

Random Dice & Nonlinear function plotting

參考範例: 模擬一顆公平骰子 N 次，求骰子點數的出現機率？

```
import random
Dice = [i for i in range(1,7)]
p = []
times = 20
for i in range(times):
    dice = random.choice(Dice)
    p.append(dice)
print("p:",p)
a = set(p)
for j in a:
    print(j, " appears ", p.count(j), "times", p.count(j)/times)
```

匯入套件

將range(1,7)的整數指定為串列元素
串列Dice包含整數
1到6

```
p: [2, 3, 2, 4, 4, 3, 1, 4, 6, 2, 4, 4, 3, 6, 5, 3, 4, 2, 3, 3]
1 appears 1 times
2 appears 4 times
3 appears 6 times
4 appears 6 times
5 appears 1 times
6 appears 2 times
```

```
import random
Dice = [i for i in range(1,7)]
p = []
times = 20
for i in range(times):
    dice = random.choice(Dice)
    p.append(dice)
print("p:",p)
a = set(p)
for j in a:
    print(j, " appears ", p.count(j), "times", p.count(j)/times)
```

迴圈執行20次

使用套件中的方法
choice，從串列Dice
中，隨機選擇一個元素

將dice附加在
串列p

```
p: [2, 3, 2, 4, 4, 3, 1, 4, 6, 2, 4, 4, 3, 6, 5, 3, 4, 2, 3, 3]
1 appears 1 times
2 appears 4 times
3 appears 6 times
4 appears 6 times
5 appears 1 times
6 appears 2 times
```

```
import random
Dice = [i for i in range(1,7)]
p = []
times = 20
```

```
for i in range(times):
    dice = random.choice(Dice)
    p.append(dice)
```

```
print("p:",p)
```

```
a = set(p)
```

```
for j in a:
```

```
    print(j, " appears ", p.count(j), "times", p.count(j)/times)
```

a為集合，包含串列p中的每一個元素

```
p: [2, 3, 2, 4, 4, 3, 1, 4, 6, 2, 4, 4, 3, 6, 5, 3, 4, 2, 3, 3]
1  appears  1 times
2  appears  4 times
3  appears  6 times
4  appears  6 times
5  appears  1 times
6  appears  2 times
```

a為集合，j代表集合中的每一個元素，p.count(j)代表j在串列p中出現的次數

```

1 import random
2 Dice = [i for i in range(1,7)]
3 p = []
4 times = 20000
5 for i in range(times):
6     dice = random.choice(Dice)
7     p.append(dice)
8     # print("p:",p)
9 a = set(p)
10 for j in a:
11     print(j, " appears ", p.count(j), "times", "prob : ", p.count(j)/times)

```

```

1 1 appears 3303 times prob : 0.16515
2 2 appears 3290 times prob : 0.1645
3 3 appears 3367 times prob : 0.16835
4 4 appears 3370 times prob : 0.1685
5 5 appears 3306 times prob : 0.1653
6 6 appears 3364 times prob : 0.1682

```

Calculate cross-entropy of simulated probabilities $\{p_1, \dots, p_6\}$ with respect to $\{q_1 = 1/6, \dots, q_6 = 1/6\}$

