ANALYSIS OF THE NEURAL NETWORK MODEL FOR ALPHABET SOUP

1. ***Overview*** of the analysis:

The nonprofit foundation Alphabet Soup wants a tool based on machine learning model that can help the organization in selecting the applicants, that have best chance of success in their ventures, for funding. The aim of the project is to create a deep learning model (neural network) that will use the features provided in the dataset (projects funded in the past) to create a binary classifier. This binary classification (Success or Not Success) will help predict whether applicants will be successful, if funded by Alphabet Soup.

1. ***Results****:*

* Data Preprocessing
  + What variable(s) are the target(s) for your model?

‘IS\_SUCCESSFUL’ is the target for the model. Outcome of 1 will predict that Alphabet Soup should fund the project as it has high chances of success. 0 outcome will predict that the venture has higher chance of failure.

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

Rest of the columns (besides NAME and EIN) in the dataset are the features. Some of the features that will be used in the model are Application type, Affiliation, Classification, funding amount asked, etc. [APPLICATION\_TYPE, AFFILIATION, CLASSIFICATION, USE\_CASE, ORGANIZATION, STATUS, INCOME\_AMT, SPECIAL\_CONSIDERATIONS, ASK\_AMT]

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

Columns NAME and EIN need to be removed from the dataset.

* Compiling, Training, and Evaluating the Model
  + How many neurons, layers, and activation functions did you select for your neural network model, and why?

I selected 3 layers with 43 input features for the first layer. There are 80 neurons in the second layer and 30 neurons in the third layer. ReLU activation method is used in first and second layer and Sigmoid is used the final layer.

The decision for number of layers and neurons in each layer is used to build a robust deep learning model without running into the risk of Overfitting or Underfitting the model.

The rectified linear unit (ReLU) function seemed like an ideal choice for the first and second layer of the model as we are handling positive and nonlinear input data for classification. To build an efficient and optimized model, the outer layer of the model using Sigmoid activation function.

* + Were you able to achieve the target model performance?

The model runs with an accuracy of 72.6%. It is not the best model but it is not bad either.

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

I took 3 steps to increase the model performance.

* + 1. Redistributed the bins for APPLICATION\_TYPE feature. Aggregated all 'APPLICATION\_TYPE' less the 1000 in one bin (earlier it was 500).
    2. Adding one more hidden layer (now total of 3) and readjusting the nodes for each layer. Moved to 80, 50, 30 (3 hidden layers) from 80, 30 (2 hidden layers) model.
    3. Changing the activation function to Tanh from ReLU for first three layers of the model.

1. ***Summary***:

Model Performance:

* Loss: 0.557
* Accuracy: 0.726
* Activation Method: ReLU

The model used ReLU and Sigmoid activation functions and achieved an accuracy of 72.6%. The loss value of the model is 55.7%, indicating the discrepancy between the predicted and actual values.

Even after making different adjustment to the model, there is no significant improvement in loss and accuracy of the model. My recommendation is to train the model on more data for better accuracy. If more data is unavailable, then alternatively break the data into batches and run different combination of batches to train the model.