

Deep Learning Lab3 實驗討論
資工四乙 407262316 高挺恩

一、神經元數量的設定與觀察

我是用 Batch 的方式去撰寫程式，所以接下來的神經元設定部分會以實際運行在 Batch 上的結果去做觀察與分析。

實驗上分成單層與多層 Hidden Layer 去做測試，發現多層的設定上好像不管怎麼調，表現的成果都還是比單層的來的差，或許是太複雜也說不定；而在單層方面，嘗試設定較近於輸入資料 x 的神經元或是較趨近於輸出 y 的神經元，會發現在較多神經元的部分大概在 300~400 間會有比較佳的結果、較少的神經元大概在 36~12 之間會有比較高的準確率，且收斂速度也都比較多的神經元還要快。

於是綜合上述觀察，最後我以 [784, 16, 4] 作為整個神經網路的設定並將答案壓縮在資料夾裡面給助教你們瞧瞧><。

二、不同學習率的表現

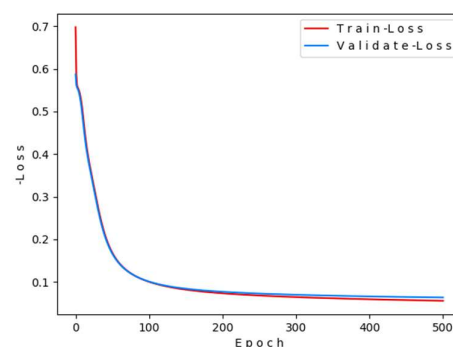
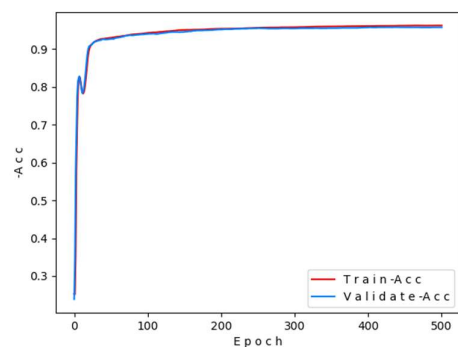
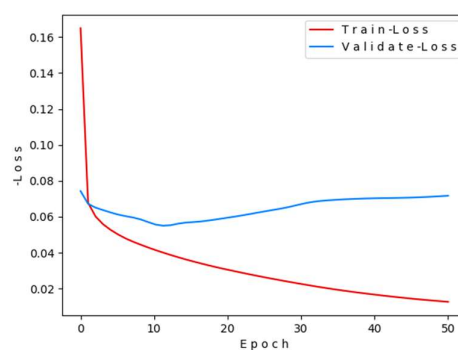
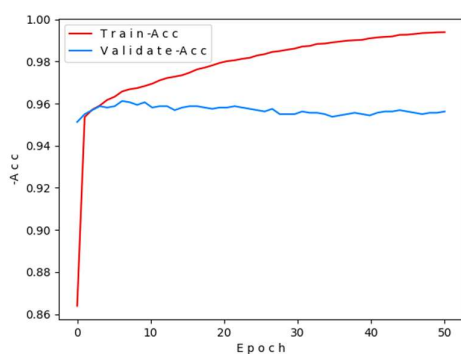
剛剛上述提到，我是用 Batch 去實作，然而其實我最一開始是使用 Stochastic 去跑，在學習率方面我是設定成 0.01，我自己的推論是逐筆作更新所以盡可能不要太大怕讓單筆資料影響後續判讀，而與較大的學習率下去測試的結果也的確讓我確認了，Stochastic 的學習率是可以設定得小

一點、保守一點的；而回到 Batch，由於我是看完所有筆資料後才去做更新，這部分我的學習率就設定的較大許多(0.5)，模型看了這麼多資料後心理至少要有個底該往哪裡走了對吧~

三、圖示分析

下圖分別是 Stochastic(Epoch:50, LR=0.01, [784, 16, 4])

以及 Batch(Epoch:500, LR=0.5, [784, 16, 4])



左邊是訓練與驗證準確率，右邊是訓練與驗證的損失。

雖然運行代數不同，但這是盡量以兩種方法準確率相近去做

設定的，故以此條件下去進行觀察，可以發現 Stochastic 在收斂速度上較為快速，不過很明顯在驗證的損失方面就開始有點失靈，似乎有種 overfitting 的味道飄出來了，除了運行很久之外，還有機會退化，那我跳槽到 Batch 是對的。

至於 Batch 呢，大概收斂到 90% 左右時就會趨緩上升，雖然損失一直有在往下掉，但始終無法低於 10^{-2} 來提早結束訓練。

四、其他心得

這次實驗有幾點小發現，覺得很有趣，實際上網查找也印證了這樣的事！

1. weights 和 bias 的初始化不能都設 0 !!

=> 會導致 gradient 更新都一模一樣，然後準確率就停在那，一動也不動了。

2. 在進行神經元或是學習率的測試時，常會發現準確率會不斷在特定區間逗留!!

=> 這難道就是區域最佳解的坑嗎?!

3. 電腦不是很好、Stochastic 跑 500 代給它爽!!

=> 對啦我 train 到風扇吹壞掉了!!