**Department of Electronic and Telecommunication Engineering**

**University of Moratuwa, Sri Lanka**

**EN2550 - Fundamentals of Image Processing and Machine Vision**



**ASSIGMENT 4**

**Submitted By**

Nagasinghe K.R.Y 180411K

**Submitted on**

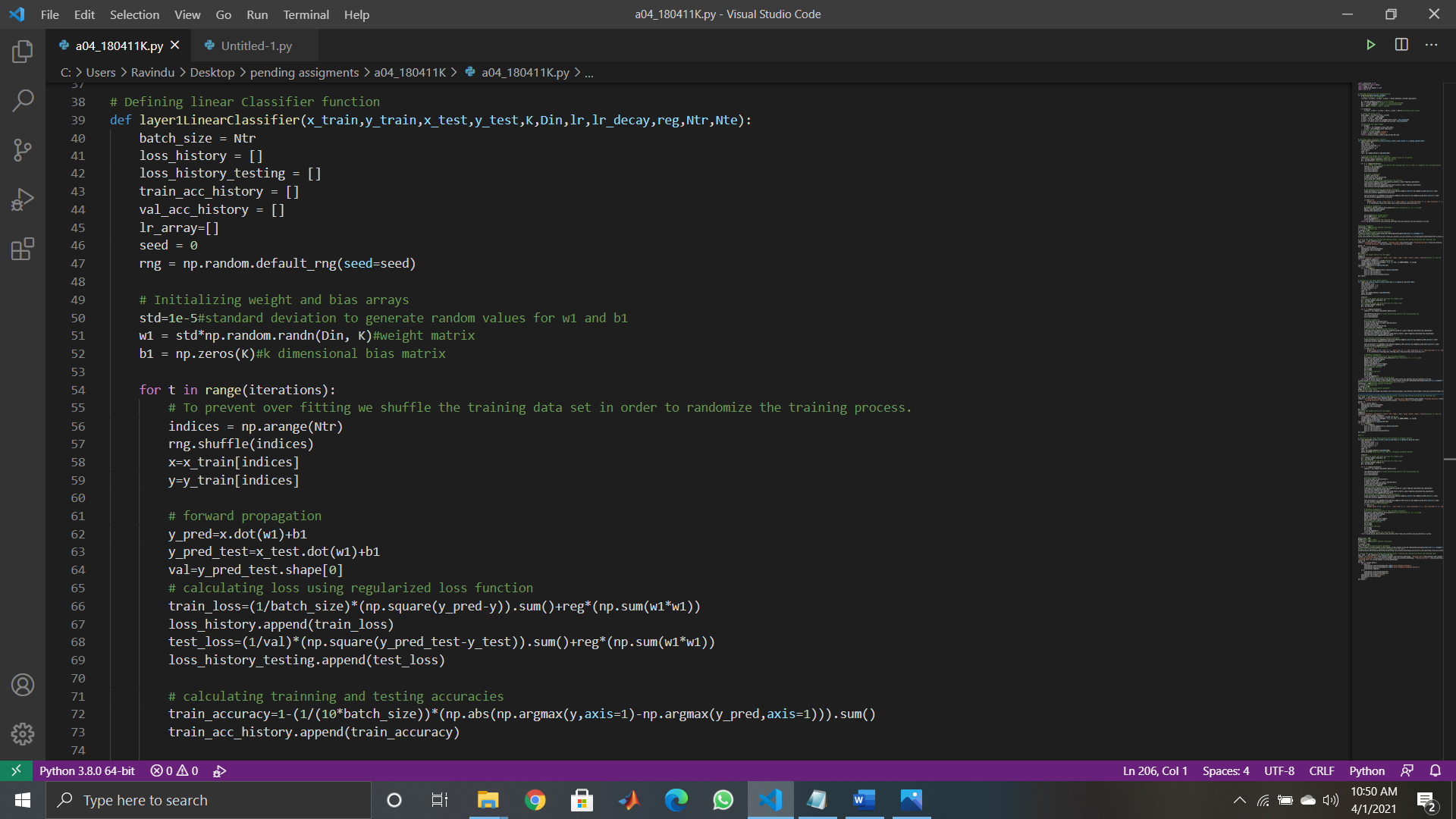
**April 2, 2021**

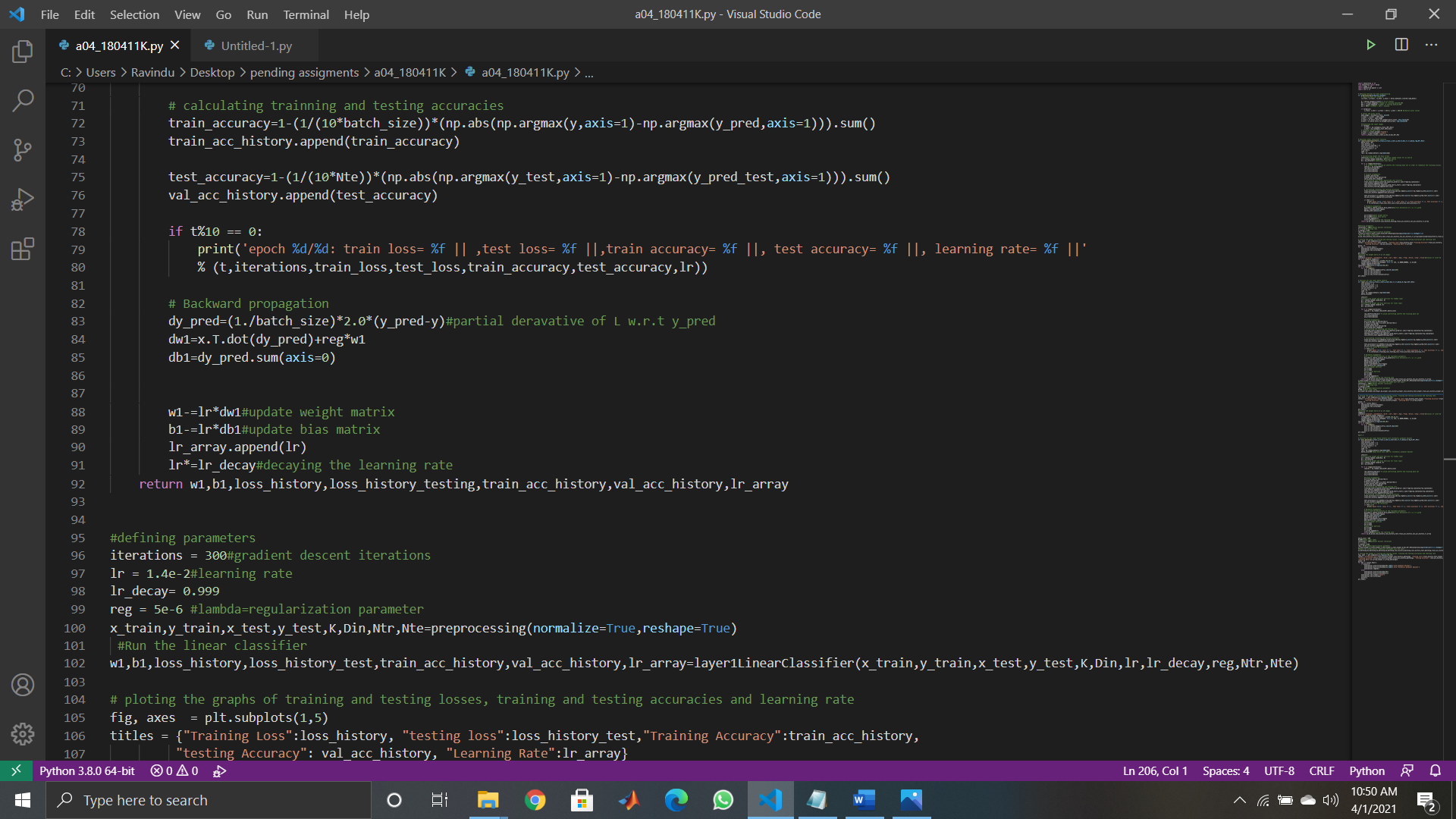
Full code: <https://github.com/Ravindu-Yasas-Nagasinghe/EN2550-Computer-Vision-and-Image-Processing-Assigments>

**1)Linear Classification using gradient descent.**

Text

Description automatically generatedHere our data set is CIFAR-10. There are 10 different classes in this data set. I use tensorflow to import the data set to python. Our score function for the linear classifier is f (x) = W x +b, and the loss function is the mean sum of squared errors function. I run for 300 epochs as instructed in the assignment. The code for 1layer linear classifier using gradient descent is as follows.





