**USING NEURAL NETWORK TO PREDICT VIEWER’S FEELING TOWARDS A YOUTUBER’S STYLE THROUGH FOOD DESCRIPTION (Part 2)**

**The current problems:**

* The previous model with 1 hidden layer of 300 nodes resulted in a 45% accuracy in the training set and 44% accuracy in the test set.

**The solution:**

* Since the previous model doesn’t show any signs of overfitting (train – test accuracy is quite close), the model’s complexity will be increased to find a way to increase the model’s performance on predictions.
* After increasing the model’s complexity, if its predictions show signs of overfitting, some squared features will be added.
* Since the output of this problem is multi-class, it might be correct to implement a softmax activation function for the last layer.

**Summary of the report and conclusion:** (detailed report below)

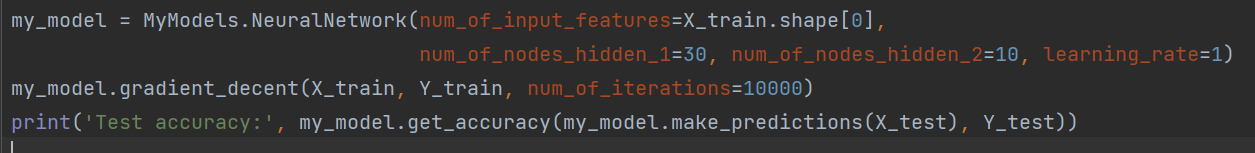
* Increasing the complexity of the model did help with the test set accuracy (41%). However, with more iterations, that model overfits itself.
* After adding the squared feautres, the model no longer overfits, and the train set accuracy improved from 41 to 43%. No changes in test set accuracy.
* Softmax activation function did help the model with both training and predicting the data (46% and 43% respectively). However, using squared features heavily overfits the model.

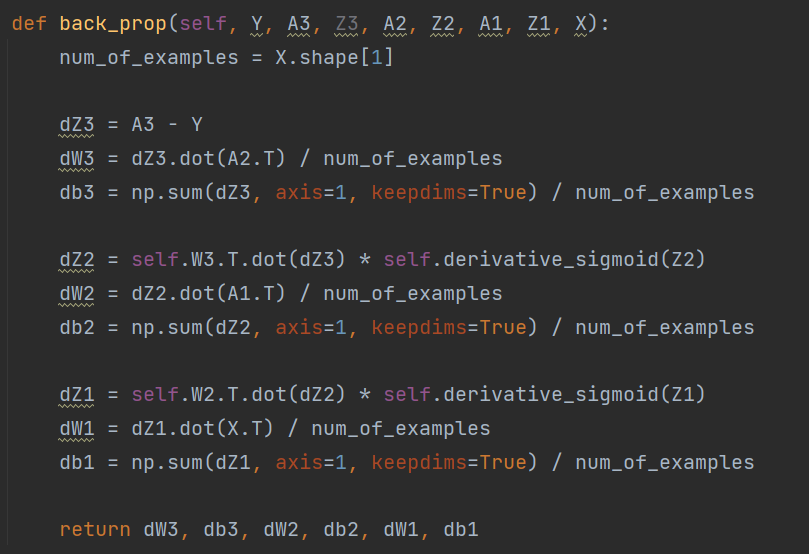
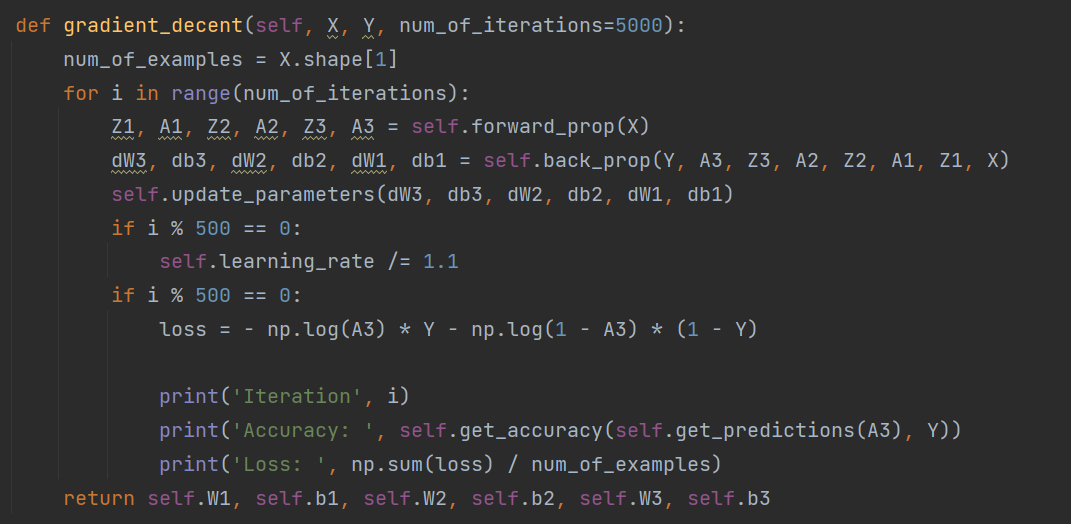
Link to the codes: [Google Colab](https://colab.research.google.com/drive/11Kkgjs_T98NQs9WG3VmnebnozhG-9wWt?usp=sharing)