$$CE(q || p) = \sum_i q_i \log\left(\frac{q_i}{p_i}\right)$$

練習：模擬兩顆公平骰子 N 次，求兩顆骰子不同點數和的出現次數。將出現次數除以 N ，得到機率值，求各點數和的出現機率值？

修改提示

參考範例

- 範例中的第一個迴圈，改成擲兩次骰子，並計算分數，附加在串列中
- 第二個迴圈中，可增加列印機率

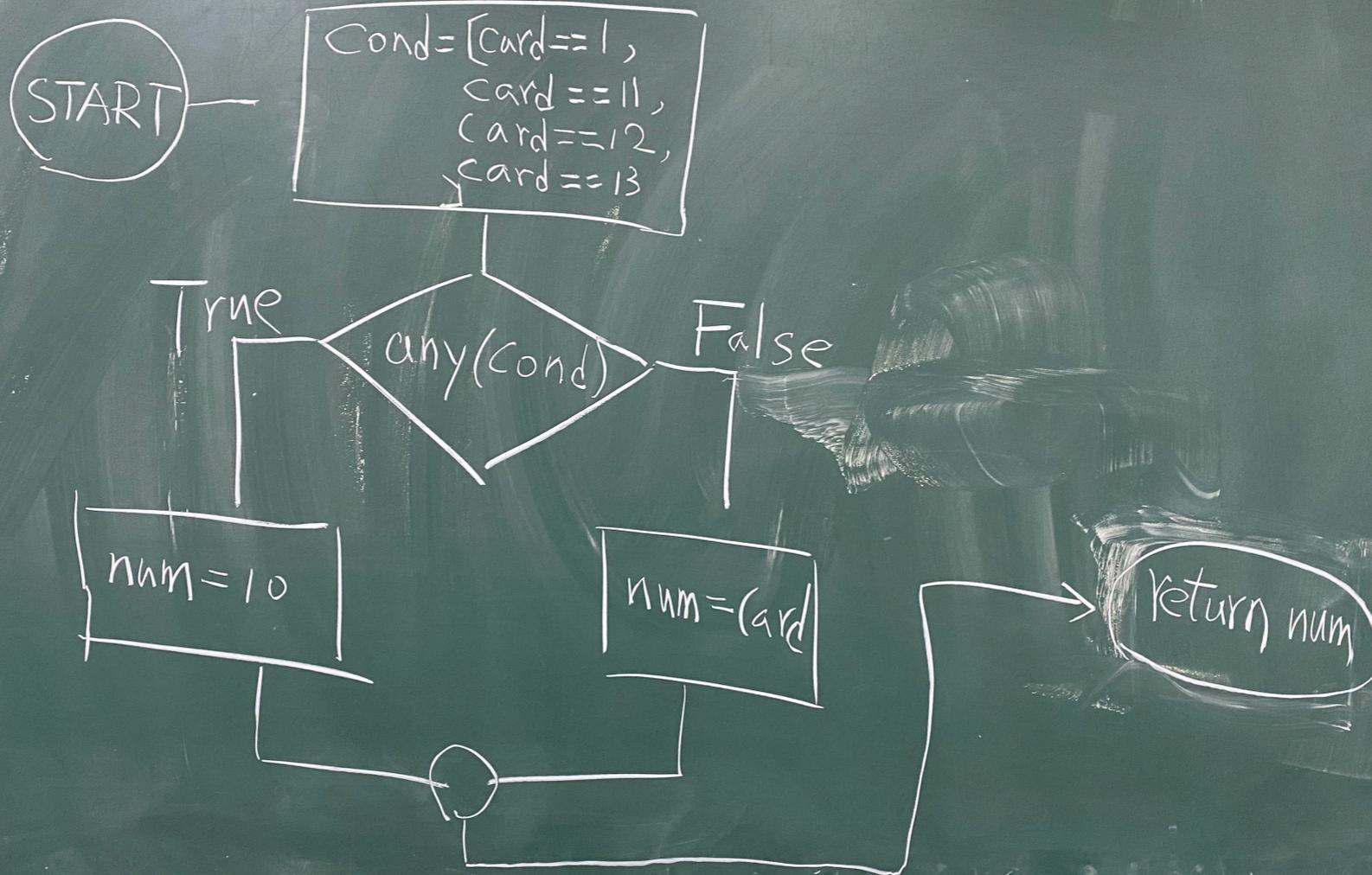
```
for i in range(times):  
    dice = random.choice(Dice)  
    p.append(dice)
```

```
for j in a:  
    print(j, " appears ", p.count(j), "times")
```

**Translate one card to
score**

```
1 import random
2 def card2score(card):
3     cond = [card == 1,
4             card == 11,
5             card == 12,
6             card == 13
7             ]
8     if any(cond):
9         num = 10
10    else:
11        num = card
12    return num
13
14 card = 11
15 score = card2score(card)
16 print(score)
```

```
def card2score(card)
```



blackJack :

1. determine the score of two cards

2. Perform 10^6 times and calculate the average score of two randomly generated cards

```
1 import random
2 def card2score(card):
3     cond = [card == 1,
4             card == 11,
5             card == 12,
6             card == 13
7             ]
8     if any(cond):
9         num = 10
10    else:
11        num = card
12    return num
13
14 lst = [i for i in range(1,14)]
15 scoreSum = 0
16 times = 1000000
17 for i in range(times):
18     card1 = random.choice(lst)
19     card2 = random.choice(lst)
20     score = card2score(card1) + card2score(card2)
21     scoreSum += score
22 print('average score: ', scoreSum/times)
```

