Deep Learning: Charity Funding Predictor

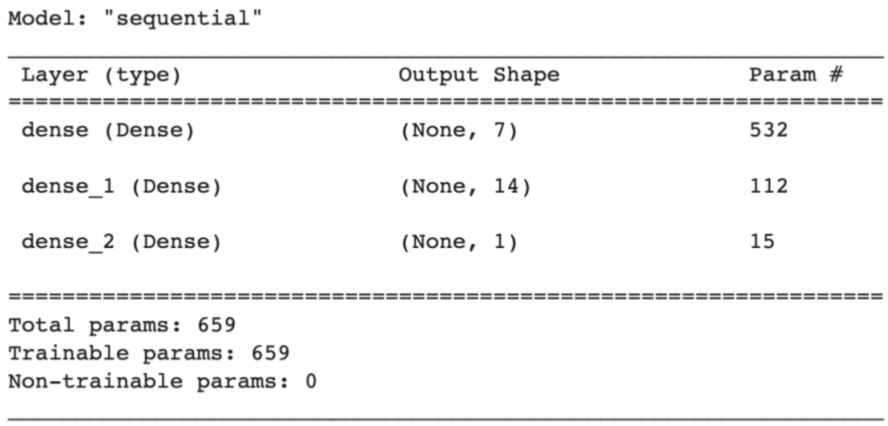
We were able to predict from a dataset, containing more than 34,000 organizations that have received funding from Alphabet Soup over the years, whether a campaign would be successful or not using machine learning and neural networks.

- Data Preprocessing:
    - On the first try, columns EIN and NAME were removed from the dataset the as irrelevant information. These columns were neither targets, nor features. Everything else was considered as features for the model.

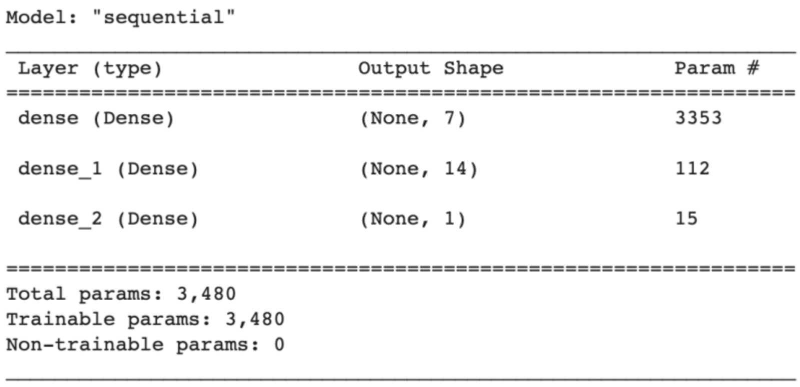
    - The data was split by IS_SUCCESSFUL and the rest of the features. IS_SUCESSFUL was the variable target for the model.

    - For the Optimizer, column NAME was part of the features. Only EIN was dropped.

- Compiling, Training, and Evaluating the Model:
    - We applied a three-layer model. The number of hidden nodes were decided by the number of features. The first model gave out 659 params vs 3480 in the optimized one.

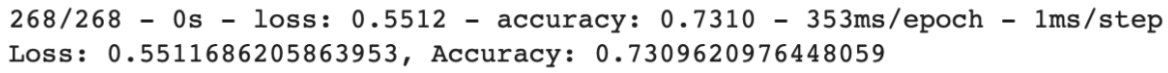


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 7)                 532

 dense_1 (Dense)             (None, 14)                112

 dense_2 (Dense)             (None, 1)                 15

=================================================================
Total params: 659
Trainable params: 659
Non-trainable params: 0
_________________________________________________________________
```

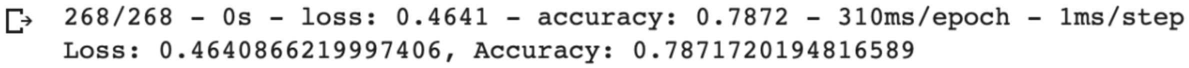


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 7)                 3353

 dense_1 (Dense)             (None, 14)                112

 dense_2 (Dense)             (None, 1)                 15

=================================================================
Total params: 3,480
Trainable params: 3,480
Non-trainable params: 0
_________________________________________________________________
```

- On the first try we achieved 73% accuracy, once we optimized the model we were able to achieve 78%



```
268/268 - 0s - loss: 0.5512 - accuracy: 0.7310 - 353ms/epoch - 1ms/step
Loss: 0.5511686205863953, Accuracy: 0.7309620976448059
```



```
268/268 - 0s - loss: 0.4641 - accuracy: 0.7872 - 310ms/epoch - 1ms/step
Loss: 0.4640866219997406, Accuracy: 0.7871720194816589
```

- We played around increasing the hidden layers and the numbers of nodes. After several tries, the accuracy would not surpass 75%
- Increasing the data by incorporating the column NAME, gave the model more to train with. That is how 78% accuracy was achieved.

Summary:

This deep learning model was most effective when provided more data. By leaving in NAME and encoding categorical values, more unique features were added to the model. This gave the model more params to train with.