A picture containing text, gallery, room, colorful

Description automatically generatedAfter that I plot the weight matrix W, as 10 images and plot the training loss, testing loss, training accuracy, testing accuracy and learning rate.

Weight matrix as 10 images

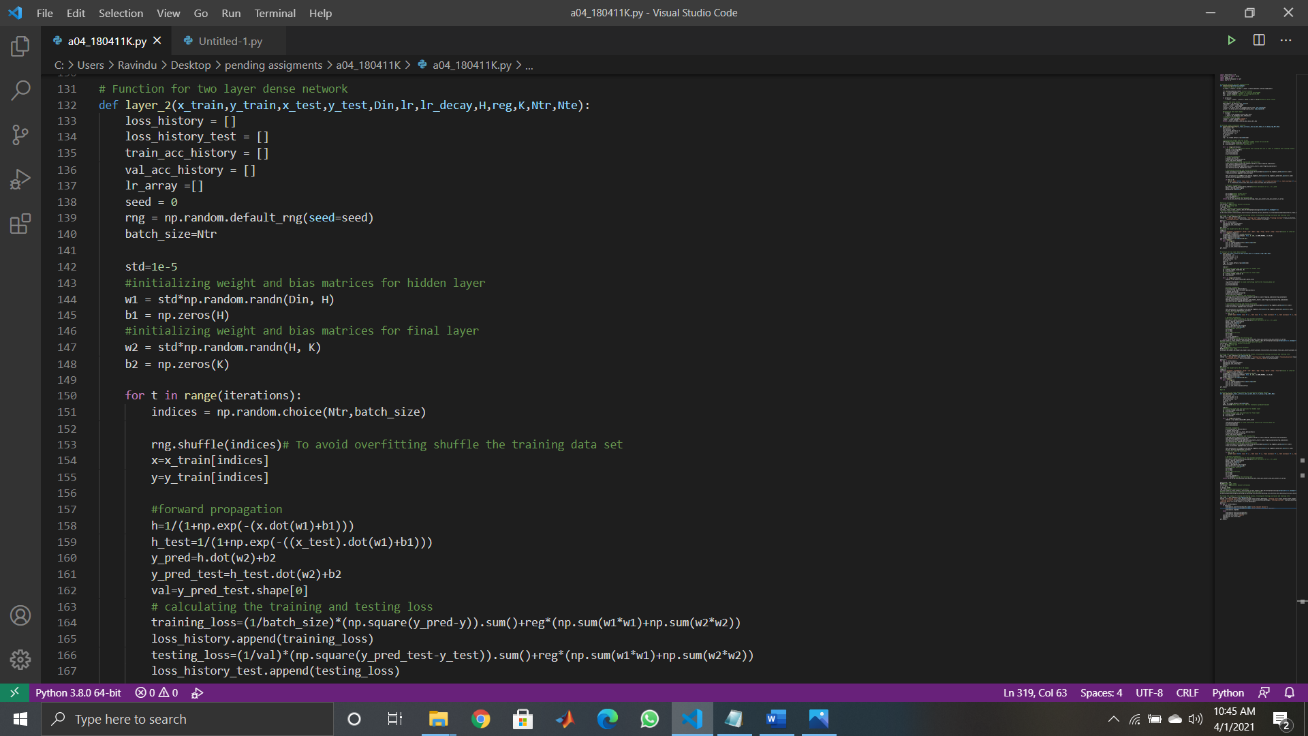
A picture containing histogram

Description automatically generatedInitial learning rate = 1.4 x 10

Loss, testing loss, training accuracy, testing accuracy, learning rate of the linear classifier for 300 epochs.

