Deep Learning HW1

106061703 藍國瑞

1. Regression:

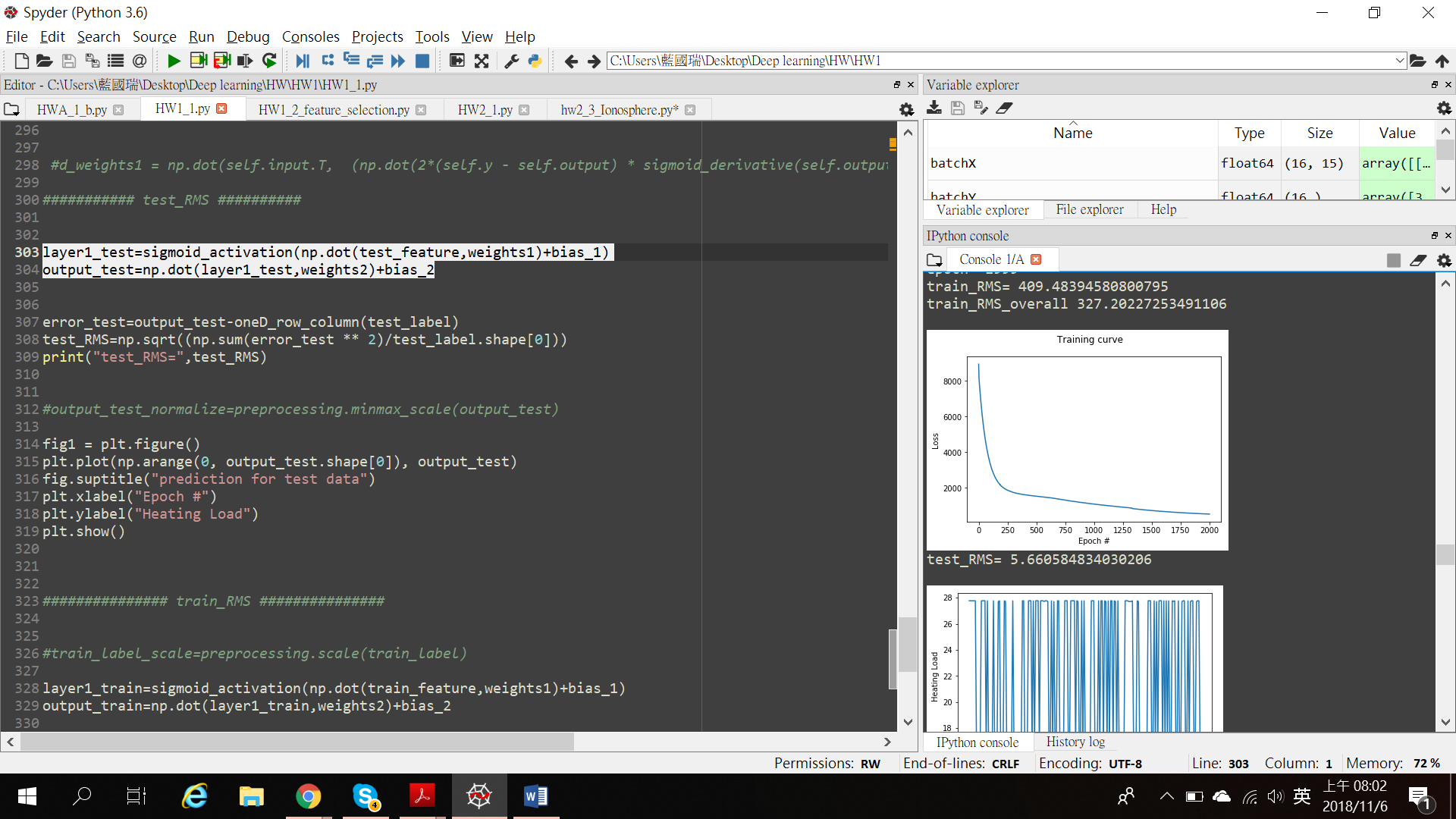
Learning rate= 0.000001

Number of epochs=2000

