

# **Report on the Neural Network Model**

## **Overview:**

The nonprofit foundation Alphabet Soup wants a tool that can help it select the applicants for funding with the best chance of success in their ventures.

With the help of machine learning and neural networks, we will use the features in the provided dataset to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup.

## **Results:**

### *Data Preprocessing:*

What variable(s) are the target(s) for your model?

- The variable "IS\_SUCCESSFUL" which shows whether the money was used effectively or not is our target.

What variable(s) are the features for your model?

- The variables 'APPLICATION\_TYPE', 'AFFILIATION', 'CLASSIFICATION', 'USE\_CASE', 'ORGANIZATION', 'SPECIAL\_CONSIDERATIONS', 'INCOME\_AMT', 'STATUS', and 'ASK\_AMT' are the features.

What variable(s) should be removed from the input data because they are neither targets nor features?

- The variables 'EIN', and 'NAME' are neither targets nor features and hence removed from the input data.

### *Compiling, Training, and Evaluating the Model :*

How many neurons, layers, and activation functions did you select for your neural network model, and why?

- To create the neural network model, I used two hidden layers each with 6 and 4 neurons. ReLU Rectified Linear Unit is used as activation function for both the hidden layers and Sigmoid is used for the output layer.

To avoid underfitting, I used two hidden layers with 6 and 4 neurons.

Were you able to achieve the target model performance?

- I achieved an Accuracy of 0.725 which is more than 70%.

What steps did you take in your attempts to increase model performance?

- To increase the model performance, I created a model with an additional hidden layer and an increased number of neurons, and another model with the same number of hidden layers but with an increased number of neurons.

This helped in increasing the number of trainable parameters in the model but the accuracy remained about the same with an increase of 0.002.

## **Summary:**

Looking at the summary of the model, we have

Model: Sequential

There are three dense layers. Output shapes are 6,4, and 1 same as the number of neurons.

The parameter weights are 702, 28 and 5 respectively.

The number of total trainable parameters in the model is 735 and there are no Non-trianable parameters in this model.

We got an accuracy of 0.72 with a loss of 0.559.

If I could use a different model, I would use a Random forest for this data. Random forest will sample the data and will build several smaller and simpler decision trees which can avoid underfitting and loss of data while performing the test.