Apply a for-loop

average score: 14.464955

練習. 模擬擲三顆骰子，求分數。規則如下：

1. 三顆點數都不一樣，重新執一次
2. 兩顆一樣，一顆不一樣，以單獨出現的點數計分
3. 三顆點數都一樣，得7分

提示

參考範例 slide 3

- 參考模擬擲骰子的指令，
擲三次骰子，分別加入串列中
- 將串列轉換為集合
- 集合的元素個數可以用來
計算分數

```
dice = random.choice(Dice)  
p.append(dice)
```

```
a = set(p)
```

集合a的元 素個數	下一個執 行命令
1	設定得分7分
2	找出出現一次 的骰子點數 設定為得分分數
3	重新擲一次

$$n_1 = 6$$

$$n_2 = 3 \times 6 \times 5 = 90$$

$$n_3 = 6 \times 5 \times 4 = 120 \text{種}$$

$$n = 6 \times 6 \times 6 = 216$$

$$S_1 = 7 \times 6 = 42$$

$$\begin{aligned} S_2 &= 3 \times (1 \times 5 + 2 \times 5 + 3 \times 5 + 4 \times 5 + 5 \times 5 + 6 \times 5) \\ &= 3 \times 5 \times \sum_{i=1}^6 i = 15 \times 21 = 210 + 105 = 315 \end{aligned}$$

$$\frac{S_1 + S_2}{n} = \frac{42 + 315}{216} =$$

Python Console

```
>>> (42+315)/216
```

```
1.6527777777777777
```

```
1 import random
2 Dice = [i for i in range(1,7)]
3 scoreSum = 0
4 gameTimes = 500000
5 for j in range(gameTimes):
6     p = []
7     times = 3
8     for i in range(times):
9         dice = random.choice(Dice)
10        p.append(dice)
11    a = set(p)
12    if len(a) == 3:
13        score = 0
14    else:
15        if len(a) == 2:
16            for j in a:
17                if p.count(j) == 1:
18                    score = j
19            else:
20                score = 7
21    scoreSum += score
22 print("sum of scores:", scoreSum)
23 print("average of score: ", scoreSum/gameTimes)
```

3 random
dices

Translate a
list to a set

Count	
1	1.653054
2	1.655168
3	1.65218
4	1.649984
5	1.649687
6	1.651582
7	1.652143
8	1.649858
9	1.654013
10	1.655465
Mean	
Var	

```

import numpy as np
a = [1.653054,
1.655168,
1.65218,
1.649984,
1.649687,
1.651582,
1.652143,
1.649858,
1.654013,
1.655465]
print('mean :', np.mean(a))
print('var :', np.var(a))