Mini\_batch size=16

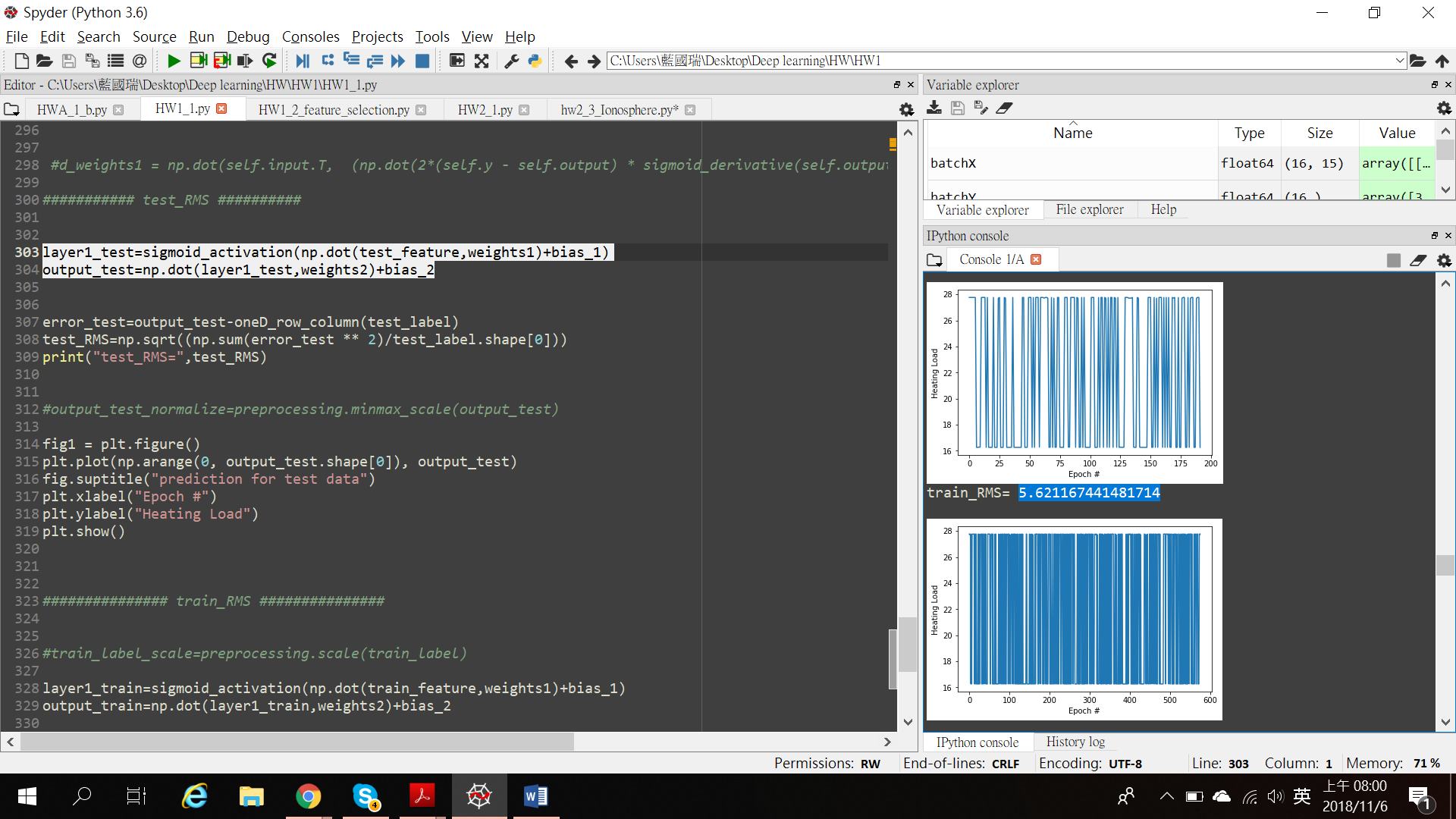
b. (1) number of hidden layer:1, number of neurons:12