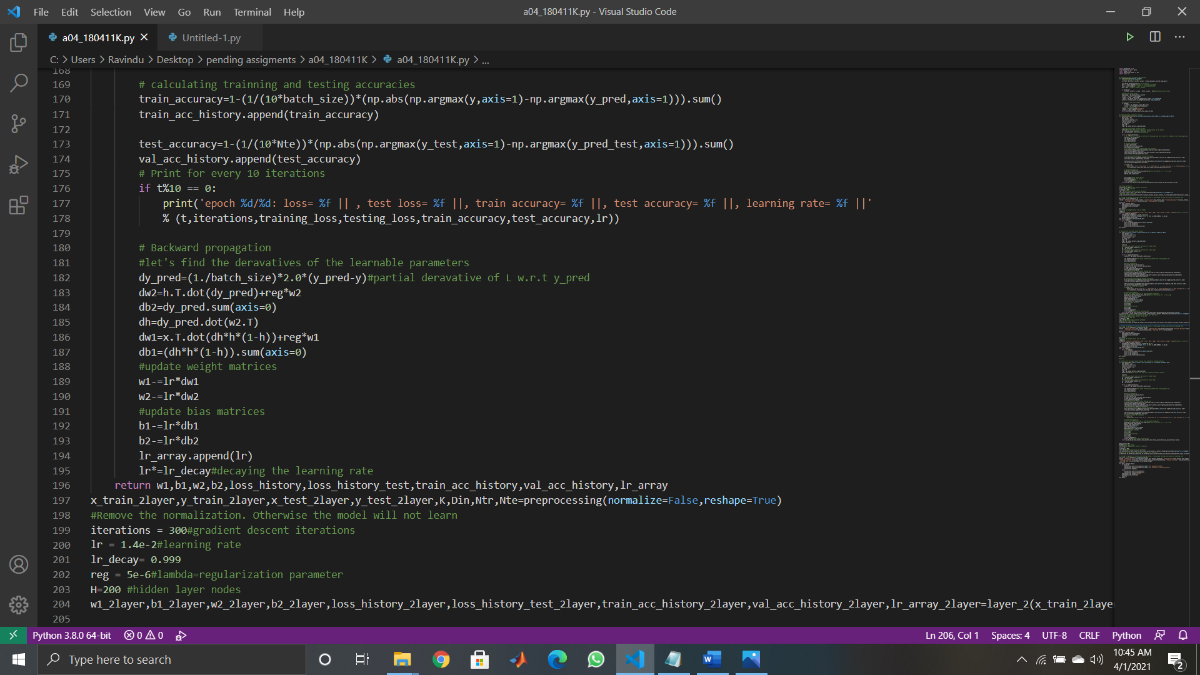
After 300 epochs train loss= 0.783117, test loss= 0.787730, train accuracy= 0.779958, test accuracy= 0.774120, learning rate= 0.010474.

**2) 2 layer fully connected network**

A computer screen capture

Description automatically generated with low confidenceA computer screen capture

Description automatically generated with low confidenceHere I use a two-layer dense network with H=200 hidden nodes. Code for this part is as follows.



Here gradients are computed in the direction from output to input layers and combined using chain rule.

* Input layer to hidden layer
* Hidden layer to the output layer
* Total number of learnable parameters in the network = (200 x 3072+200) + (10 x 200+10) = 616,610

As we can see from the below results when the number of iterations increase the loss decreases and accuracy increases. But the rate of the loss decreasing and accuracy increasing reduces with the iterations.

As we can see from the results, when we use 2 layer fully connected network instead of single layer as in part 1, we can reduce the train and test loss and increase training and testing accuracy. So, if we increase the number of layers further, we can achieve more and more accuracy and reduce loss.

Chart, histogram

Description automatically generatedInitial learning rate = 1.4 x 102

Loss, testing loss, training accuracy, testing accuracy, learning rate of the 2 layer fully connected network for 300 epochs.

After 300 epochs, loss= 0.735866, test loss= 0.759436, train accuracy= 0.795186, test accuracy= 0.784170 and learning rate= 0.01047|.

**3) Stochastic gradient descent with a batch size of 500**.

**A screenshot of a computer

Description automatically generated with medium confidence**Here instead of using 50,000 samples I only uses 500 samples which are selected randomly. The code for this part is as follows. (only the parts different from the code in part 2 is included. The full code is available in the link at cover page).

**A picture containing text, screenshot, monitor, computer

Description automatically generatedText

Description automatically generated**

Chart, histogram

Description automatically generatedLoss, testing loss, training accuracy, testing accuracy, learning rate of the 2 layer fully connected network using gradient descent and stochastic gradient descent for 300 iterations.

After 300 iterations for stochastic gradient descent with batch size =500, loss= 0.737830, test loss= 0.765587, train accuracy= 0.808400, test accuracy= 0.783190 and learning rate = 0.010474.