AI HW1

41047025S 王重鈞

檔案說明

HW1.ipynb是我用來方便訓練AI模型的notebook
我最後將三個神經元的模型程式碼都放在run.py裡
Data.csv是成績的資料

1.

輸入:

> 抽問分數、作業一二三四、作業平均、測驗一二三四、測驗平均，共10個channels

輸出:

> 總成績，共一個channel

訓練資料:

> 我將所有的data作為訓練資料，一共14筆

測試資料:

> 我用最後一筆數值作為測試資料，一共1筆

預處理:

> 為了使模型訓練不會因精度而有誤差，我將所有資料除以一個scale(=150)，並輸出解答時乘回scale

訓練次數:

> 多為100次

使用python版本:

> 3.10.6

使用套件:

> 1. pandas 用於讀入csv檔及分離解答
> 2. numpy 用於矩陣運算
> 3. matplotlib 用於繪製loss折線圖
> 4. 額外輔助：利用notebook方便分段訓練

硬體設備作業系統:

> MacOS 記憶體8GB

為何使用該規格:

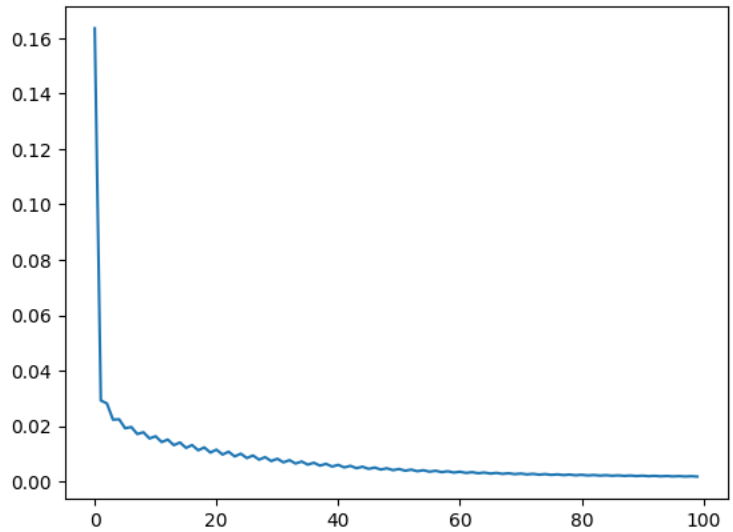
手邊僅有如此設備

聯絡電話:

0905866533

2.

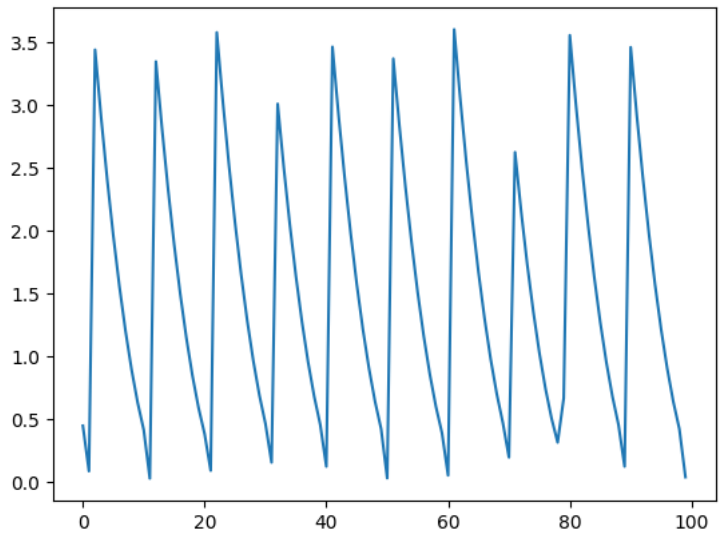
訓練資料的MSE:
經過100epoch訓練後，MSEloss停在
0.0017297795872850037



