

The purpose of this analysis was to use our knowledge of neural networks and deep learning to be able to predict if the applicants that have the best chance of success if we fund them. We would use the deep learning model to help inform our decisions.

For this analysis:

- The variable that we mainly targeted for this model was: IS_SUCCESSFUL as our target variable.
- Our Feature Variables were APPLICATION_TYPE, AFFILIATION, CLASSIFICATION, USE_CASE, ORGANIZATION, STATUS, INCOME_AMT, SPECIAL CONSIDERATION, and ASK_AMT.
- Removed from the analysis were the EIN and the NAME of the applicants.

When it came to training and evaluating the model, we attempted several different optimizations with the data.

- Our first attempt was able to reach an accuracy of 72.01%. For this, we had two hidden layers, one with 80 neurons and the other with 30, and it ran for 50 epochs.

```
268/268 - 0s - loss: 0.5542 - accuracy: 0.7201 - 282ms/epoch - 1ms/step  
Loss: 0.5541788339614868, Accuracy: 0.7201166152954102
```

- Our second attempt had the same data, however, with select changes. In this test, we added a third hidden layer, this with 10 neurons. We also let it run for an additional 50 epochs. This attempt yielded an accuracy of 72.41%.

```
268/268 - 0s - loss: 0.5617 - accuracy: 0.7241 - 358ms/epoch - 1ms/step  
Loss: 0.5617002844810486, Accuracy: 0.7240816354751587
```

- Our third attempt we changed once more, however this time with the preprocessed data. In addition to the columns previously removed from our data frame, we also dropped AFFILIATION, USE_CASE, and ORGANIZATION. The same number of hidden layers, neurons, and epochs yielded an accuracy of 62.5%.

```
268/268 - 0s - loss: 0.6330 - accuracy: 0.6250 - 321ms/epoch - 1ms/step  
Loss: 0.6330333948135376, Accuracy: 0.6249562501907349
```

- Our final attempt we went restored the dropped columns from the third attempt. This time, we added more neurons to our two hidden layers: a total of 100 for the first hidden layer, and 50 for the second. This time, we tried a different activation: from 'relu' to 'tanh'. This attempt had an accuracy of 72.45%.

```
268/268 - 0s - loss: 0.5638 - accuracy: 0.7245 - 214ms/epoch - 798us/step  
Loss: 0.5637790560722351, Accuracy: 0.7245481014251709
```

In summary, our feature variables were indeed the ones listed, and removing them decreased our accuracy. Our final optimization gave us an increase in our accuracy score, however, there is still room for improvement and more testing to reach it. While the tanh function, while more effective than ReLU, does require more computation more experimenting with that may yield a better accuracy than we attained.