Neural Network Report for Deep Learning

Purpose of report:

The nonprofit foundation Alphabet Soup wants a tool that can help it select the applicants for funding with the best chance of success in their ventures. I’ll use the features in the provided dataset to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup.

From Alphabet Soup’s business team, I have received a CSV containing more than 34,000 organizations that have received funding from Alphabet Soup over the years. Within this dataset are a number of columns that capture metadata about each organization, such as:

* **EIN** and **NAME**—Identification columns
* **APPLICATION\_TYPE**—Alphabet Soup application type
* **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
* **SPECIAL\_CONSIDERATIONS**—Special considerations for application
* **ASK\_AMT**—Funding amount requested
* **IS\_SUCCESSFUL**—Was the money used effectively

Results:

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

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense (Dense) (None, 80) 5360

dense\_1 (Dense) (None, 30) 2430

dense\_2 (Dense) (None, 1) 31

=================================================================

Total params: 7,821

Trainable params: 7,821

Non-trainable params: 0

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

The column IS\_SUCCESSFUL is an example of a feature.

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

| **APPLICATION\_TYPE\_Other** | **APPLICATION\_TYPE\_T10** | **APPLICATION\_TYPE\_T12** | **APPLICATION\_TYPE\_T13** | **APPLICATION\_TYPE\_T19** | **APPLICATION\_TYPE\_T2** | **APPLICATION\_TYPE\_T** |
| --- | --- | --- | --- | --- | --- | --- |

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

The one with the best accuracy is 3 layers, with 100, 50, 10 functions.

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

No, but came close with 73%.

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

I used Sigmoid instead of Relu, and added layers.

A close-up of a computer screen

Description automatically generated with low confidence

A screenshot of a computer program

Description automatically generated with low confidence

Summary of results:

After 3 reoptimizing attempts the second attempt was the most at 73.09. In the first attempt I dropped the USE\_CASE column and decreased the number of values for each bin. The second attempt, I added more neurons to hidden layers and added another hidden layer, which was the most accurate. The third optimization attempt, I used different activation functions for the hidden layers and doubled the number of epochs.

Different Model:

A different model to use would be one with more hidden layers and neurons since the optimizations had the most accuracy with this strategy. The problem with this model would be the resources to compute this model and time involved.