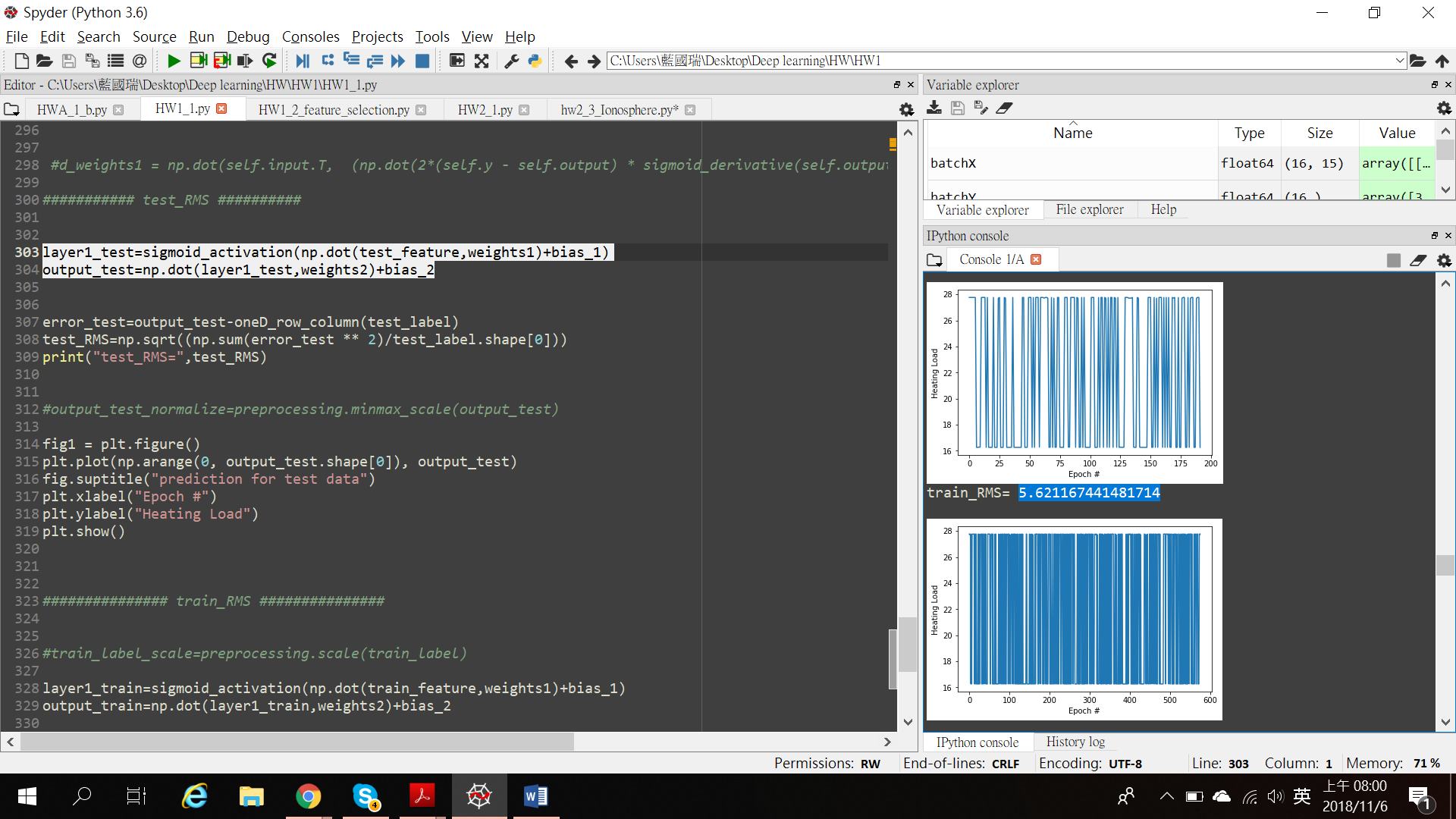
(2) learning curve:



(3) training RMS: 5.621167441481714

(4) test RMS: 5.660584834030206

(5) regression with training labels: 

(6) regression with test labels: 

c. 首先我先假設這個神經網路實際上是和7個feature最有關係，接著再用排列組合在總共8個feature中，去挑選當中的7個，這樣的可能性就會有C8 7種，並且把每種可能性的feature當作神經網路的輸入去訓練，最後可以算出TEST RMS和 TRAIN RMS，並且會發現有些TEST RMS會比較小，這樣就可以反推回去知道是那些feature對這個網路的影響比較重要。

Ex: 以下是我假設有七種feature對該網路最有關係的可能性，當然也可以假設有四種、六種、五種對該網路最有關係，這邊只是以七種當作舉例，因為組合數有點多，所以我的epoch變成500:

|  |  |  |
| --- | --- | --- |
| Feature combinations | TRAIN\_RMS | TEST\_RMS |
| (0, 1, 2, 3, 4, 5, 6), | 9.99204643456245, | [10.426207263247182, |
| (0, 1, 2, 3, 4, 5, 7), | 9.96057753988234, | 10.463071008588653, |
| (0, 1, 2, 3, 4, 6, 7), | 9.998802705535875, | 10.339597730170135, |
| (0, 1, 2, 3, 5, 6, 7), | 10.165236398743737, | 9.852481632060469, |
| (0, 1, 2, 4, 5, 6, 7), | 9.255983047953801, | 9.4179086771627, |
| (0, 1, 3, 4, 5, 6, 7), | 10.201982596978027, | 9.720631238935269, |
| (0, 2, 3, 4, 5, 6, 7), | 5.1224113138048155, | 5.274300109735227, |
| (1, 2, 3, 4, 5, 6, 7) | 10.099897608389446] | 10.061407179240687] |

