方法描述-選擇參數的策略

這次作業要關注的參數有torch.manual\_seed()、batch size、epoch、learning rate。

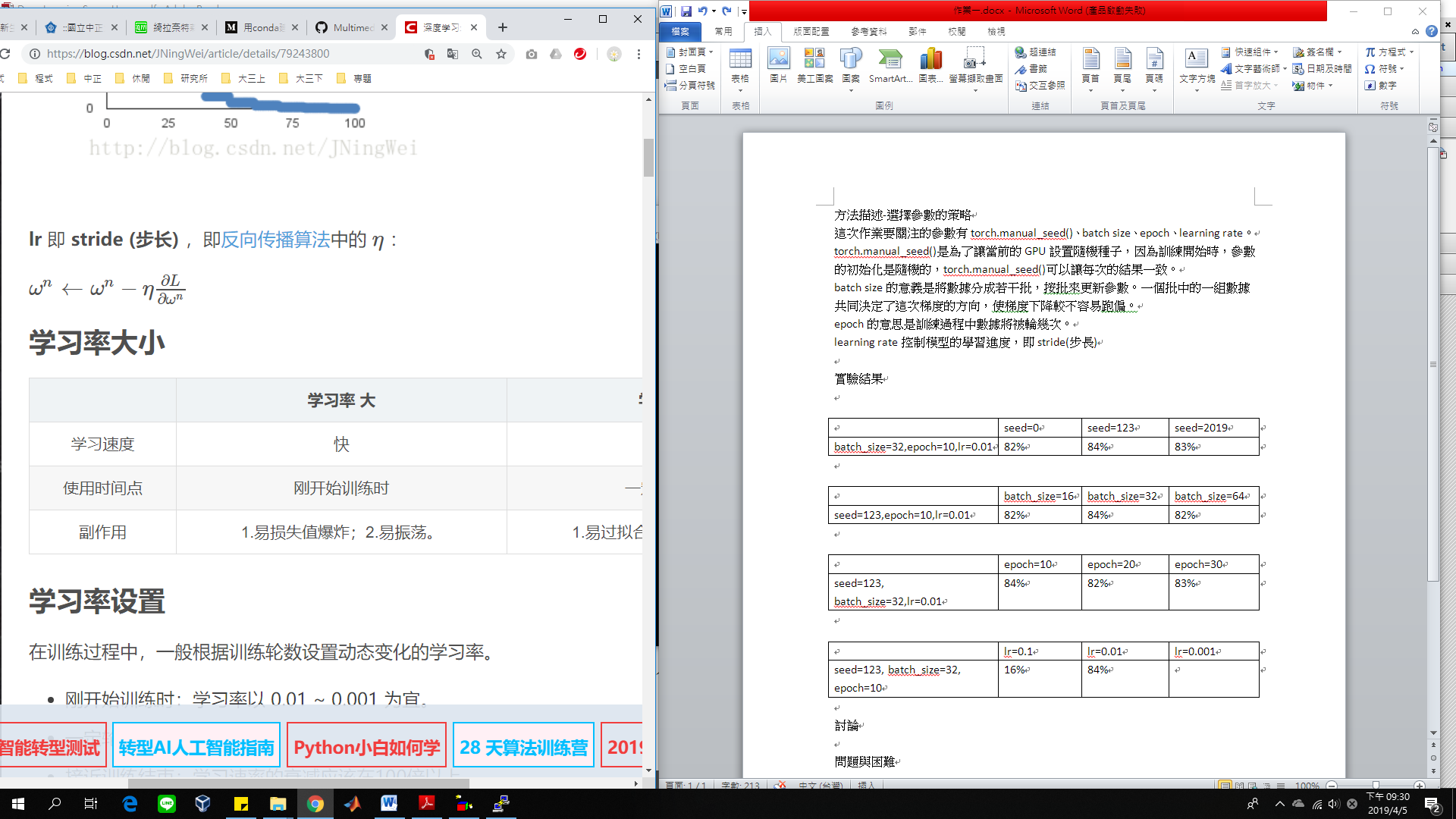
torch.manual\_seed()是為了讓當前的GPU設置隨機種子，因為訓練開始時，參數的初始化是隨機的，torch.manual\_seed()可以讓每次的結果一致。