**Detailed report:**

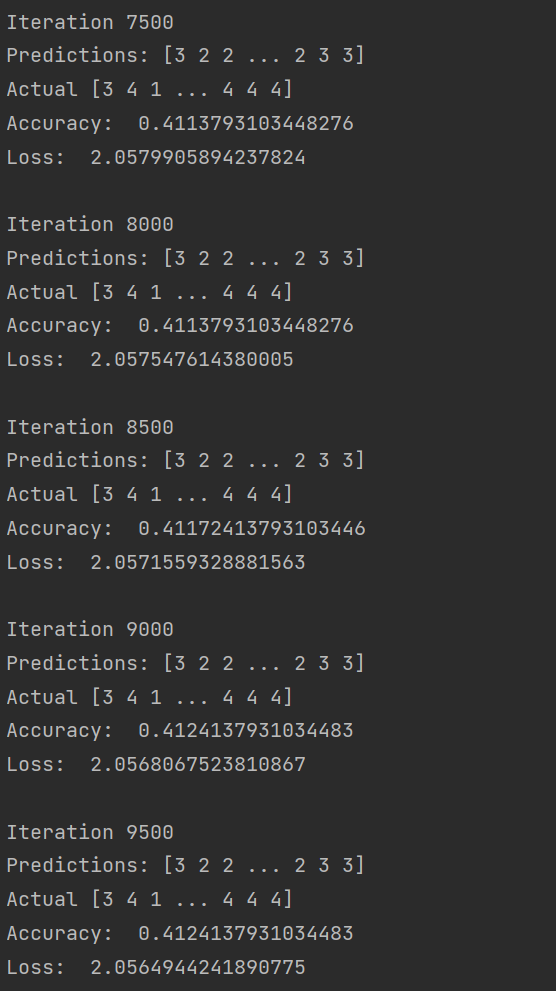
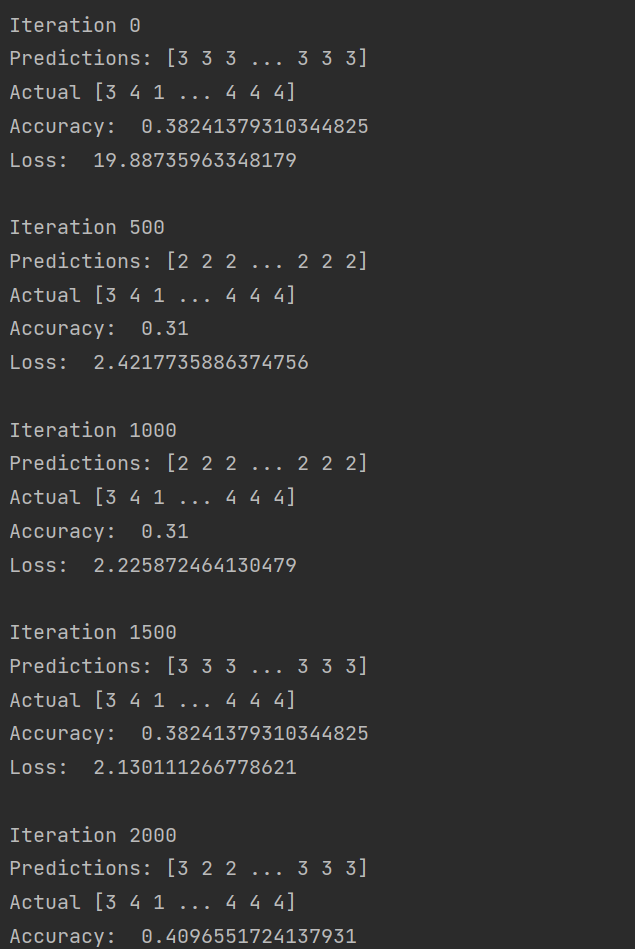
**Increasing the model’s complexity:**

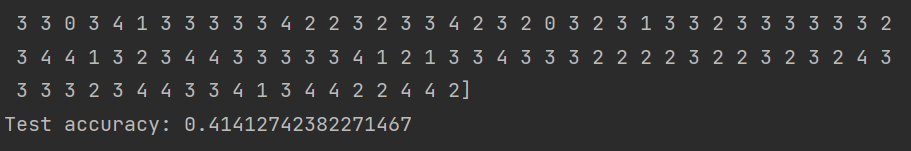
* Previously, we were stopping at a neural network with 300 nodes and 10000 iterations over the training set.
* Now, let’s increase the number of hidden layers, and set the nodes to 30 and 10 respectively.



