**Deep Learning Challenge: Alphabet Soup**

**Overview:**

Alphabet Soup, a non-profit foundation wants a tool that can help select applicants to be funded with the best chance of success in their ventures. With the knowledge of machine learning and neural networks, we were tasked to use the features provided in a dataset to create a classifier (algorithm) capable of predicting applicants, if funded by the foundation, will be successful.

**Results:**

***Data Processing (First Trial):***

The data were processed by first removing any irrelevant information such as the EIN and NAME. The remaining columns were then considered as features for the model. The data was then split into two, used as training and testing sets. The target variable for the model, “IS\_SUCCESSFUL”, was then labeled 1 for yes 0 for no. the “APPLICATION\_TYPE” was then used for analysis and “CLASSIFICATION” was used for the binning process. The data was scaled for easier comparison and computation. And several data points were used as a cutoff and binned possible outliers to a new value named “OTHERS” for a cleaner data.

Compile-Train-Evaluate the Data Model.

Initially, we used a deep neural network with two (2) layers for the first trial. The hidden nodes were stated by the numbers below.

A screen shot of a computer

Description automatically generated

With these, 6,461 parameters were created by the two-layer model.

A screenshot of a computer program

Description automatically generated

With a 100 epochs or iterations, accuracy was about 73.78%. A little bit lower from the desired accuracy of 75%. Because of this, a new trial must be created to achieve the desired accuracy.

A screen shot of a computer code

Description automatically generated

***Data Optimization (Second Trial):***

To optimize the model, we decided to adjust some data processing. Just dropping the EIN, cleaning and binning the NAME, APPLICATION\_TYPE and CLASSIFICATION, scaled the data, adding hidden layers but lessening the number of nodes per layer did the job of increasing the accuracy more than the desired number of 75%.

We used three (3) hidden layers, instead of two (2) to really fit the model to the dataset. We also reconfigured the number of nodes from an 80-30 node structure to a 21-21-21 node structure which gave a slightly higher parameters as compared with the first trial.

A screenshot of a computer program

Description automatically generated

6,679 parameters were used in this trial.

A screenshot of a computer

Description automatically generated

With only 50 epochs or iterations, we got an accuracy of 77.59%, 2.59% higher than the desired target of 75%.

A screen shot of a computer code

Description automatically generated

**Recommendations:**

Multiple layers is a given in deep learning models. Multiple layers can learn features at different levels of abstraction. They can learn more detailed features between the input data to get the best model out of the dataset given.