

**CSCE4604: Practical deep machine Learning**

**Assignment 2- Report**

|  |  |
| --- | --- |
| **Submitted by:** | **Mohammed Abuelwafa** |
| **Student ID:** | **900172603** |

|  |  |
| --- | --- |
| **Date:** | 11th November 2020 |

**Fall 2020**

# Data Preprocessing:

* The following was done while preprocessing the data for the training :

1. Subtract the mean image from the data set   
   X -= np.mean (X , axis = 0 )
2. Divide by 255.0 in order to normalize the pixel values to be between 0 and 1  
   X /= 255.0
3. Before calculating the loss, the largest score was subtracted from the scores array in order to avoid numerical explosion when calculating the NLL loss.

scores -= np.max(scores) # to avoid numerical blowup

# Choosing the network Architecure:

* A two hidden layer fully connected network was selected since two hidden layers add enough degree for non-linearity for this problem and similar problems.
* I started with learning rate = 1 , regularization = 1e-3 and run a fine search in the search space to find the most suitable values for both while monitoring the training loss , validation loss , training accuracy and validation accuracy to avoid overfitting and to determine when to stop the training.
* SGD was used and the weights were updated after each epoch ( minibatch gradient decent and other flavors would give better results but I didn’t have enough time to implement )
* The ReLU activation function was used in addition to keeping a close eye on the learning rate.

Best training had a value of : learning rate = 5e-2 , regularization = 1e-3

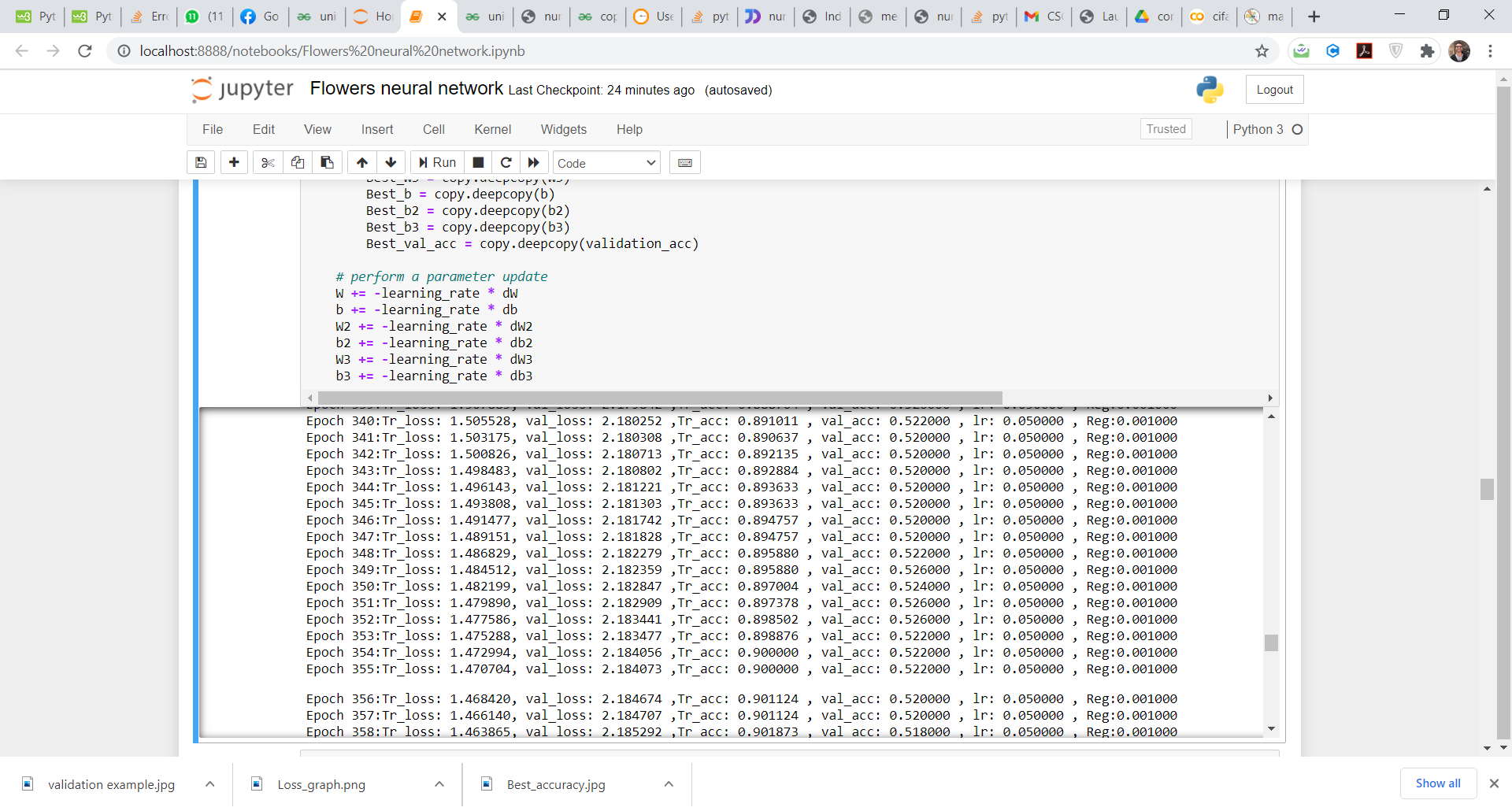
Some numerical examples:

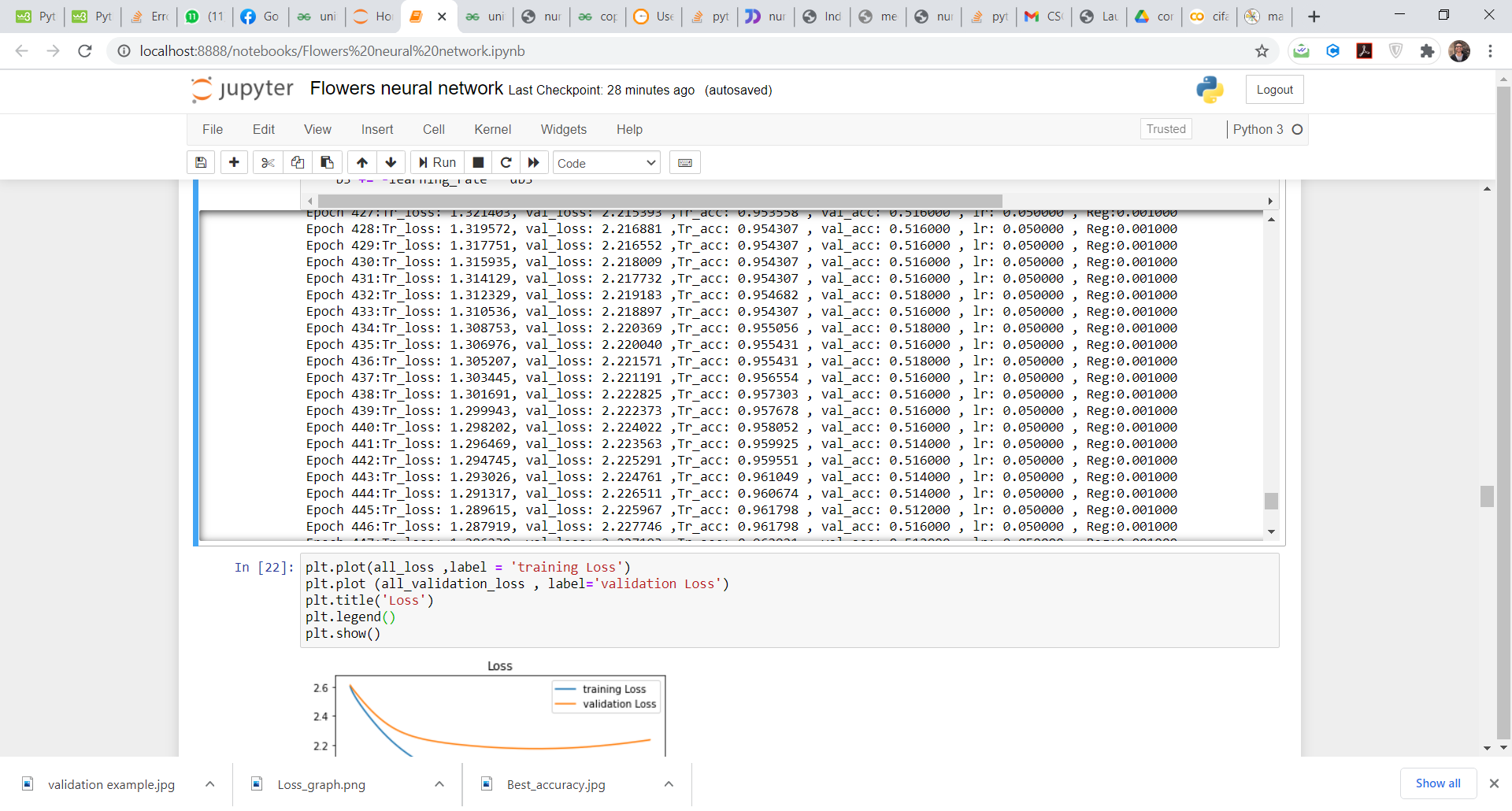
* Very high learning rate cause such errors in the validation accuracy causing it to saturate at 20%

