**Report on Neural network Model**

**Overview of Analysis**

The purpose of creating a binary classifier using machine learning and neural networks for Alphabet Soup's dataset is to assist the foundation in identifying which applicants are most likely to succeed if funded. By analyzing the features of the organizations that have received funding in the past, the classifier aims to predict whether future applicants will be successful or not.

**Results**

1. **Data Preprocessing:**

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

**Answer:** The target for this model was “IS\_SUCCESSFUL”.

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

**Answer:** The variables for features of this model were 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?

**Answer:** The removed variables from the input data were EIN, NAME because they were neither targets nor features.

1. **Compiling, Training, and Evaluating the Model:**

* How many neurons, layers, and activation functions did you select for your neural network model, and why?
* **Answer:** For compiling, training, and evaluating the model I have used four layers that is input layer, two hidden layers and output layer. 80 nodes for first hidden layers, 30 nodes for second hidden layers and 1 node (binary classification) for output layers. The activation functions that I have used for hidden layer was Relu and for output layer sigmoid activation function.

The reason I have used these activation functions were Relu activation is commonly used in hidden layers as it avoids vanishing gradient issues. Sigmoid is useful for binary classification problem in the output layer to predict probability values between 0-1.

The hidden layers with decent node provide enough complexity for the model.

* Were you able to achieve the target model performance?

**Answer:** No, I was not able to achieve the target predictive accuracy which were 75% for the model.

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

**Answer:** The changes I have made to make model performance better are:

1. 4 hidden layers are used instead of 2 with additional layers of 256, 128 nodes. More 500 nodes in first hidden layer compared to 80 previously and 300 nodes in second hidden layer verses 30 earlier. This indicates a larger and deeper neural network architecture.
2. I have added Regulation techniques of dropout of 0.25 and 0.3 after layers and Batch normalization. These technique help prevent overfitting.
3. Added early stopping to prevent overfitting and trains for lesser epochs (50) but with tuned batch size.
4. So, the overall architecture is now a larger and deeper neural network with regularization for preventing overfitting. This allows model to learn more complex patterns and perform better.

**Summary**

**The neural network model is achieving an accuracy of 72.66% on the test set which indicates that the model is learning and modeling the patterns reasonably well, but there is still room for improvement. For a binary classification task, 70% is decent but not very high. The model capacity might not be sufficient to capture the underlying complex relationship in the data. Factors like number of layers, nodes, model depth affect what patterns can be learned. Overfitting could be a likely issue as neural nets can tend to overfit small or noisy datasets.**

**This model lacks regularization techniques like dropout. The optimization algorithm, batch size, number of epochs could be tuned further through hyperparameter optimization using cross validation.**

**However, I would recommend trying a different model altogether that is more suitable for tabular data with categorical features. Specially, a random forest classifier could work better for this problem.**