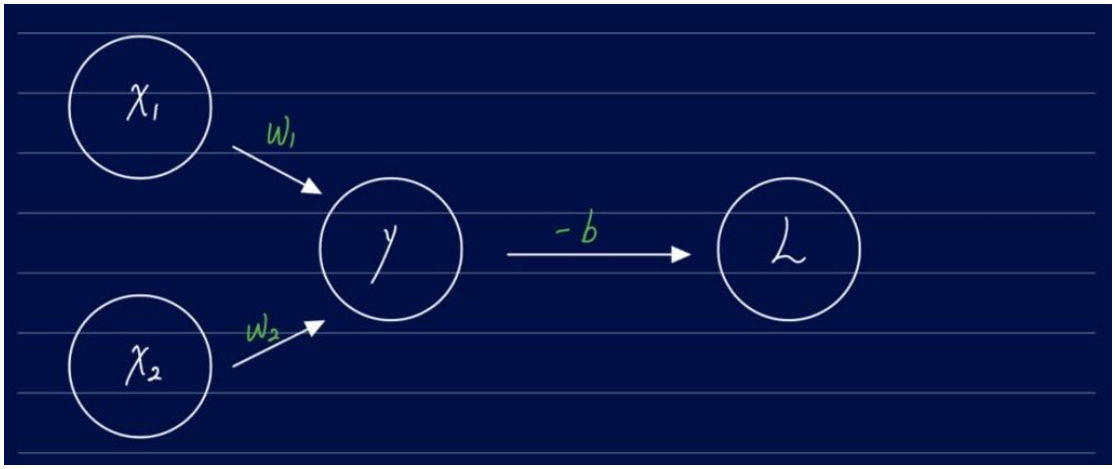
程式實作-感知器(perceptron)

目的：觀察 XOR Gate 傳遞訊號的範圍

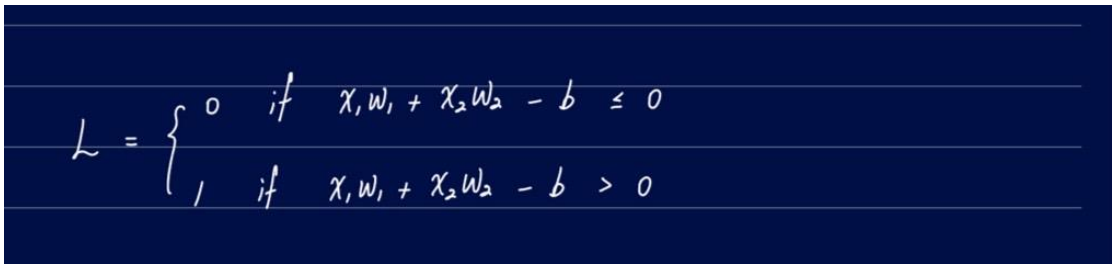
感知器一共可分為四個閘：分別為及閘(AND Gate)、反及閘(NAND Gate)、或閘(OR Gate)和互斥或閘(XOR Gate)，感知器在收到多個訊號後，再當作一個訊號輸出，主要影響的是權重(w1,w2)與偏權值(b)

如下圖：



(圖 1)

而輸出的值只為 0 (代表不傳遞訊號)或 1 (代表傳遞訊號)，如下圖：



(圖 2)

以下為四個閘的真值表：

及閘 (AND Gate) 的真值表

x_1	x_2	y
0	0	0
1	0	0
0	1	0
1	1	1

反及閘 (NAND Gate) 的真值表

x_1	x_2	y
0	0	1
1	0	1
0	1	1
1	1	0

或閘（OR Gate）的真值表

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	1

互斥或閘（XOR Gate）的真值表

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	0

以下用程式碼來執行各個閘的定義：

```
import numpy as np
import matplotlib.pyplot as plt
def OR(x1,x2):
    L = w31*x1+w32*x2-b3
    if L > 0 : return 1
    else : return 0

def NAND(x1,x2):
    L = w21*x1+w22*x2-b2
    if L > 0 : return 1
    else : return 0

def AND(x1,x2):
    L = w11*x1+w12*x2-b1
    if L > 0 : return 1
    else : return 0

def XOR(x1,x2):
    return AND(NAND(x1,x2) , OR(x1,x2))
```