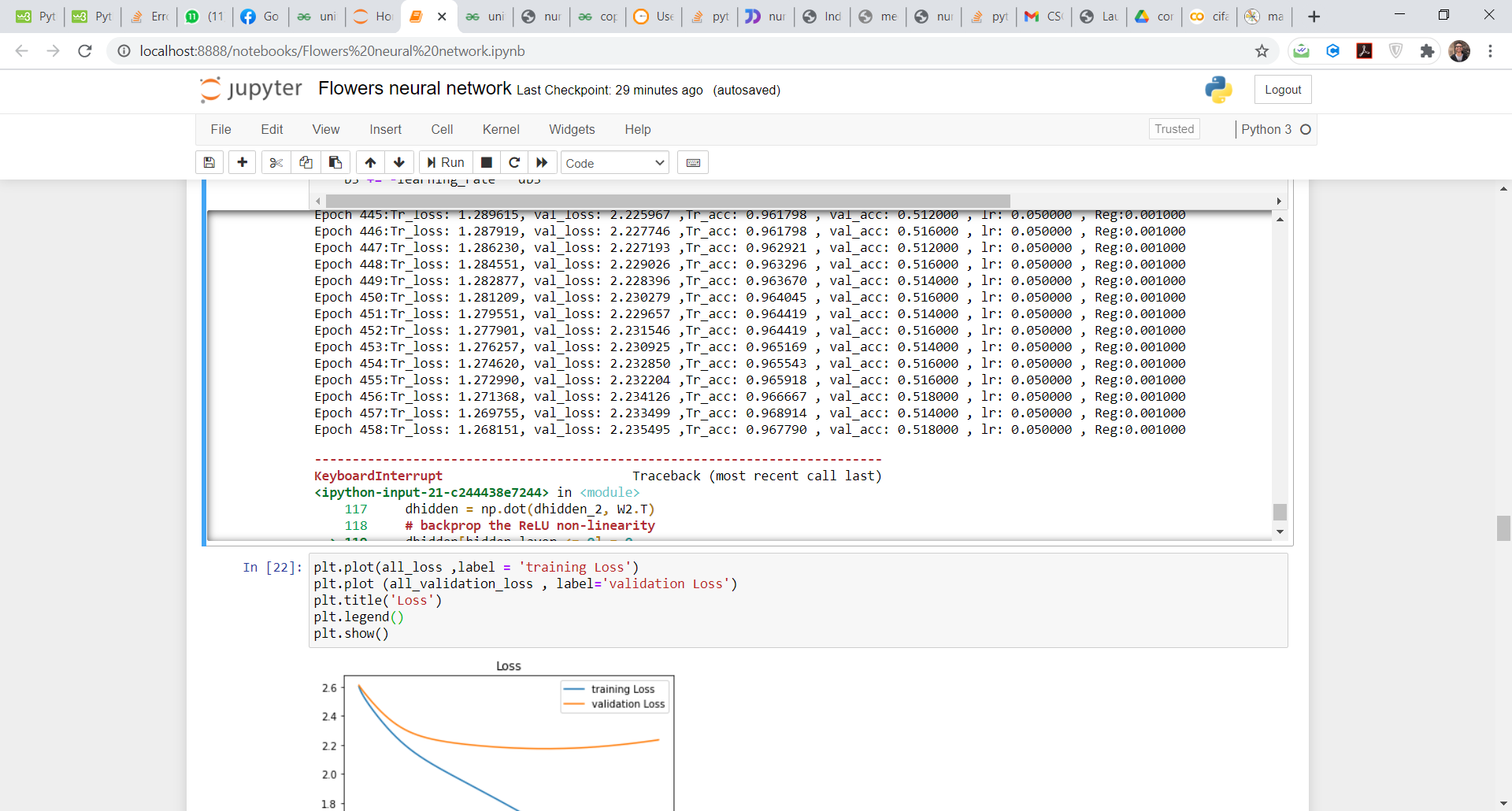
Table

Description automatically generated

* Best Training Progress:



