**Deep Learning Challenge**

This deep learning challenge is for a non-profit foundation, Alphabet Soup, to determine to which applicants they should provide the funding. The CSV file contains historical data of organizations which have received fundings from Alphabet Soup, including name of the organizations, application type, affiliated sector of industry, government organization classification, use case for funding, organization type, active status, income classification, any special aspects to consider for application, funding amount, and finally whether the applicant was successful or not. With machine learning and neural networks, we set up a model and made it to predict future applicants.

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

* The 'IS\_SUCCESSFUL' column from application\_df is the target variable, this is what we are trying to predict. This shows if the money was used effectively.
* What variable(s) are the features for your model? The feature variables we used are:

AFFILIATION—Affiliated sector of industry

CLASSIFICATION—Government organization classification

USE\_CASE—Use case for funding

ORGANIZATION—Organization type

STATUS—Active status

INCOME\_AMT—Income classification

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

Identification columns: The "EIN" and "NAME" columns are identification columns that typically provide unique identifiers for each organization. These columns usually have no direct impact on the target variable and can be dropped without affecting the model's accuracy.

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

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

A screenshot of a computer

Description automatically generated

At first model, I used a two-layer architecture with a specific choice for the number of neurons, layers, and activation functions.

Opting for an architecture with 80 neurons in the initial hidden layer (units\_1 = 80) and 30 neurons in the subsequent hidden layer (units\_2 = 30), coupled with the utilization of the ReLU activation function (activation="relu") for both concealed layers, your intention was to forge a model possessing the necessary intricacy to adeptly identify and internalize significant patterns residing within the data. This deliberate use of ReLU activation serves to inject non-linearity into the model, thus enabling it to adeptly discern intricate interdependencies that exist between the input features and the ultimate target variable.

#### Were you able to achieve the target model performance?

#### As evident from the results displayed below, my accomplishment was confined to a 73.47% success rate, falling short of the aimed-for model performance of 75%.

#### A screen shot of a computer code Description automatically generated

Model 2

Adding more layers to the model:

A screenshot of a computer program

Description automatically generatedAdding more layers can furnish the model with augmented capacity to grasp and depict intricate interconnections ingrained within the data. Every layer can assimilate distinct strata of abstraction, empowering the model to extract deeper, more insightful characteristics, consequently elevating the potential for accuracy enhancement. Models of profound depth, characterized by multiple layers, exhibit a propensity to master hierarchical depictions of the data, thereby conferring a strategic edge for addressing intricate challenges.

A screen shot of a computer code

Description automatically generated

We have the same accuracy 73.4%

Model 3

Using a different activation function for the hidden layers

A screenshot of a computer program

Description automatically generated

Incorporating an alternative activation function, like tanh, can exert a distinct influence on the model's perception and manipulation of the inputs. Varied activation functions come with diverse attributes, enabling them to encapsulate varying forms of non-linear behaviors. The adoption of tanh introduces a unique non-linearity that might align more harmoniously with the specific intricacies of the problem, thus potentially fostering an uptick in accuracy.

The accuracy in this model is 72.97%

A screen shot of a computer code

Description automatically generated