- ## Execution description：

1. PLA：

在 PLA.py 所在的資料夾中開啟 terminal，輸入要執行的程式和 dataset 的指令，例如要用 PLA 執行 data_30_1.txt，則輸入：python PLA.py –path "..\2025_CCU_ML_HW2_Data\data_30_1.txt"，其中須確保 DataLoader.py 與此程式在相同資料夾下。
程式收到指令後開始讀取輸入的 dataset 是否存在，若不存在，則輸出 File not found, please try again；若存在，則透過 DataLoader 讀取 dataset 送入 PLA 函數執行。
初始權重為隨機一個資料點(因為需要多做幾次，所以需要不同起始點才可能有不同結果)，開始尋找下一個錯誤點，利用以下公式尋找錯誤點：$sign(\boldsymbol{w}_t^T \boldsymbol{x}_{n(t)}) \neq y_{n(t)}$，若確定此為真正的錯誤點，則使用以下公式更新權重：$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t^T + y_{n(t)}\boldsymbol{x}_{n(t)}$，並計為 1 次 iteration。重複執行直到利用此權重完全分割所有資料點。最後顯示 iterations 次數與執行時間，並利用最終權重畫出學習到的分割線。

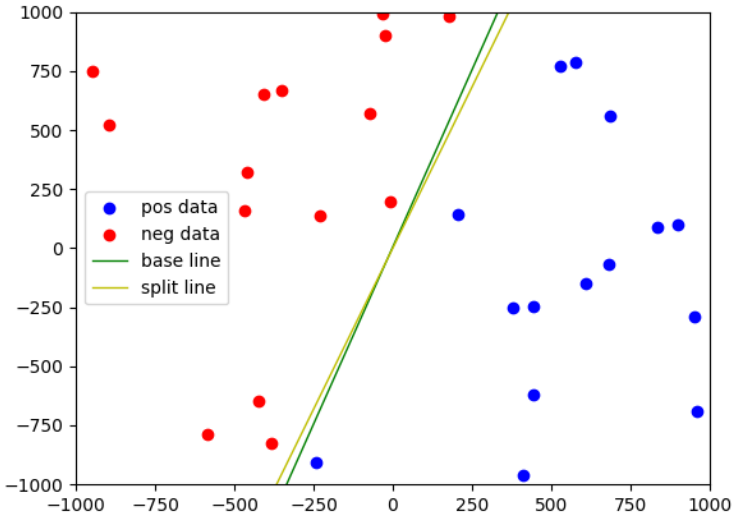
2. Pocket algorithm：

在 Pocket.py 所在的資料夾中開啟 terminal，輸入要執行的程式和 dataset 的指令，例如要用 Pocket algorithm 執行 data_2000.txt，則輸入：python Pocket.py –path "..\2025_CCU_ML_HW2_Data\data_2000.txt"，其中須確保 DataLoader.py 與此程式在相同資料夾下。
程式收到指令後開始讀取輸入的 dataset 是否存在，若不存在，則輸出 File not found, please try again；若存在，則透過 DataLoader 讀取 dataset 送入 PLA 函數執行。
初始權重為[0,0,0]，首先利用以下公式尋找現有的權重有多少個錯誤點(以下稱為 wrong1)：$sign(\boldsymbol{w}_t^T \boldsymbol{x}_{n(t)}) \neq y_{n(t)}$，再選取這些錯誤點中隨機一個資料點，使用以下公式紀錄一個新權重：$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t^T + y_{n(t)}\boldsymbol{x}_{n(t)}$，並利用此新權重分類看看並計算有多少個錯誤點(以下稱為 wrong2)。比較 wrong1 與 wrong2，如果 wrong2 比 wrong1 還小，就將權重更新為新權重，並計為 1 次 iteration。重複執行直到 iteration 數達到上限。最後顯示分類準確度與執行時間，並利用最終權重畫出學習到的分割線。

