Romario Leal
07/23/23

Module 21: Deep-Machine-Learning

Overview:

The goal of this project was to create a tool to assist the Alphabet Soup non profit organization. They intend to use this tool to predict if applicants will be successful if they receive funding from Alphabet Soup. Using a csv file containing a list of more than 34,000 organizations I will utilize machine learning in making predictions on potentially successful candidates.

Data Pre-Processing:

I pre-processed the data using the steps below:

- I dropped the non essential columns from the dataset ('EIN', 'NAME')
- Finding the number of unique values for each of the columns and for columns with more than 10 unique values I had to determine the number of data points for those unique values
- Choosing to cutoff at 500 to bin together "rare" categorical values together as "Other"
- Used "pd.get_dummies" to encode the categorical data into numeric
- Separated the data into separate arrays, "IS_SUCCESSFUL" and features
- Created a testing and training set using "train_test_split"
- Scaled the training and testing features datasets by using StandartScaler then by using the transform function

Compiling, Training, and Evaluating the Model:

To obtain the best possible model I made three attempts to achieve an accuracy rating above 75% with the predictions. Using machine learning and neural networks I was only able to obtain an accuracy rating of around ~72% between each of the three attempts.

Attempt 1:

For my first attempt I set up the model to test two layers. The first layer contained 10 neurons and the second layer contained 20, both layers were activated using "relu". The model also utilized 100 epochs. Using these modifications for the first model I was able to obtain an accuracy rating of 73% which proved to be my highest result between my models.

```
268/268 - 0s - loss: 0.5527 - accuracy: 0.7300 - 489ms/epoch - 2ms/step
Loss: 0.5527034401893616, Accuracy: 0.7300291657447815
```

Attempt 2:

For the second attempt I set up the model using three hidden layers instead of two. The first layer contained 10 neurons, the second layer contained 20 neurons, and the third layer

contained 30. The model contained 100 epochs and each of the layers was once again activated using the “relu” function. Even with the use of the additional layer the accuracy decreased to 72.86%.

```
268/268 - 1s - loss: 0.5584 - accuracy: 0.7286 - 503ms/epoch - 2ms/step
Loss: 0.558414876461029, Accuracy: 0.7286297082901001
```

Attempt 3:

For my final attempt to obtain an accuracy model of 75% or higher I developed another model that contained three layers. Like the previous model it contained the same number of neurons in each of its layers. The model used 100 epochs. The change in this model came from modifying the activating functions that were used for the layers. In the previous models “relu” was the only used, for this instance I changed the “relu” activation for the second and third layer to “tanh”. Despite the additions the accuracy result for the model showed almost no change from the previous attempt with only a 72.86% accuracy.

```
268/268 - 0s - loss: 0.5532 - accuracy: 0.7286 - 479ms/epoch - 2ms/step
Loss: 0.5532208681106567, Accuracy: 0.7286297082901001
```

Summary:

Overall the three attempts that I made to create a model to predict potentially successful candidates for the Alphabet Soup funding proved to be unsuccessful with obtaining an accuracy of 75% or higher. Even with hypertuning the models and increasing the number of layers, each subsequent attempt proved to be less successful than the first model. For possible improvement on future models I would suggest implementing a model with a better correlation between the input and the output this may increase the accuracy just enough to obtain the desired 75% accuracy goal. One way to improve the correlation would be for more initial data clean up in the beginning before the models are created for teaching and testing.