由上圖可以發現，及閘、反及閘和或閘是結構相同的感知器，差別只在於設定不同的權重與偏權值。

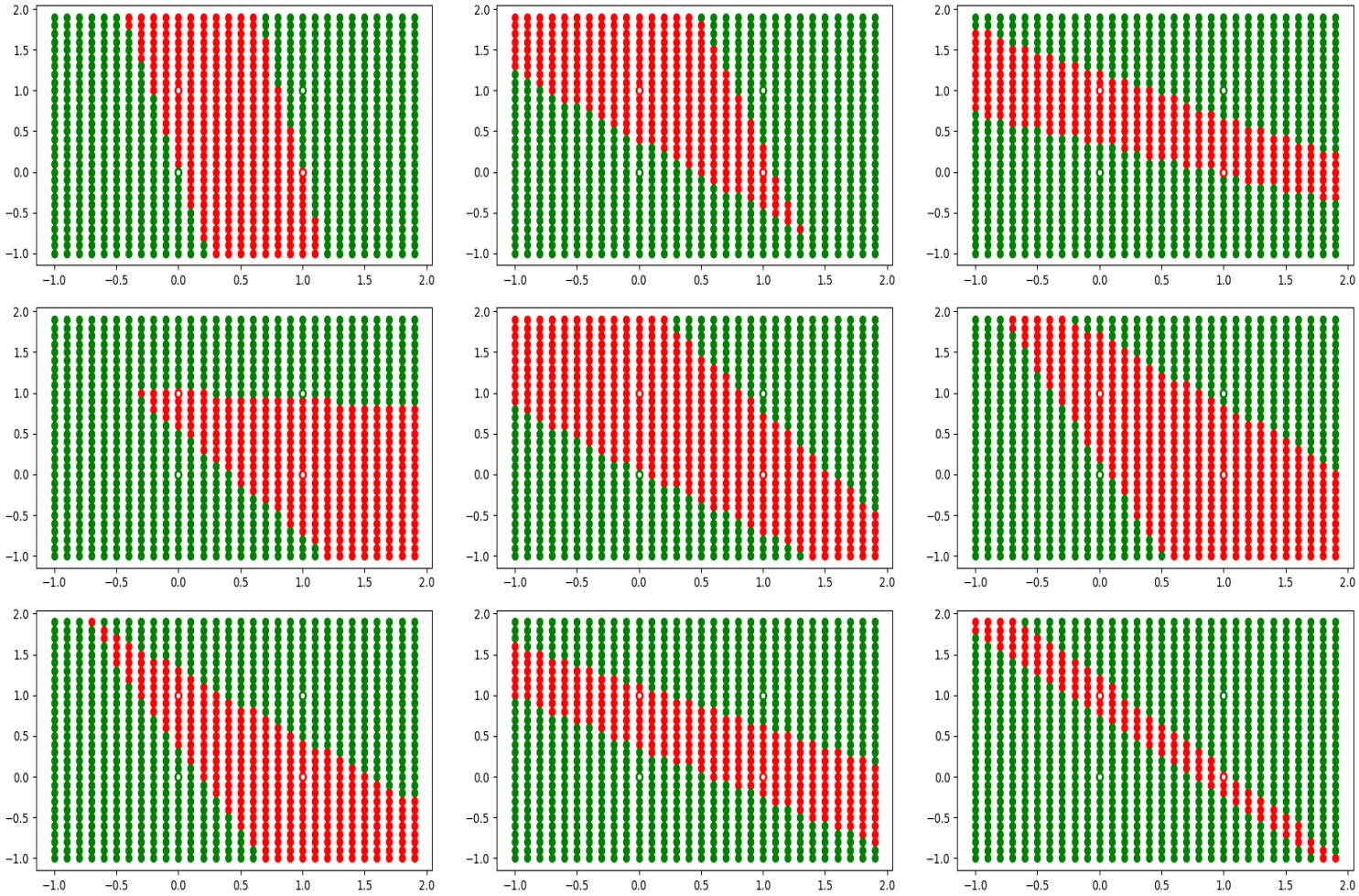
因為 XOR 在數線上無法用一條線來區分，因此稱為「非線性」，所以這邊的 XOR 使用的是雙層感知器，因為單層感知器無法分離非線性區域

那為了可以呈現出 XOR 的區域範圍，我給定權重與偏權值一個範圍，並在 XOR 傳遞訊號時畫上紅色的點，未傳遞訊號時畫上綠色的點，且隨機運算 9 次來觀察 XOR 的區域變化，程式碼如下圖：

```
plt.figure(1,figsize = (14,9))
for i in range(1,10):
    while True:
        w11,w12,b1 = np.random.uniform(-9,9,3)
        if -b1 <= 0 and w11-b1 <= 0 and w12-b1 <= 0 and w11+w12-b1 > 0 :  #設定(0,0),(1,0),(0,1),(1,1)的情況
            break
    while True:
        w21,w22,b2 = np.random.uniform(-9,9,3)
        if -b2 > 0 and w21-b2 > 0 and w22-b2 > 0 and w21+w22-b2 <= 0 :
            break
    while True:
        w31,w32,b3 = np.random.uniform(-9,9,3)
        if -b3 <= 0 and w31-b3 > 0 and w32-b3 > 0 and w31+w32-b3 > 0 :
            break
    X = np.arange(-1,2,0.1)   #數線範圍設定在[-1,2)之間，每次前進0.1
    Y = np.arange(-1,2,0.1)
    rx , ry = [],[]
    gx , gy = [],[]
    for x in X:
        for y in Y:
            if XOR(x,y) > 0 :  #若XOR傳遞訊號則畫上紅點，否則畫上綠點
                rx.append(x)
                ry.append(y)
            else :
                gx.append(x)
                gy.append(y)
    plt.subplot(3,3,i)
    plt.scatter(rx,ry , c = 'r' , s = 35)
    plt.scatter(gx,gy , c = 'g' , s = 35)
    plt.scatter([0,1],[1,0] , c = 'w' , edgecolors = 'r' , linewidths = 2 , s = 35)
    plt.scatter([0,1],[0,1] , c = 'w' , edgecolors = 'g' , linewidths = 2 , s = 35)

plt.tight_layout()
plt.show()
```

結果



上圖紅色區域表示 XOR 傳遞訊號的區域