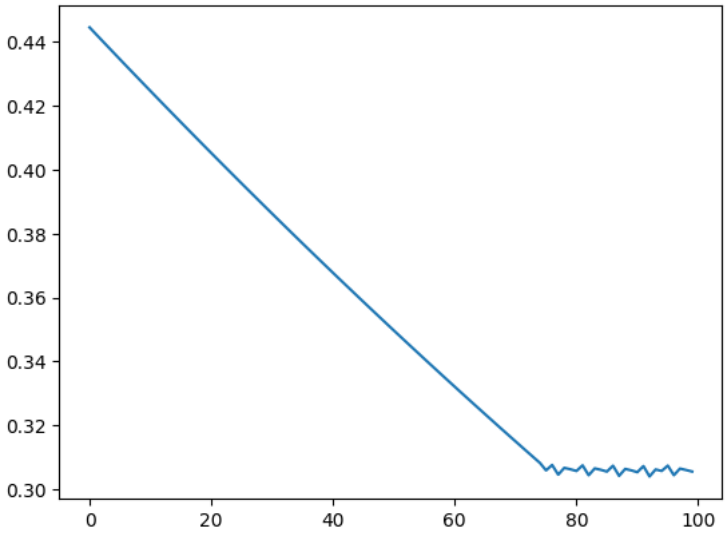
測試資料的MSE:
0.035156479147562834

3.

我嘗試使用了 leakly ReLU 作為激活函數，產生這樣的loss情況

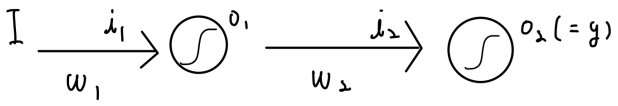


我猜測是因為 learning rate 太大，導致每次都跳過了最低點，因此我將
learning rate 設為 0.01 得到這樣的結果



但測試資料的 MSEloss 也有0.3054781713384966，就結果而言似乎沒有原先的好

4.

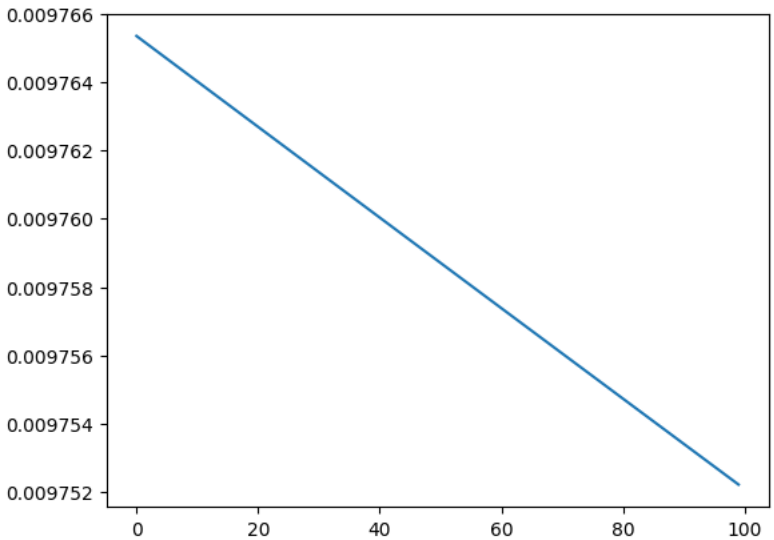


$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial o_2} \times \frac{\partial o_2}{\partial i_2} \times \frac{\partial i_2}{\partial w_2} = -(t - o_2) \times o_2 (1 - o_2) \times o_1$$

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial o_2} \times \frac{\partial o_2}{\partial i_2} \times \frac{\partial i_2}{\partial o_1} \times \frac{\partial o_1}{\partial i_1} \times \frac{\partial i_1}{\partial w_1} = -(t - o_2) \times o_2 (1 - o_2) \times w_2 \times o_1 (1 - o_1) \times I$$

5.

