Reid Hyman

7/21/2023

**Module 22 Challenge: Deep Learning**

**Alphabet Soup Charity Funding Success Predictor**

The purpose of this challenge was to design a machine learning model to assist Alphabet Soup to select applicants for funding with the best chance of success for their ventures. Deep learning and neural networks were the concepts used in this assignment. Goggle CoLab, Keras, and Tensor Flow were the tools used in this assignment.

**Preprocessing Data:**

After importing charity\_data.csv, which was cleaned prior to importing it, I preprocessed the data by taking the following steps:

* Removing columns EIN and NAME
* Binning the APPLICATION\_TYPE based on the count of the number of applications, then choosing the APPLICATION\_TYPE values that are greater than 500.
* Binning the CLASSIFICATION based on the count of the number of applications, then choosing the CLASSIFICATION values that are greater than 100.
* Converting categorical data to numeric data using pd.get\_dummies.
* Splitting the data into the features and target arrays. The target array was the column IS\_SUCCESSFUL. The features array contained the other columns in the original dataframe.
* Splitting the data into testing and training sets
* Using StandardScaler to scale the data

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

Three models were created, as shown below.

The first model had 2 layers consisting of 30 neurons in the first layer and 10 in the second.

A screenshot of a computer

Description automatically generated

The accuracy of this model was 73%. This is slightly below the 75% threshold that indicates an appropriate model.

I optimized model 1 by increasing the number of neurons to 80 for the first layer and 30 for the second layer. I named this model 2.

A screenshot of a computer

Description automatically generated

Model 2 had equal performance as model 1, with an accuracy of 73%.

To further optimize my model, I drastically increased the amount of neurons with 300 for layer 1 and 100 for layer 2.

A screenshot of a computer

Description automatically generated

This third model also failed to achieve the desired target accuracy as accuracy remained at 73%.

**Summary:**

The results of the models are summarized in the table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IPYNB File | Layer 1(Neurons) | Layer 2(Neurons) | Parameters | Accuracy |
| Starter Code (Model 1) | 30 | 10 | 1821 | 73% |
| Optimization\_1 (Model 2) | 80 | 30 | 6461 | 73% |
| Optimization\_2 (Model 3) | 300 | 100 | 45201 | 73% |

Increasing the number of neurons in a layer had no effect in obtaining the target accuracy of 75%. I cannot recommend any of the neural network models using Keras that I created. To achieve the desired accuracy, coders may have to perform trial and error using other supervised learning models such as logistic regression, decision trees, and random forest.