```

1.6527

```

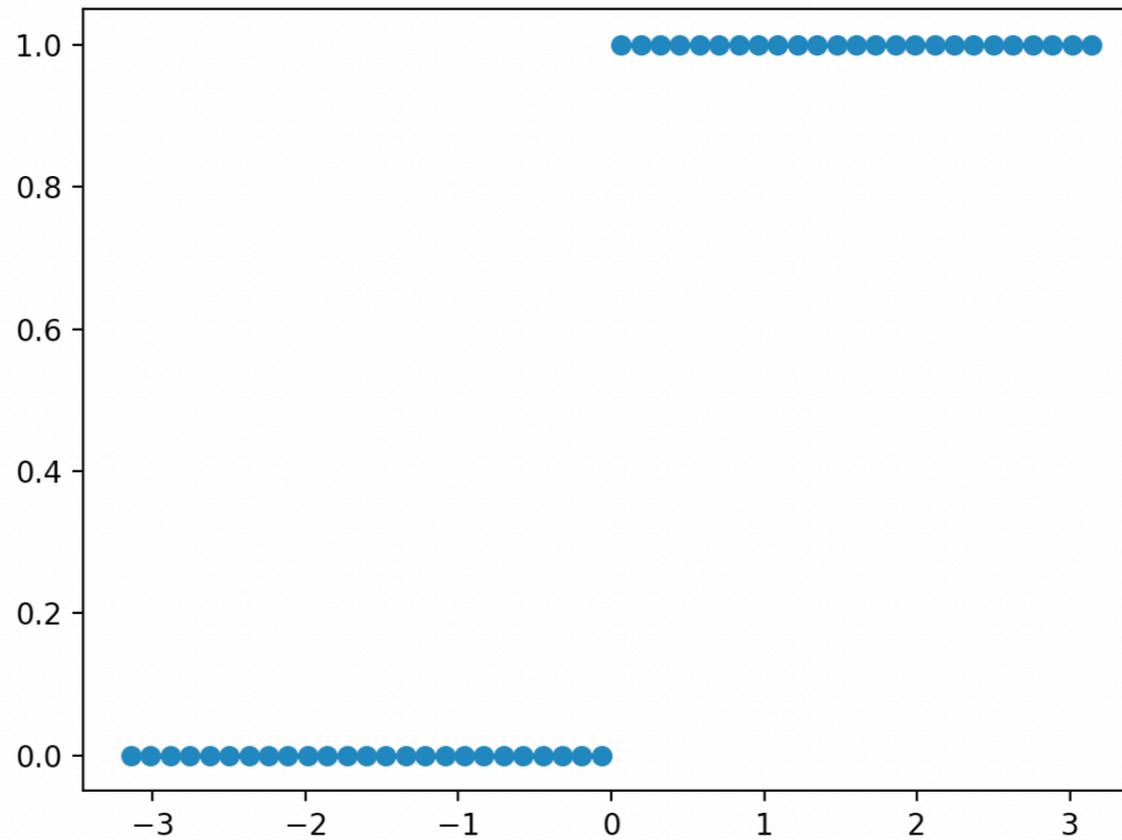
mean : 1.6523134
var : 4.045330040000014e-06

```

練習. 模擬擲四顆骰子，求分數。規則如下：

1. 四顆點數都不一樣，重新執一次
2. 三顆一樣，一顆不一樣，重新執一次
3. 四顆點數都一樣，得12分
4. 兩顆一樣，兩顆不一樣，以不一樣的兩顆骰子和計分
5. 兩顆一樣，另外兩顆也一樣，但四顆骰子出現兩種不同點數，以出現的高點數的兩倍計分

參考範例：畫出門檻函數



門檻函數

如果橫座標的值小於
threshold，縱座標值為0，
否則為1

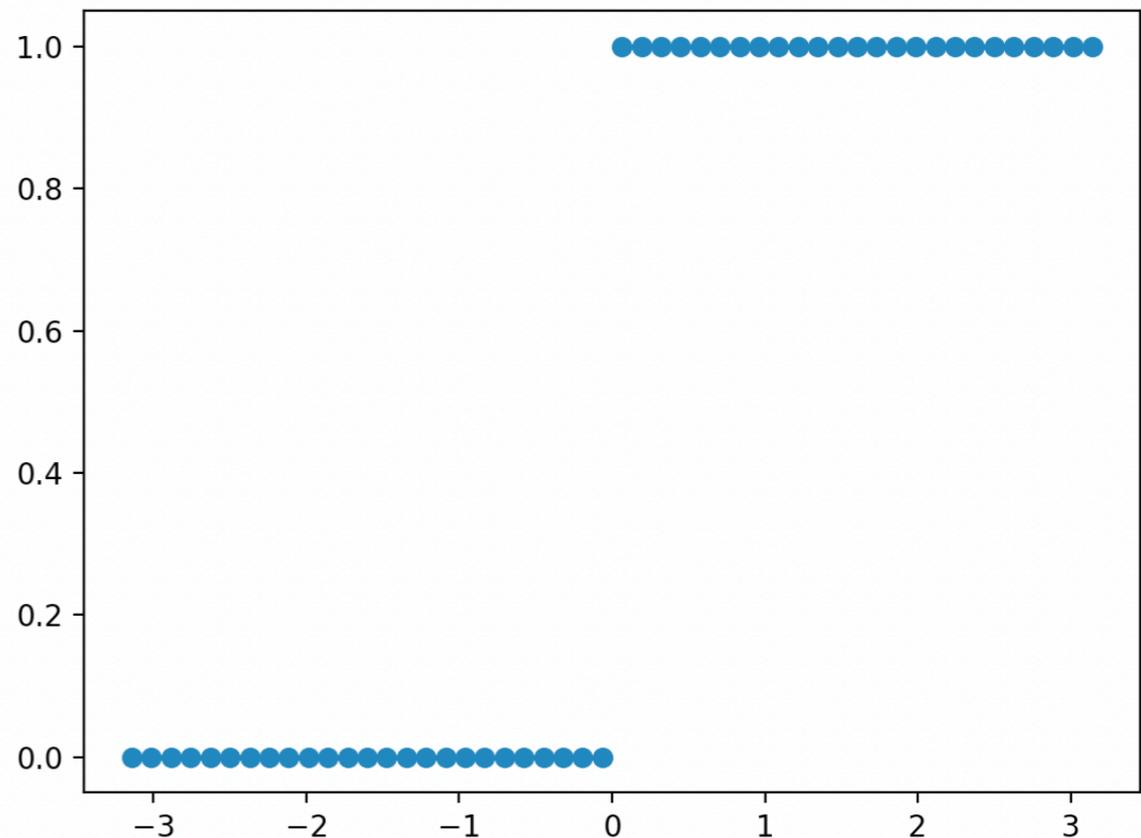
```
import numpy as np
import matplotlib.pyplot as plt
```

```
x = np.linspace(-np.pi,np.pi)
threshold = 0
y = []
for value in x:
    if value >= threshold:
        y.append(1)
    else:
        y.append(0)
plt.figure()
plt.scatter(x,y,marker='o')
plt.show()
```