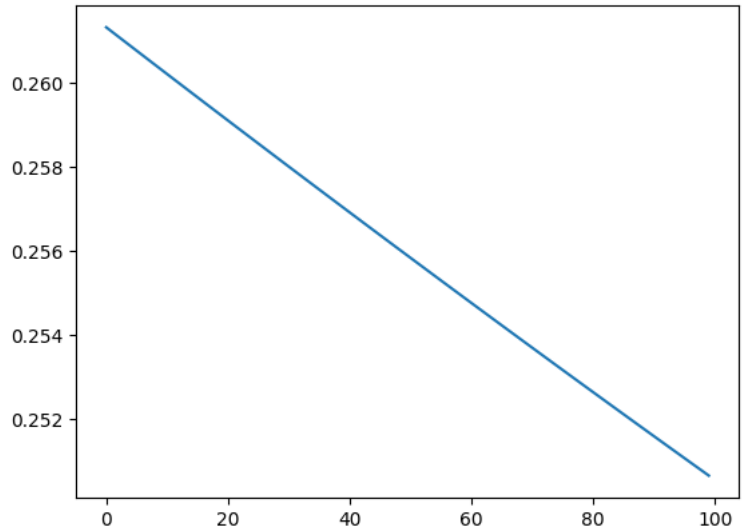
我利用上圖架構以及 sigmoid 作為激活函數，並將 learning rate 設為 0.01，產生這樣的 loss



訓練 loss 可達 0.00975221597756065
測試 loss 也有0.28203427954421606，實驗數據沒有單一神經元來得好，或許可能有過擬合的情況發生？

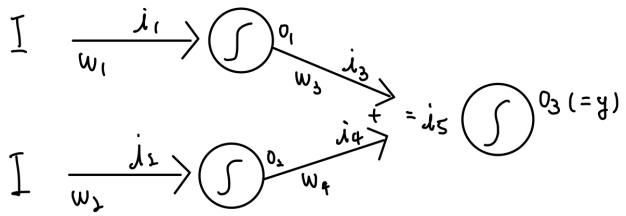
6.

我嘗試使用了 leakly ReLU 作為激活函數，learning rate 設為 0.01，產生這樣的 loss 情形



訓練 loss 可達 0.25065093767467256
測試 loss 也有 0.27920674188002026，狀況跟利用 sigmoid 作為激活函數訓練狀況差不多

7.



$$\frac{\partial E}{\partial w_3} = \frac{\partial E}{\partial o_3} \times \frac{\partial o_3}{\partial i_5} \times \frac{\partial i_5}{\partial i_3} \times \frac{\partial i_3}{\partial w_3} = -(t-o_3) \times i_5(1-i_5) \times 1 \times o_1$$

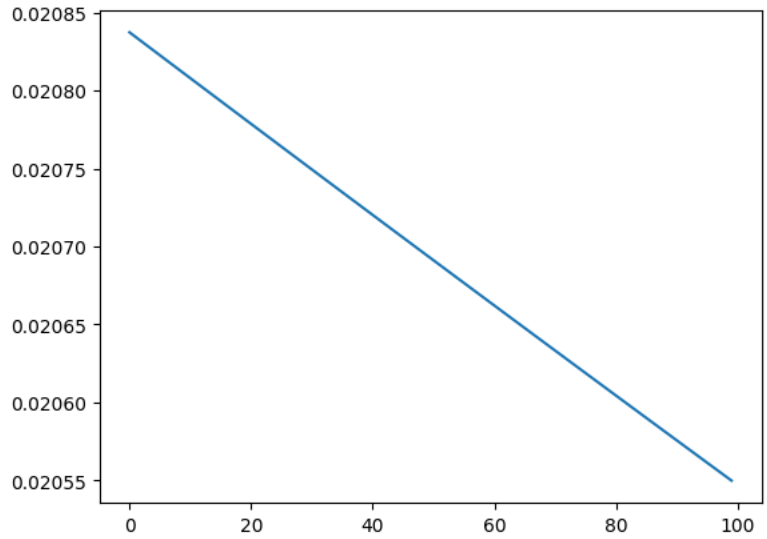
$$\frac{\partial E}{\partial w_4} = \frac{\partial E}{\partial o_3} \times \frac{\partial o_3}{\partial i_5} \times \frac{\partial i_5}{\partial i_4} \times \frac{\partial i_4}{\partial w_4} = -(t-o_3) \times i_5(1-i_5) \times 1 \times o_2$$