Stopping the training point :



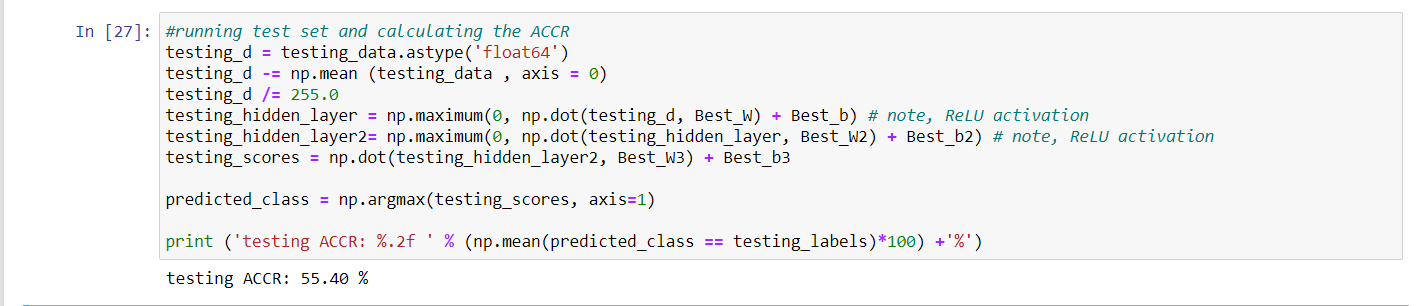
* The sign of overfitting occurred where the training loss was decreasing whereas the validation loss was increasing. The best weights were saved then and was used on the testing set.

Best Training Loss vs epochs Graph:

A picture containing line chart

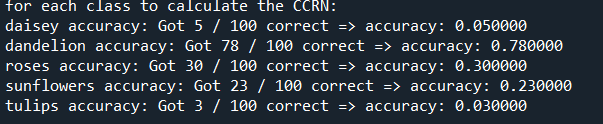
Description automatically generated

# Best ACCR:

An Accuracy of : 55.4% was obtained. (in comparison to 27.8% in the KNN Classifier)

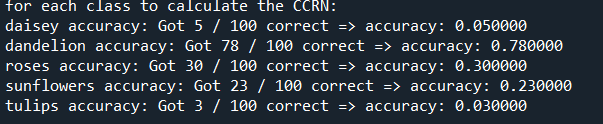
# Correct classification rate (CCRn) comparison:

## For Daisy:

KNN : 5%

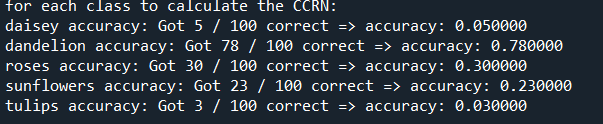
Neural network: 41%

## For Dandelion:

KNN : 78%

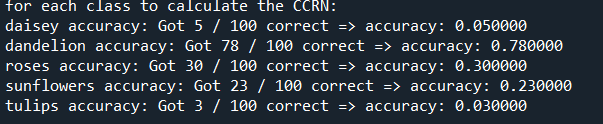
Neural network: 70%

## For Sunflowers:

KNN : 23%

Neural network: 71 %

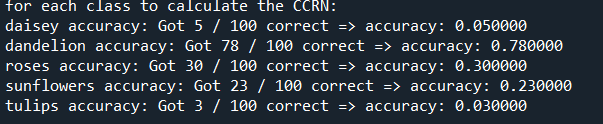
## For Roses:

KNN : 30%

Neural network: 38%



## For Tulips:

  
KNN : 3%

Neural network: 57%

The neural network generally performed better than the KNN classifier in both the ACCR and the CCRn.