- ## Experimental results：

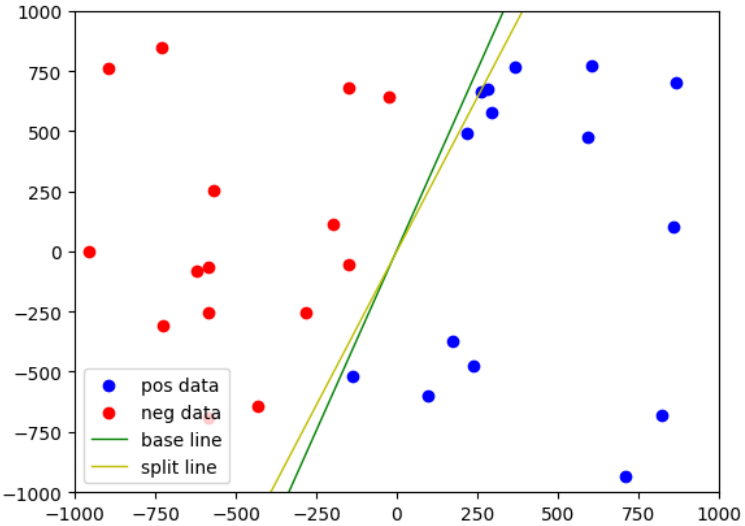
1. PLA：
   Data_30_1.txt 的結果

```
PS D:\下載\碩一下\機器學習\homework\HW2\2025_CCU_ML_HW2_SampleCodes>
 python PLA.py --path "..\2025_CCU_ML_HW2_Data\data_30_1.txt"
第 1 次： iterations次數為 2 次
第 2 次： iterations次數為 32 次
第 3 次： iterations次數為 25 次
第 4 次： iterations次數為 8 次
第 5 次： iterations次數為 2 次
第 6 次： iterations次數為 1 次
平均iteration次數為 11.666666666666666 次
ex time : 0.001008
```



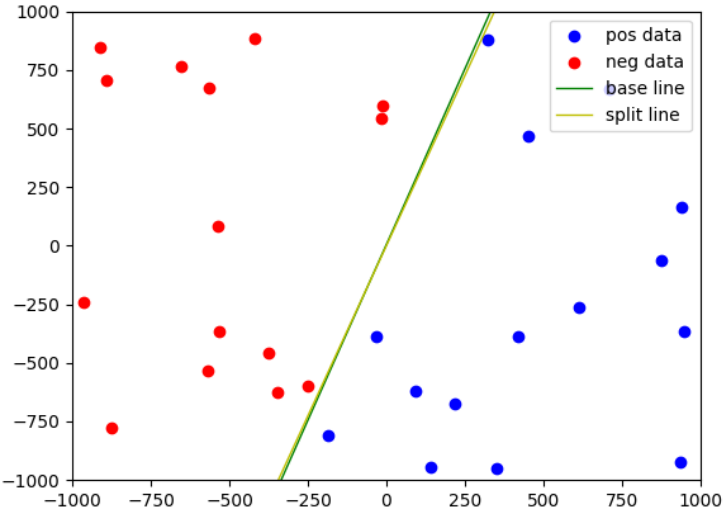
Data_30_2.txt 的結果

```
PS D:\下載\碩一下\機器學習\homework\HW2\2025_CCU_ML_HW2_SampleCodes>
 python PLA.py --path "..\2025_CCU_ML_HW2_Data\data_30_2.txt"
第 1 次： iterations次數為 42 次
第 2 次： iterations次數為 42 次
第 3 次： iterations次數為 32 次
第 4 次： iterations次數為 39 次
第 5 次： iterations次數為 13 次
第 6 次： iterations次數為 61 次
平均iteration次數為 38.166666666666664 次
ex time : 0.004519
```