門檻函數

如果橫坐標的值小於
threshold，縱座標值為0，
否則為1

門檻函數
為非線性
函數



```
import numpy as np
import matplotlib.pyplot as pl
```

```
x = np.linspace(-np.pi,np.pi)
threshold = 0
y = []
for value in x:
    if value >= threshold:
        y.append(1)
    else:
        y.append(0)
pl.figure()
pl.scatter(x,y,marker='o')
pl.show()
```

匯入numpy，以
np代表套件名稱

使用np的特性，np.pi代表
 π ，將區間 $[-\pi, \pi]$ 等距切
割，取得50個切割值
linspace代表等距切割方法

```
import numpy as np
import matplotlib.pyplot as pl
```

```
x = np.linspace(-np.pi,np.pi)
```

```
threshold = 0
```

```
y = []
```

```
for value in x:
```

```
    if value >= threshold:
```

```
        y.append(1)
```

```
    else:
```

```
        y.append(0)
```

```
pl.figure()
```

```
pl.scatter(x,y,marker='o')
```

```
pl.show()
```

依照順序，將串列x中的元素代入value
執行迴圈指令

```
import numpy as np
import matplotlib.pyplot as pl
```

```
x = np.linspace(-np.pi,np.pi)
```

```
threshold = 0
```

```
y = []
```

```
for value in x:
```

```
    if value >= threshold:
```

```
        y.append(1)
```

```
    else:
```

```
        y.append(0)
```

```
pl.figure()
```

```
pl.scatter(x,y,marker='o')
```

```
pl.show()
```

條件指令，如果value大於或等於threshold，將1附加在串列y中，否則將0附加在串列y中

```
import numpy as np
import matplotlib.pyplot as plt
```

匯入套件，以plt
代表套件名稱

```
x = np.linspace(-np.pi,np.pi)
threshold = 0
y = []
for value in x:
    if value >= threshold:
        y.append(1)
    else:
        y.append(0)
```

