - Binning
      * Reduced by combining all classifications with and occurrence of less than 500 into an instance of other.
  + Categorical Data
    - Converted using Get\_dummies
* Compiling, Training, and Evaluating the Model
  + Keras tuner was used to determine the number of neurons, layers and activation functions for the neural network.
    - Activations sent to keras tuner for evaluation.
      * Relu
      * Tanh
      * Sigmoid
    - Neuron Range
      * 1 to 50
    - Layer Range
      * 1 to 14
    - Epochs
      * 1-50
    - Optimizers
      * Adam
      * Nadam
      * Adamax
  + Experienced Results
    - The desired rate of 75% was not achieved. The following steps were taken to improve the results
      * Changed the range of Neurons to test
      * Changed the range of Layers to test
      * Changed the range of Epochs to test
      * Changed the Optimizers that were tested in each run.
      * Changed the number of Features used\*\*

The optimizer reports the best model as:

{'activation': 'sigmoid',

'first\_units': 37,

'num\_layers': 3,

'units\_0': 7,

'units\_1': 11,

'units\_2': 7,

'units\_3': 7,

'units\_4': 7,

'units\_5': 1,

'tuner/epochs': 50,

'tuner/initial\_epoch': 17,

'tuner/bracket': 1,

'tuner/round': 1,

'tuner/trial\_id': '0074'}

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense (Dense) (None, 37) 1850

dense\_1 (Dense) (None, 7) 266

dense\_2 (Dense) (None, 11) 88

dense\_3 (Dense) (None, 7) 84

dense\_4 (Dense) (None, 1) 8

=================================================================

Total params: 2,296

Trainable params: 2,296

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

However, the model that I opted to use is:

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense (Dense) (None, 3) 150

dense\_1 (Dense) (None, 9) 36

dense\_2 (Dense) (None, 11) 110

dense\_3 (Dense) (None, 1) 12

=================================================================

Total params: 308

Trainable params: 308

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Which results in:

268/268 - 0s - loss: 0.5576 - accuracy: 0.7341 - 498ms/epoch - 2ms/step

Loss: 0.5575608611106873, Accuracy: 0.7341107726097107

The model that was ultimately used was discovered in a keras turner run that had more restrictive ranges. While it would stand to reason that it would also be the best model when the available ranges are increased, it is not reported as the best model for the larger run. There is no significant difference in the accuracy of these models. The one I selected has 308 total parameters where the suggested model has 2,296 parameters. We can achieve marginally better results with many fewer parameters.

\*\*Note on Identified features: models were tested with most of the identified features removed one at a time, the absence of any specific column did not improve any of the testing instances.

Alternate Models:

There are several models that work well in predicting binary results. Our result is a binary result as an activity either succeeds or it fails. Among those Naïve Bayes, Logistic Regression, K-Nearest Neighbors, SVM and several others. If I were to try one, I might write if function to run through 5 or 6 of them to determine the most accurate fit. If I were to recommend one without testing, I would recommend Logistic Regression out of the box. There are a couple reasons for my selection, it is simple and widely used and as a result widely understood. It also benefits from clean and easy to understand visuals.