Timothhy L. Coleman

# Assignment 21: Deep Learning Alphabet Soup Charity Funding Success Predictor

The purpose is to design a tool to assist Alphabet Soup to select candidates with the best chance of success. Deep learning and neural networks were used. Jupyter Notebook, keras, and t-SNE are the primary tools used.

**Preprocessing Data:**

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. With your knowledge of machine learning and neural networks, you’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, you 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

**Compile, Train, and Evaluate the Model**

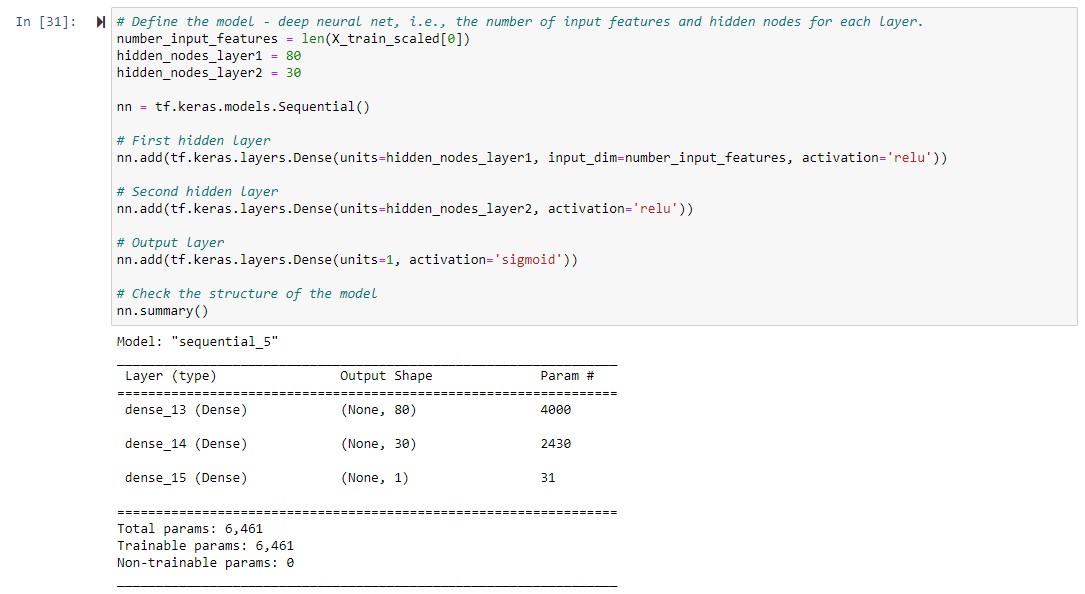
Four models were created. The first model had 2 layers, 30 neurons in the first layer and 10 in the second.

A screenshot of a computer

Description automatically generated

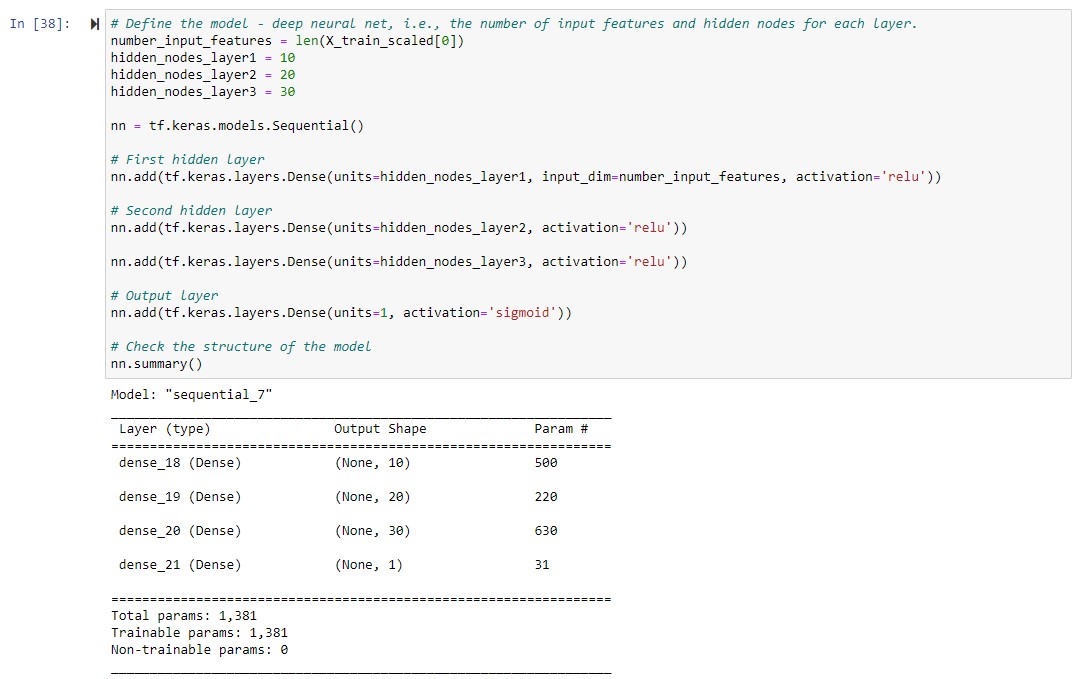
The accuracy of this model was 73.65 and under the 75% barrier.