Data_30_3.txt 的結果

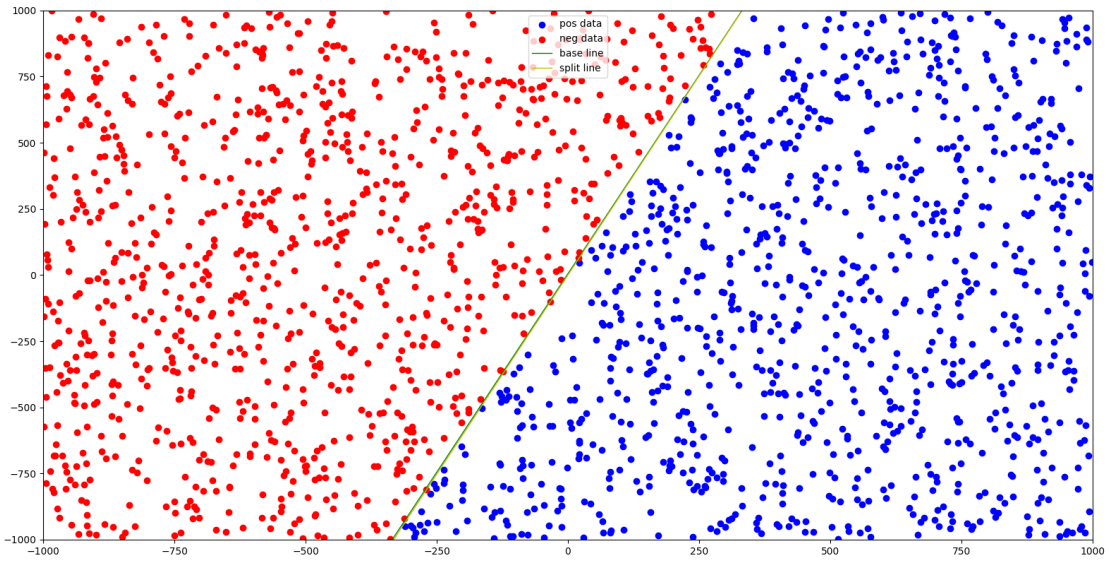
```
PS D:\下載\碩一下\機器學習\homework\HW2\2025_CCU_ML_HW2_SampleCodes>
 python PLA.py --path "..\2025_CCU_ML_HW2_Data\data_30_3.txt"
第 1 次： iterations次數為 11 次
第 2 次： iterations次數為 13 次
第 3 次： iterations次數為 5 次
第 4 次： iterations次數為 6 次
第 5 次： iterations次數為 2 次
第 6 次： iterations次數為 13 次
平均iteration次數為 8.333333333333334 次
ex time : 0.001507
```



2. PLA 與 Pocket algorithm 比較：

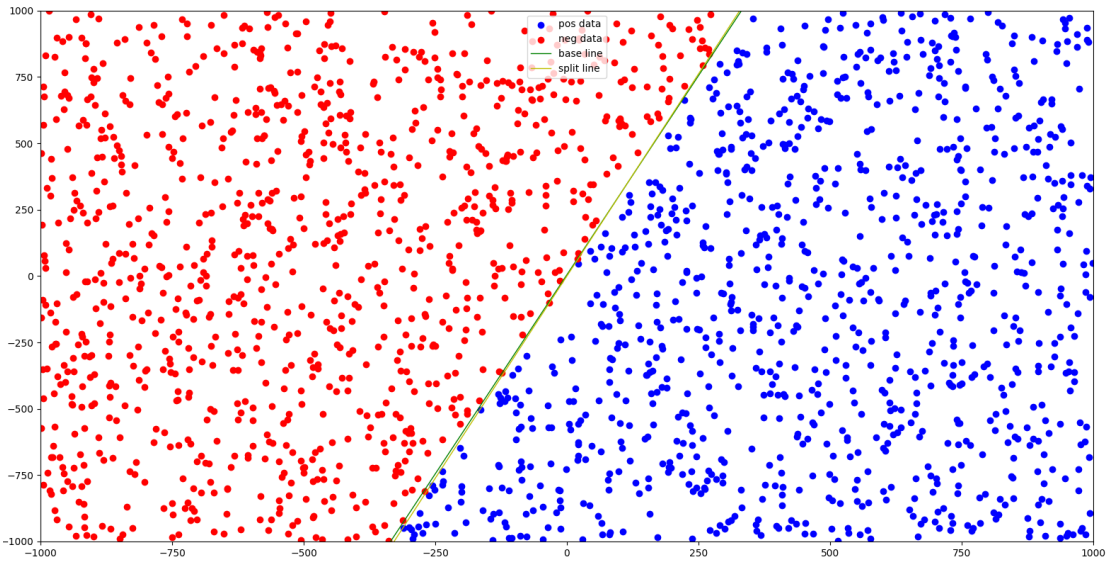
Data_2000.txt 在 PLA 的結果

```
PS D:\下載\碩一下\機器學習\homework\HW2\2025_CCU_ML_HW2_SampleCodes>
 python PLA.py --path "..\2025_CCU_ML_HW2_Data\data_2000.txt"
第 1 次： iterations次數為 24 次
第 2 次： iterations次數為 8 次
第 3 次： iterations次數為 153 次
第 4 次： iterations次數為 155 次
第 5 次： iterations次數為 57 次
第 6 次： iterations次數為 33 次
平均iteration次數為 71.66666666666667 次
ex time : 0.011819
```



