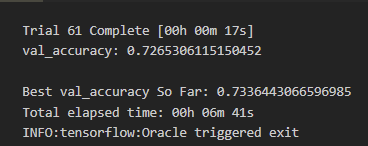
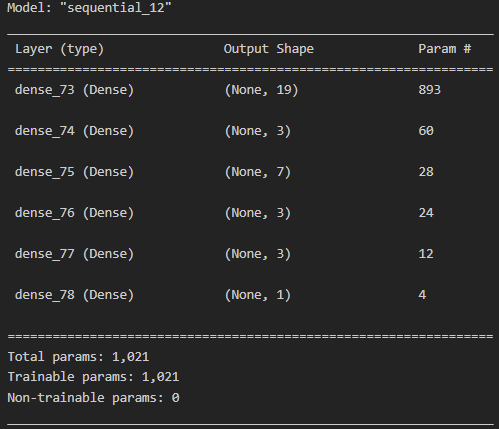
Overview:

The non-profit foundation Alphabet Soup wants to create an algorithm to predict whether or not applicants for funding will be successful. The features in the provided dataset were used to create a binary classifier using TensorFlow and Keras to predict whether applicants will be successful if funded by Alphabet Soup.

Results:

* Data Preprocessing
  + What variable(s) are considered the target(s) for your model?
    - IS\_SUCCESSFUL—Was the money used effectively
  + What variable(s) are considered to be the features for your model?
    - APPLICATION\_TYPE—Alphabet Soup application type
    - AFFILIATION—Affiliated sector of industry
    - CLASSIFICATION—Government organization classification
    - USE\_CASE—Use case for funding
    - ORGANIZATION—Organization type
    - STATUS—Active status
    - INCOME\_AMT—Income classification
    - SPECIAL\_CONSIDERATIONS—Special consideration for application
    - ASK\_AMT—Funding amount requested
    - IS\_SUCCESSFUL—Was the money used effectively
  + What variable(s) are neither targets nor features, and should be removed from the input data?
    - EIN and NAME—Identification columns
* Compiling, Training, and Evaluating the Model
  + How many neurons, layers, and activation functions did you select for your neural network model, and why?
    - Layers: 5
      * Neurons: 19, Activation: tanh
      * Neurons: 3, Activation: tanh
      * Neurons: 7, Activation: relu
      * Neurons: 3, Activation: relu
      * Neurons: 3, Activation: relu
    - Used auto-optimization function to test various hyperparameters for 1 to 6 layers, 1-22 neurons per layer, with activation relu, tanh, and sigmoid activation functions to determine the best hyperparameters. The function was commented out and just the best parameters used to save time when re-running.
    - 
    - 
  + Were you able to achieve the target model performance?
    - No; after optimization I was only able to achieve an accuracy of 73.1%
    - 
  + What steps did you take to try and increase model performance?
    - Dropping STATUS, AFFILIATION, and ORGANIZATION, but this did not improve accuracy.
    - Increasing the number of bins by only binning those with < 150 occurrences as other.
    - Performing auto-optimization function to determine ideal hyperparameters for layers, neurons, and activation functions.
    - Adding up to 200 epochs, but these did not improve score, and thus 50 was settled on to save time when re-running

Summary:

After processing the data to identify key features and optimizing the model, the model is able to predict whether not the historical applicants were successful or not with an accuracy of 73.1%. Depending on how the training data was split, the accuracy could vary by ~1%, but this is below the 75% accuracy threshold that we were targeting. As there is a target for this data set we can use other supervised machine learning models such as logistic regression or random forest classifier as an alternative to attempt to find a model that can achieve higher accuracy.