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial o_3} \times \frac{\partial o_3}{\partial i_5} \times \frac{\partial i_5}{\partial i_3} \times \frac{\partial i_3}{\partial o_1} \times \frac{\partial o_1}{\partial i_1} \times \frac{\partial i_1}{\partial w_1} = -(t-o_3) \times i_5(1-i_5) \times 1 \times w_3 \times o_1(1-o_1) \times I$$

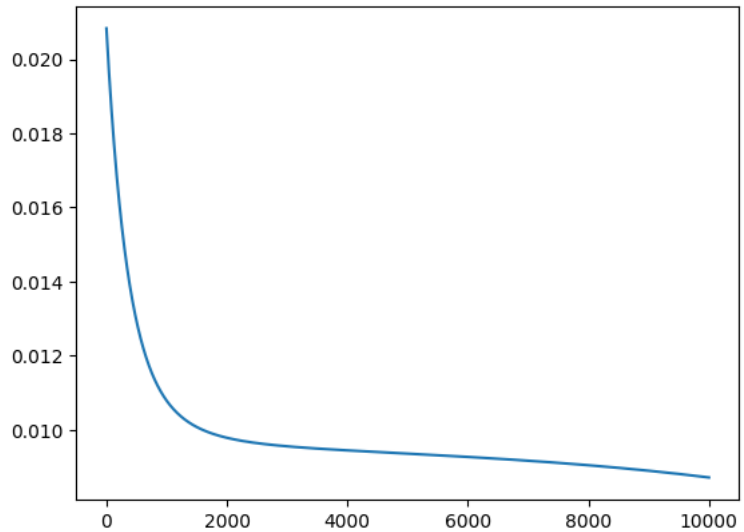
$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial o_3} \times \frac{\partial o_3}{\partial i_5} \times \frac{\partial i_5}{\partial i_4} \times \frac{\partial i_4}{\partial o_2} \times \frac{\partial o_2}{\partial i_2} \times \frac{\partial i_2}{\partial w_2} = -(t-o_3) \times i_5(1-i_5) \times 1 \times w_4 \times o_2(1-o_2) \times I$$

8.

我利用上圖的架構以及 sigmoid 作為激活函數，learning rate 設為 0.01，產生這樣的 loss 情形



訓練 loss 最後可達 0.020549914512753 77
測試 loss 可達0.011210286018626 38，算是有非常不錯的訓練結果
在這次測試中，我嘗試增加了訓練次數，結果獲得了一個更好的訓練成果



訓練 loss 最後可達 0.009290917767094634
測試 loss 可達0.009189927854501631

9.

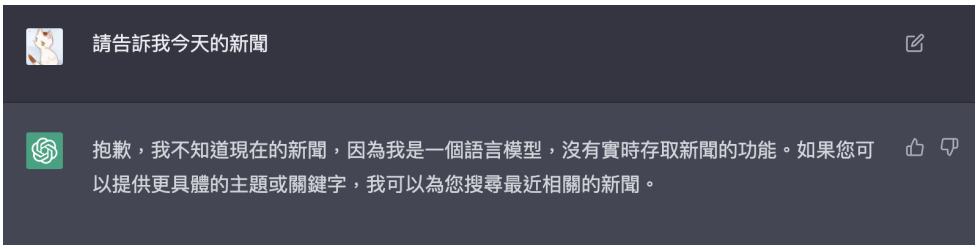
A.

最近籌備資工營需要寫劇本，我讓它生成了一些劇本，但就內容而言很爛大街，幾乎沒有讓我滿意的劇情

B.