Data_2000.txt 在 Pocket algorithm 的結果

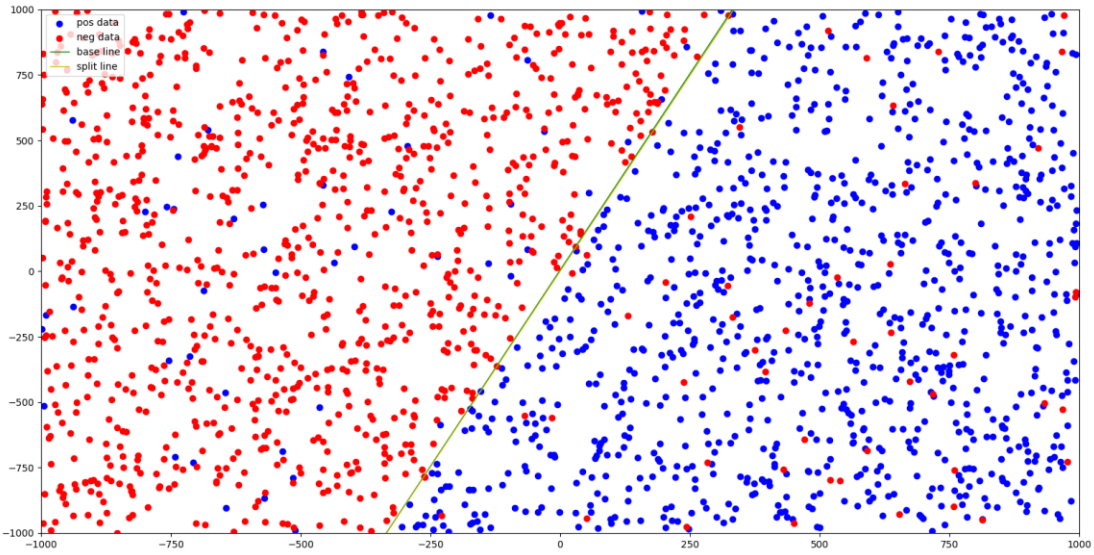
```
PS D:\下載\碩一下\機器學習\homework\HW2\2025_CCU_ML_HW2_SampleCodes>
 python Pocket.py --path "..\2025_CCU_ML_HW2_Data\data_2000.txt"
Accuracy: 100.00%
Execution time = 0.394176
```



3. Pocket algorithm：

Data_2000_50_wrong.txt 的結果

```
PS D:\下載\碩一下\機器學習\homework\HW2\2025_CCU_ML_HW2_SampleCodes>
 python Pocket.py --path "..\2025_CCU_ML_HW2_Data\data_2000_50wrong.
txt"
Accuracy: 94.95%
Execution time = 0.417520
```



- ## Conclusion：

PLA 演算法設計為直到所有資料點都被分類正確才會結束，因此一定會找到一條線分割所有資料點(前提為可線性分割)，我們可以在上面的截圖中看到，PLA 可以確實地分割所有資料點，但並不保證可以跟基準線重合；在 Pocket algorithm 中則是保證在有限制的 iteration 數可以找到表現最好的權重。在輸入為 data_2000.txt 時，因其為 linear separable，因此可以找到確實分割資料點的線。但因為比 PLA 多了相比權重的表現程度，在執行時間上會輸給 PLA。在輸入為 data_2000_50wrong.txt 中，因為不是 linear separable，因此無法找到可以確實分割資料點的線，也就因此 accuracy 落在 94.95%。

- ## Discussion：

在實作 PLA 時，因作業要求要多做幾次取平均，因此我寫了 for 迴圈讓他可以看資料點，但實際上我並沒有重置 weight_matrix，導致在於訓練第一次以後都不用任何 iteration 即可完美分割所有點，我修改 weight_matrix 定義在 for 迴圈裡面，結果又錯了，因為每次跑都是相同的 iteration 數，這樣做那麼多次就沒有意義了，因此我看了老師的講義，初始權重是設定為任意一個點，因此我就改變寫法，改為隨機選擇初始點，讓權重可以在每次做時都是不同的，才可以讓整體訓練過程不是相同，進而 iteration 數與最後的分割線不同。