The next model also had 2 layers, but the number of neurons was increased to 80 for the first layer and 30 for the second.

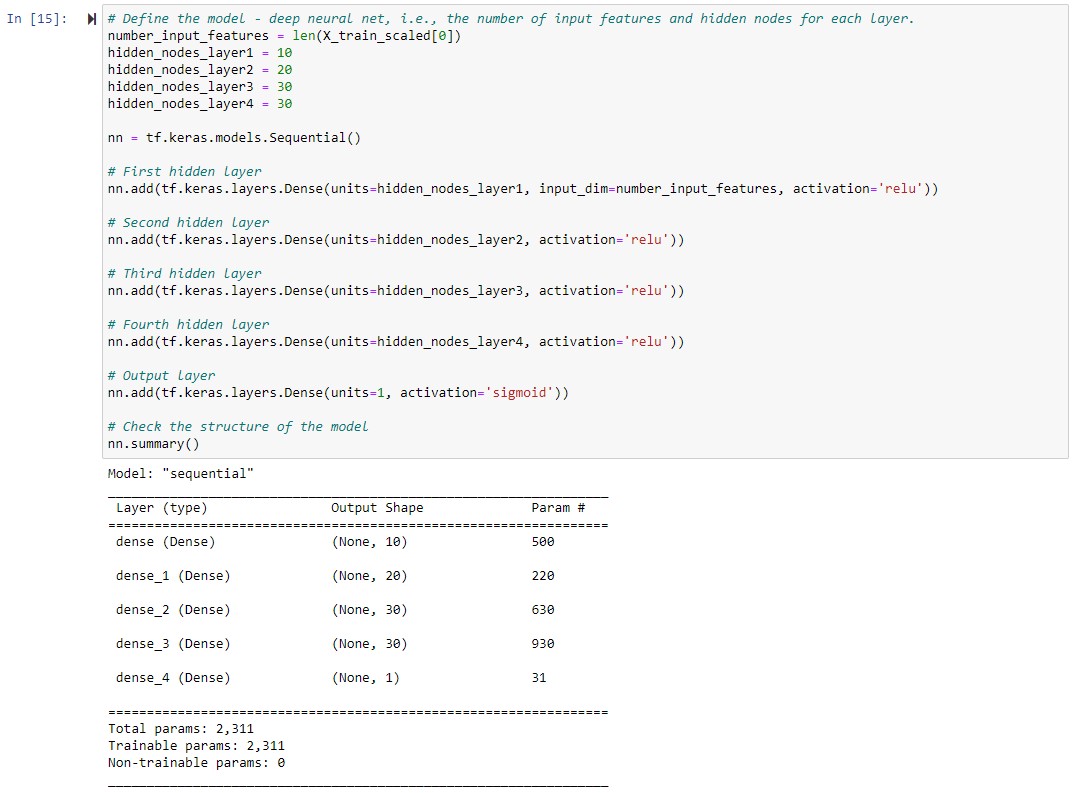


This model did not perform any better or worse, it also had an accuracy of 73.0%.

The number of layers was increased for the third model. The model consisted of 3 layers, 10 neurons for the first layer, 20 for the second, and 30 for the third.



The last model increased the number of layers to 4, 10 neurons for the first, 20 for the second and 30 for both the third and fourth layers.



This model was slightly worse than the other models with an accuracy of 72.7%.

**Summary:**

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IPYNB File | Layer 1 (Neurons) | Layer 2 (Neurons) | Layer 3 (Neurons) | Layer 4 (Neurons) | Parameters | Accuracy |
| Starter Code | 30 | 10 | N/A | N/A | 1821 | 73.0% |
| Optimization\_1 | 80 | 30 | N/A | N/A | 6461 | 73.0% |
| Optimization\_2 | 10 | 20 | 30 | N/A | 1381 | 73.0% |
| Optimization\_3 | 10 | 20 | 30 | 30 | 2311 | 72.7% |

Increasing the layers and adjusting the neurons had little effect in obtaining the target accuracy of 75%.