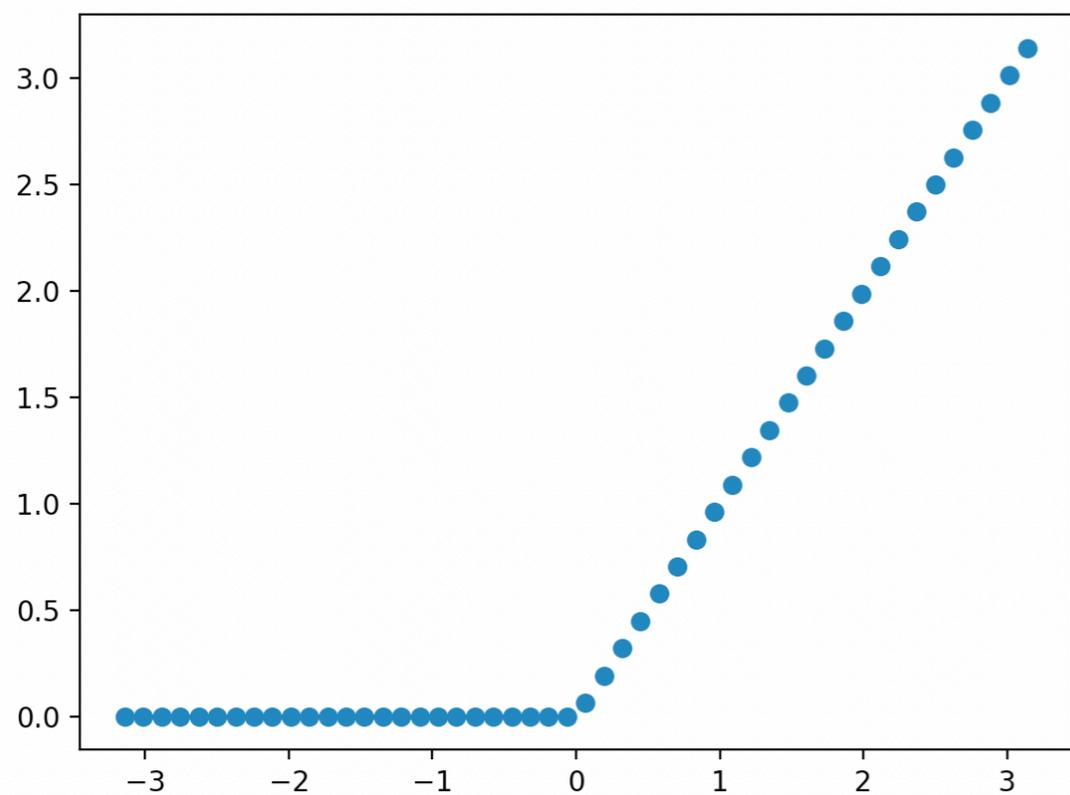
使用套件plt中方法繪製圖
形，plt.figure()新增一張圖

```
plt.figure()
plt.scatter(x,y,marker='o')
plt.show()
```

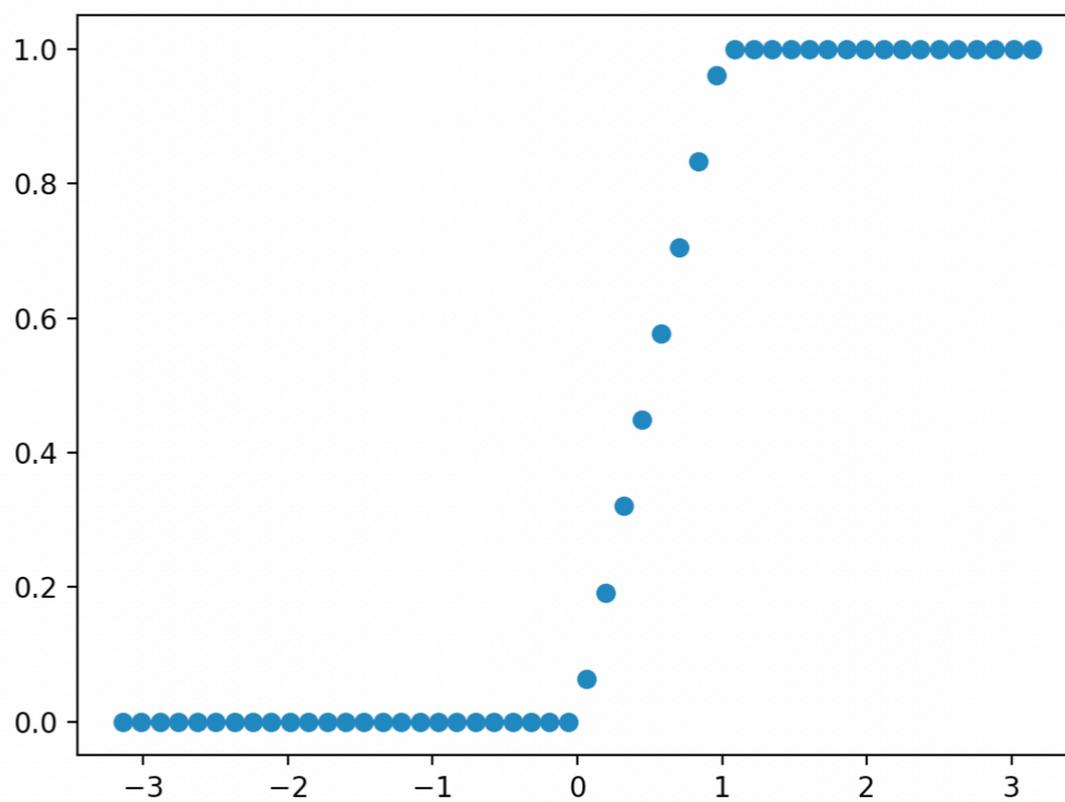
使用套件plt中方法，
plt.scatter()繪製點狀圖。使
用時傳入三個參數，串列x
代表橫座標的值，串列y代
表縱座標的值，使用'o'標示
圖點