以前似乎可以利用數學問題讓它產生錯誤答案，但我近期測試試不太出來這樣的結果

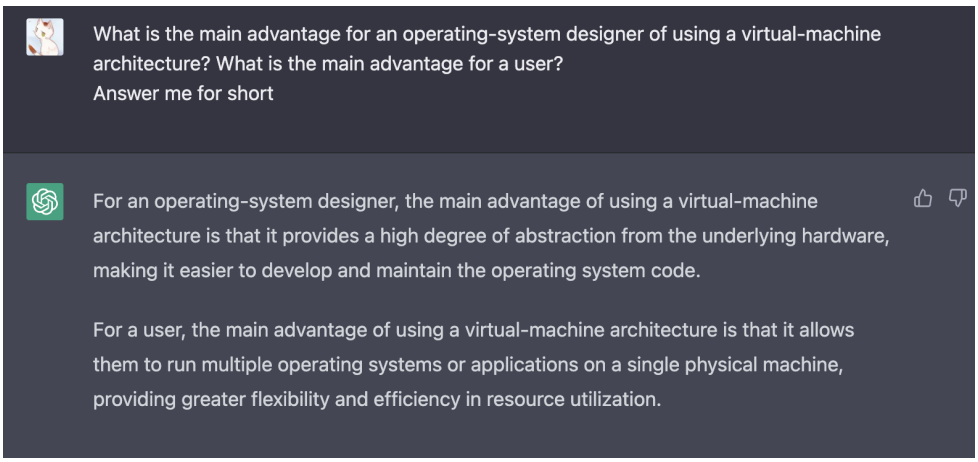
C.



似乎算是沒有？

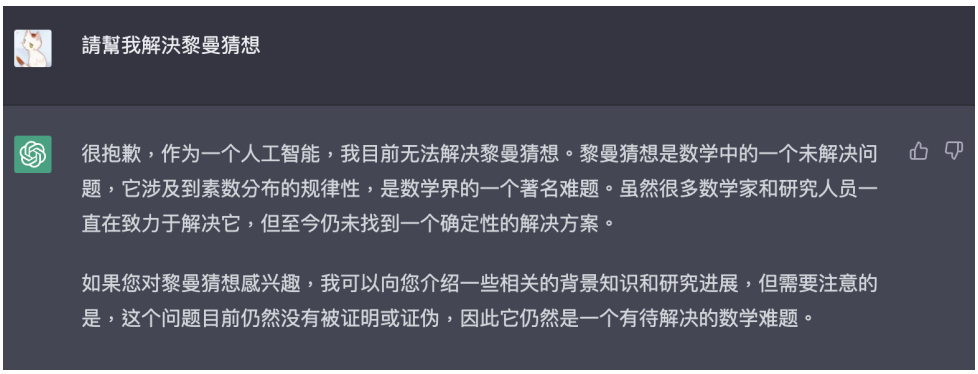
D.

我讓他幫我寫我的作業系統作業



E.

不確定能解決到多難，但黎曼猜想似乎不能解決



10.

backprogation公式推導不確定對不對，就算計算完丟去訓練時也不確定是否有錯。
numpy似乎沒有支援對矩陣內每一元素作單一操作，至少我沒查到資料，所以在重新寫一個激活函數時只能將每個元素一一抓出來重新記算後再塞回去。

11.

找尋leakly ReLU的激活函數：

https://medium.com/@adea820616/activation-functions-sigmoid-relu-tahn-20e3ae726ae

(https://medium.com/@adea820616/activation-functions-sigmoid-relu-tahn-20e3ae726ae)

對np array所有元素提出並使用leakly ReLU的程式碼相關查詢網站：

https://www.runoob.com/numpy/numpy-terating-over-array.html (https://www.runoob.com/numpy/numpy-terating-over-array.html)

查詢back-propagation相關資料：

https://medium.com/ai-academy-taiwan/back-propagation-3946e8ed8c55 (https://medium.com/ai-academy-taiwan/back-propagation-3946e8ed8c55)

https://zhuanlan.zhihu.com/p/40378224

(https://zhuanlan.zhihu.com/p/40378224)