**Neural Network Model Report**

Author: Arpita Sharma

**Overview**

I created a neural network model for the nonprofit foundation Alphabet Soup, which would like a tool that can help it select the applicants for funding with the best chance of success in their ventures. I used various features provided in the dataset to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup using a CSV containing more than 34,000 organizations that received funding form Alphabet Soup over the years.

**Results**

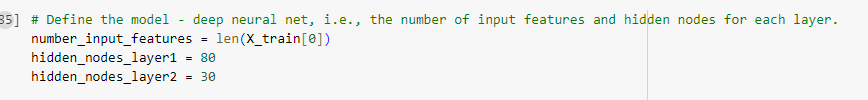
I produced a model which has 72.6% accuracy with a 50% loss.

***Data Preprocessing***

* The target variable for my model were IS\_SUCCESSFUL.
* Th feature variables for my model were:
  + APPLICATION\_TYPE
  + AFFILIATION
  + CLASSIFICATION
  + USE\_CASE
  + ORGANIZATION
  + STATUS
  + INCOME\_AMT
  + SPECIAL\_CONSIDERATIONS
  + ASK\_AMT
* The variables that were removed from the input data because they are neither targets nor features are EIN and NAME.

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

* I used two layers for the model to learn more complex representations of the input data. Each hidden layer can capture different features and relationships, enabling the network to understand hierarchical patterns and dependencies within the data. These can increase the capacity of the model to learn intricate patterns in the data, and allow it to be more expressive, improve the generalizations well to unseen data.



* I selected two neurons for the model. These are ReLU and Sigmoid. I used ReLU to introduce non-linearity to the model, enabling it to learn complex patterns and relationships in the data. It is computationally efficient and helps mitigate the vanishing gradient problem. I used Sigmoid to predict probability. The output of the Sigmoid function lies in the range (0,1), representing the likelihood of being successful.

A screenshot of a computer code

Description automatically generated

* I used three activations to introduce non-linearity to the neural network to enable it to learn and represent complex, non-linear relationships in the data. This also enables representation learning and supports backpropagation and mitigate the vanishing gradient problem.
* I was able to achieve the target model performance loss of .558 and accuracy of 0.726 using the model I created. Initially, I had a model loss of .60 and accuracy of 0.61, but when I reduced the cutoff value for the “CLASSIFICATION” from 1,800 to 1,000, it improved the performance of my model.

A screenshot of a computer

Description automatically generated

A computer code with numbers and text

Description automatically generated

Summary

In Summary, I used two hidden layers and three activations to create the neural network model to predict the applicants for funding with the best chance of success. The model relies on ReLU and Sigmoid and produces a loss of 0.5583 and an accuracy of 0.7265. To further optimize this model, I would recommend using Leaky ReLU or Parametric ReLU in addition to the ReLU and Sigmoid. This is because while ReLU is computationally and has lots of benefits, it suffers from the “dying ReLU” problem, where neurons can become inactive during training and stop learning. Leaky ReLU and Parametric ReLU could help address this issue.