由表格當中，我們可以看到當選取不同FEATURE去訓練時，會有不同的TRAIN RMS 和 TEST RMS，並且可以發現當選取的FEATURE是(0,2,3,4,5,6,7) 時，他的TEST RMS和TRAIN RMS明顯小於其他的FEATURE 組合，因此可以推知(0,2,3,4,5,6,7)的組合對於該網路的影響比較重要。

2. Classification:

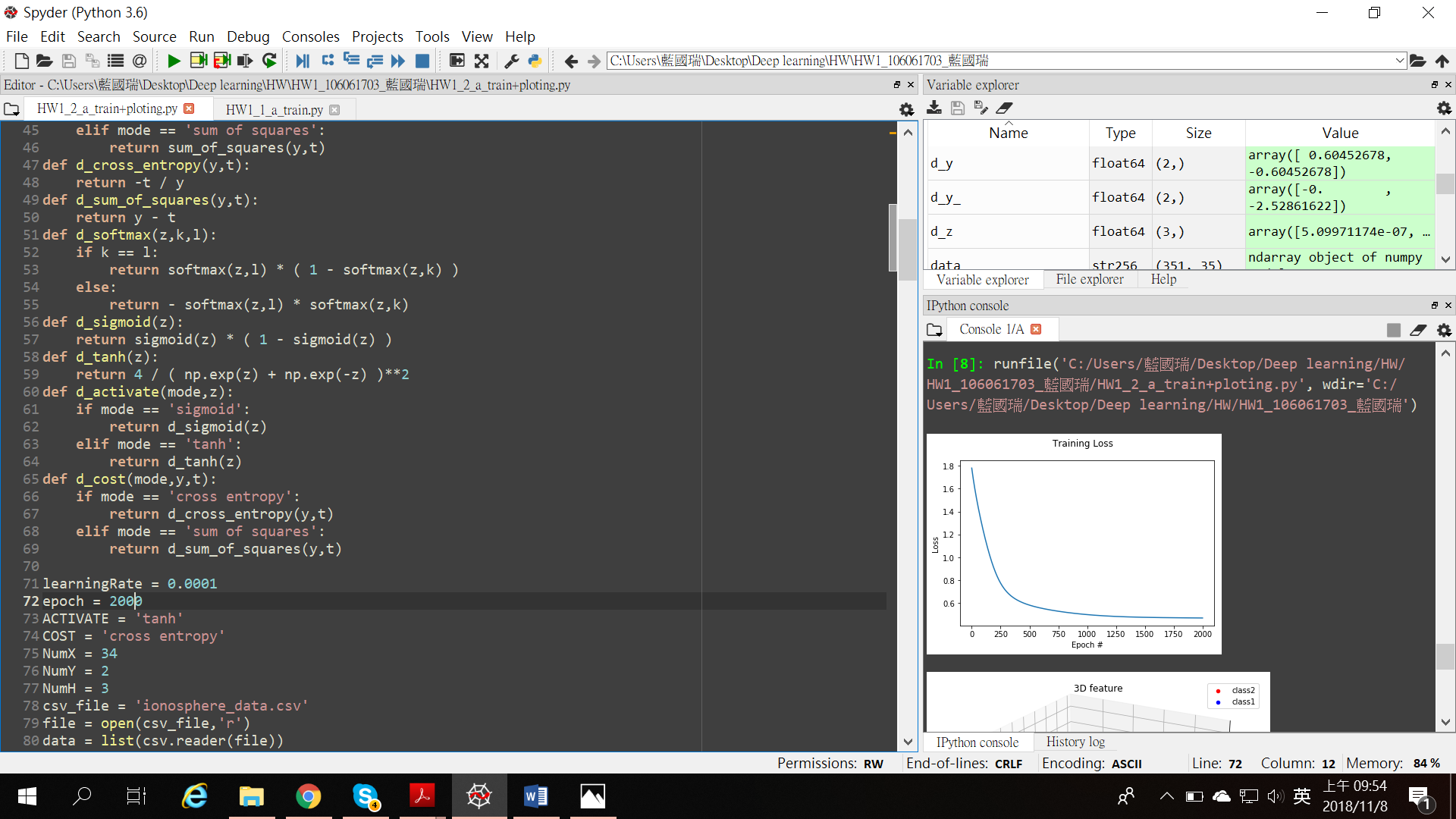
Learning rate= 0.0001

Number of epochs=1000

Mini\_batch size=16

b. (1) number of hidden layer:1, number of neurons:10

(2) learning curve:

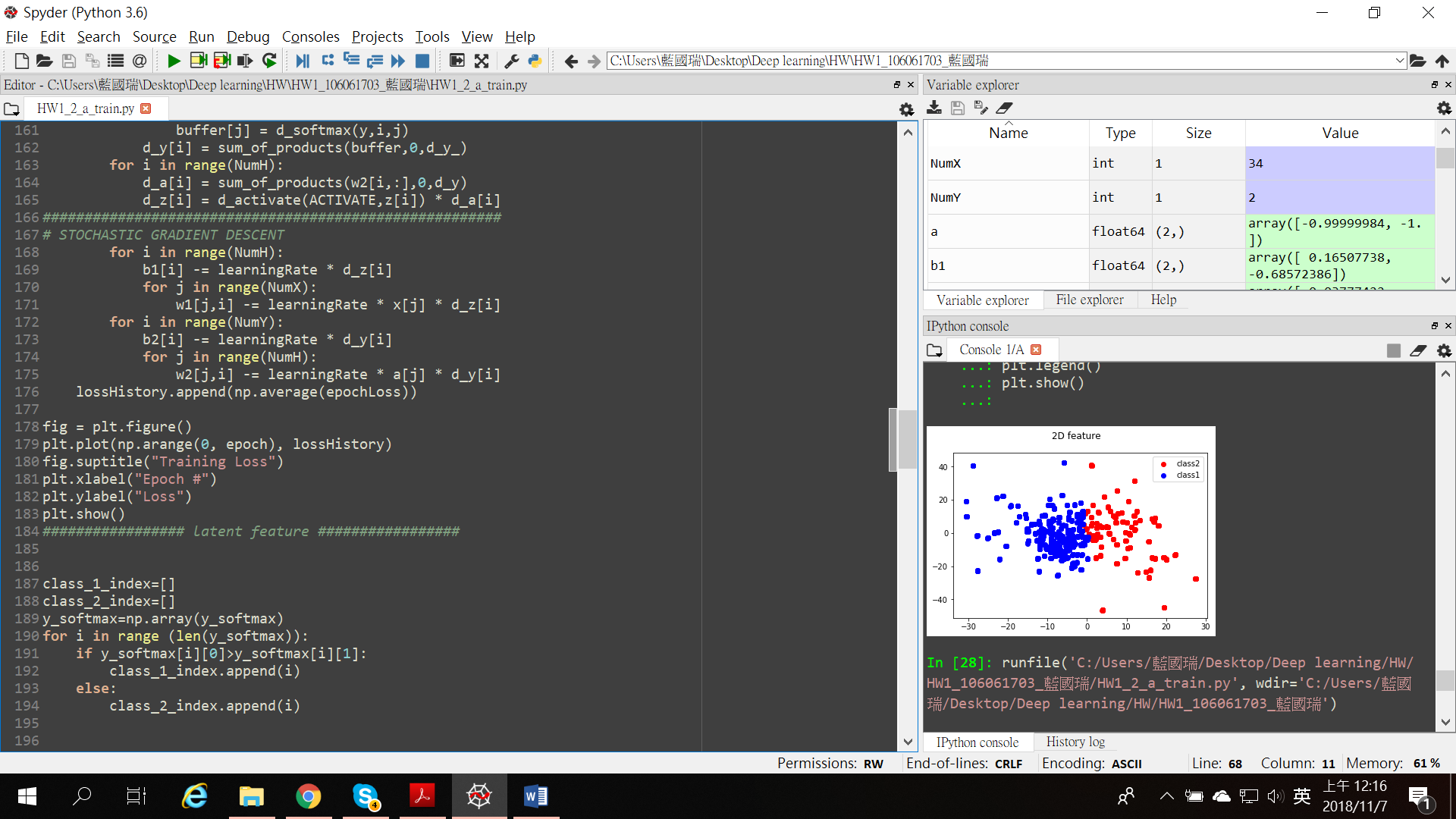


(3) training error rate: 0.08571428571428572

(4) test error rate: 0.352112676056338

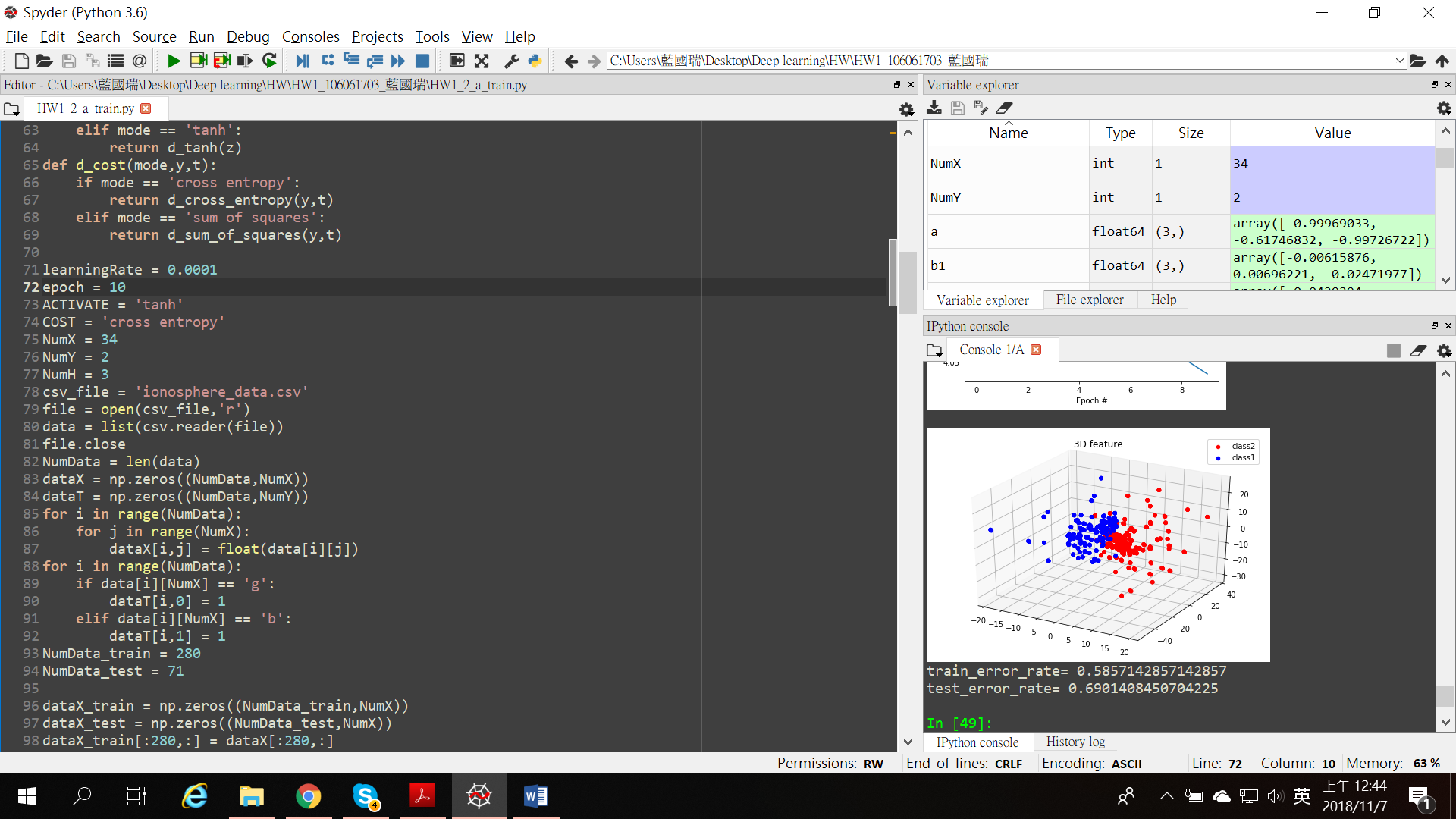
c. latent feature:

**2D feature 10 epoch:**



**可以明顯看到class1和class2分成兩區塊。**

**3D feature 10 epoch:**



**可以明顯看到class1和class2也分成兩區塊。**