由於上網查都沒有找到關於設置torch.manual\_seed()的較佳建議，所以我測試作業要求的123還要小的0、作業要求的123還要大的2019。

batch size的意義是將數據分成若干批，按批來更新參數。一個批中的一組數據共同決定了這次梯度的方向，使梯度下降較不容易跑偏。一開始在我的筆電測試時設定batch size=32，但是筆電記憶體不夠導致無法執行程式。所以嘗試batch size=16。後來等到GPU後，也嘗試了batch size=64。上網查是說，batch size沒有一定最佳的準則，要依據各個資料集做調整，所以就老實的慢慢嘗試。

epoch的意思是訓練過程中數據將被輪幾次。epoch跟batch size都是沒有一定最佳的準則，要依據各個資料集做調整，所以也老實的慢慢嘗試。不過上網查有一個網站說，通常epoch設定40~50就差不多了。

learning rate控制模型的學習進度，即stride(步長)

就是反向傳播法中的η

|  |  |  |
| --- | --- | --- |
|  | learning rate大 | learning rate小 |
| 學習速度 | 快 | 慢 |
| 使用時間點 | 剛開始訓練時 | 一定輪數過後 |
| 缺點 | 易loss值爆炸、易震盪 | 易過擬合、收斂速度慢 |

上網查到，0.01~0.001通常較好，也可以設定≤10−4後，再進行微調。一開始我設定0.1，每次除以10去嘗試。

實驗結果

我想測試每個參數對準確率的影響，所以作了以下的實驗 :

只調整seed:

|  |  |  |  |
| --- | --- | --- | --- |
|  | seed=0 | seed=123 | seed=2019 |
| batch\_size=32,epoch=10,lr=0.01 | 82% | 84% | 83% |

只調整batch\_size:

|  |  |  |  |
| --- | --- | --- | --- |
|  | batch\_size=16 | batch\_size=32 | batch\_size=64 |
| seed=123,epoch=10,  lr=0.01 | 82% | 84% | 82% |

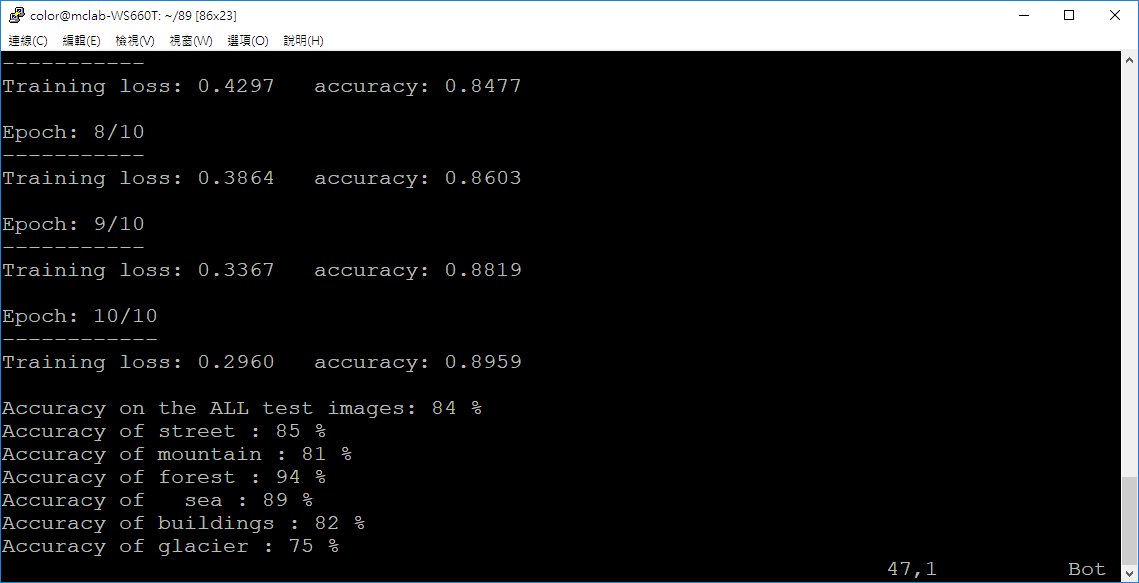
只調整epoch:

|  |  |  |  |
| --- | --- | --- | --- |
|  | epoch=10 | epoch=20 | epoch=30 |
| seed=123, batch\_size=32,lr=0.01 | 84% | 82% | 83% |

只調整lr:

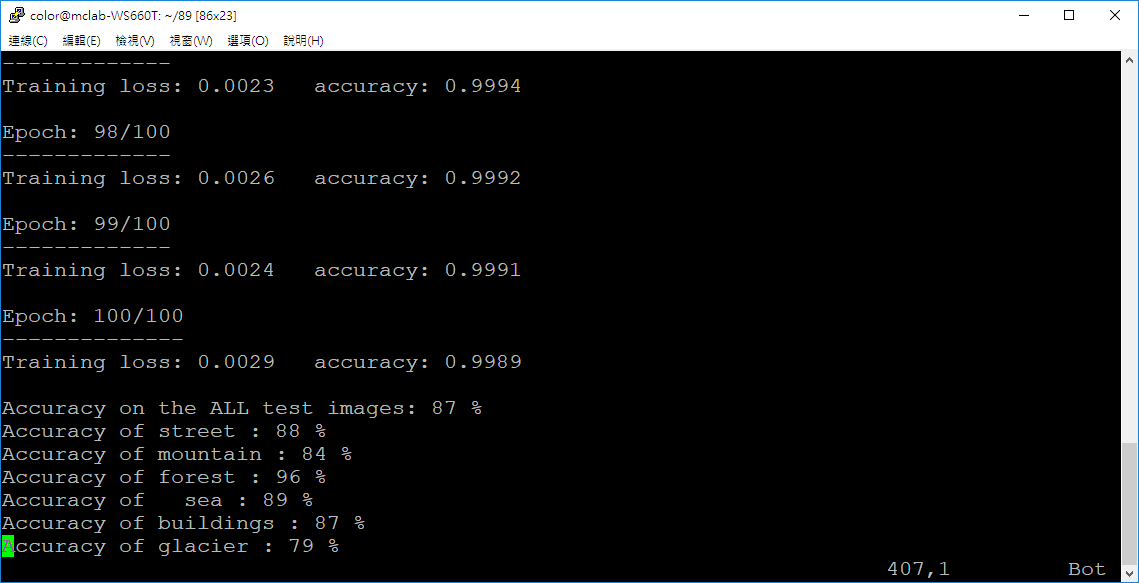
|  |  |  |  |
| --- | --- | --- | --- |
|  | lr=0.1 | lr=0.01 | lr=0.001 |
| seed=123, batch\_size=32, epoch=10 | 16% | 84% | 83% |

seed=123,batch\_size=32,epoch=10,lr=0.01(作業要求):



|  |  |
| --- | --- |
| accuracy | loss |
| C:\Users\User\Documents\大四下\機器學習\hw1\intel-image-classification\tt\batch32,epoch10,lr0.01\accuracy.png | C:\Users\User\Documents\大四下\機器學習\hw1\intel-image-classification\tt\batch32,epoch10,lr0.01\loss.png |

seed=123,batch\_size=32,epoch=100,lr=0.001:



|  |  |
| --- | --- |
| accuracy | loss |
| C:\Users\User\Documents\大四下\機器學習\hw1\intel-image-classification\tt\batch32,epoch100,lr0.001\accuracy.png | C:\Users\User\Documents\大四下\機器學習\hw1\intel-image-classification\tt\batch32,epoch100,lr0.001\loss.png |

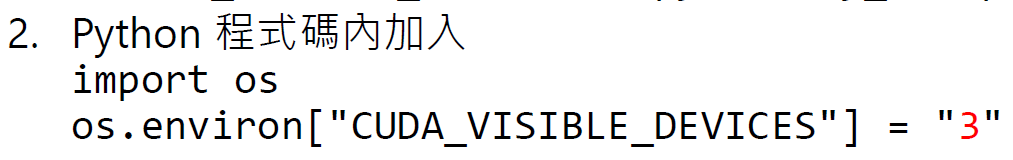
討論

在我針對只調整seed、只調整batch\_size、只調整epoch、只調整lr的實驗中，發現並非參數越大、準確率越好。而是要針對各種資料集進行測試，找出適合的參數。

在作業要求的seed=123,batch\_size=32,epoch=10,lr=0.01中，發現到最後幾次epoch時，test的loss會上升，發生過擬合現象。

而在我自己測試的seed=123,batch\_size=32,epoch=100,lr=0.001中過擬合的現象在整個訓練中更快的發生，test的曲線很快就開始震盪。我想是因為learning rate設定較小(lr=0.001)的緣故。

問題與困難

一開始我在程式碼中按照助教的投影片加入指定使用哪一顆GPU的指令(如下圖)

但是會一直出現找不到CUDA的錯誤回報，後來不使用這段程式碼後就解決了。我想是因為torch對參數處存的位置有記憶，而我有指定換過CUDA，才會發生這樣的問題。

要把train和test的數據畫在圖一張圖上，代表需要在一支程式中同時存在train和test的數據。

我多放了一個hw1.py的檔案，是把train和test結合成一支程式，所以只要執行hw1.py程式就可以產生accuracy和loss的曲線圖、testing accuracy，不產生pth檔。

train.py和test.py是跟作業要求的規定一樣，先執行train產生pth檔、accuracy和loss的曲線圖，再執行test得到testing accuracy。

在储存train和test的accuracy、loss的數據時，我使用extend的方式把數據存到list裡面，但是卻出現 train\_loss\_list.extend(training\_loss)

TypeError: 'float' object is not iterable

的錯誤回報。後來改成用append才可以執行。