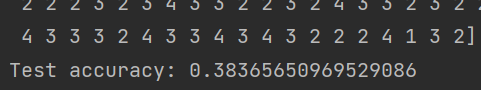
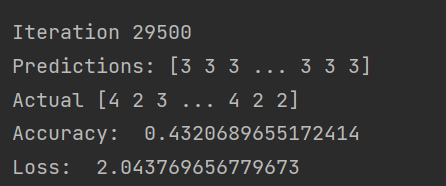


* The result:





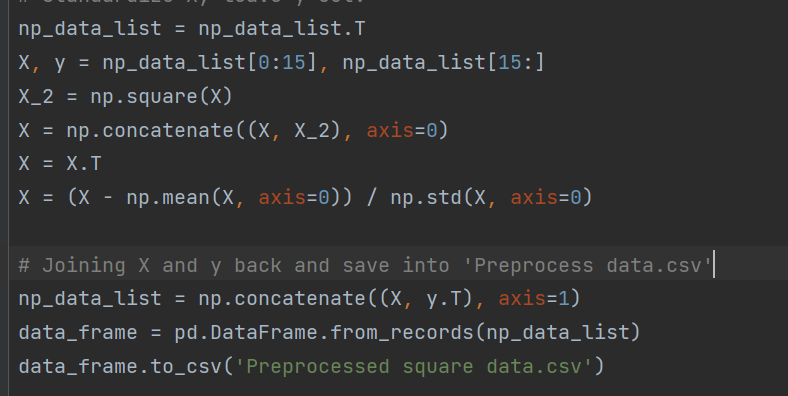
* Even though the accuracy of the training set is lower, the test set accuracy is improving (40% on the previous model).
* Let’s see if more iterations result in a better model: 30000 iterations.



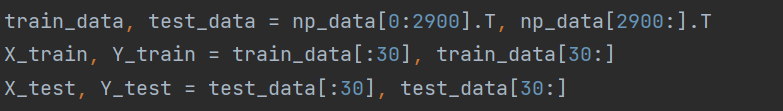
* The result isn’t getting better, it’s overfitting. Therefore, we move on to increasing the number of features.

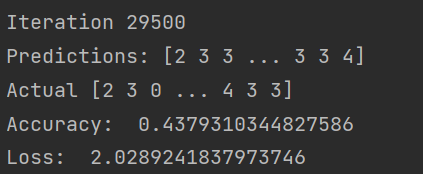
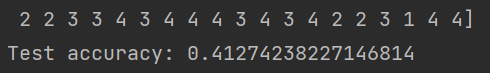
**Adding the squared features:**

* We have 15 inputs and now with the model overfitting itself, let’s add the square of all 15 features and save it to ‘Preprocess square data.csv’.



* Now, each of our examples has 30 features to lead to the viewer’s feeling. Let’s see how our model performs.

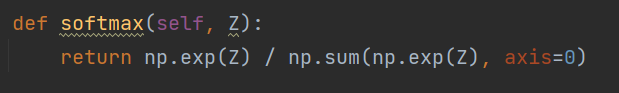


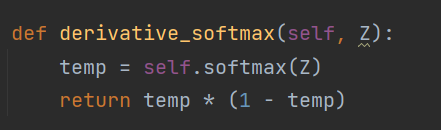
 

* The test set accuracy is actually increasing, so this is a good sign.

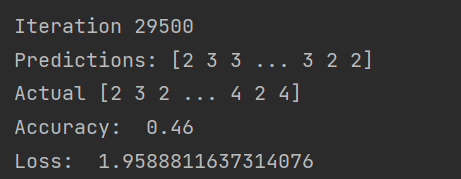
**Softmax activation function implementation:**

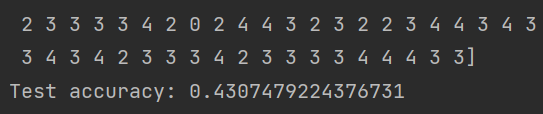
* This is the implementation of the softmax activation function:



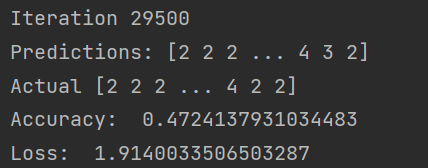


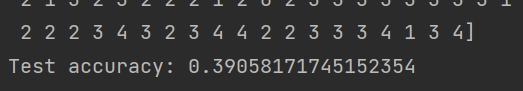
* The rest is the same, since both softmax and sigmoid have the same derivative fomular.
* The result of softmax model using 15 original features:





* The result of softmax model using the additional 15 squared features:





* **Softmax activation function improved the model upto 3% of test set accuracy comparing to the sigmoid function. However, squared features worsen the model.**