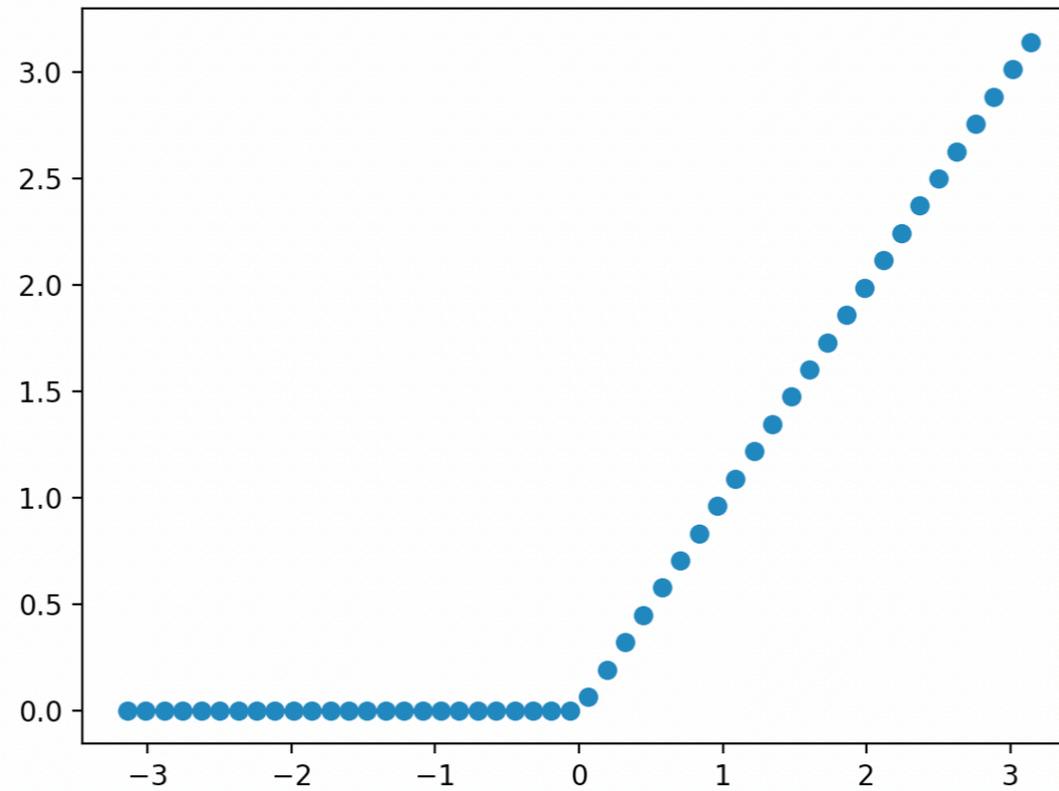
在螢幕上顯示圖形

練習.請畫出ReLU函數以
及BiLinear函數



ReLU函
數，是卷積
神經網路常
用的非線性
函數





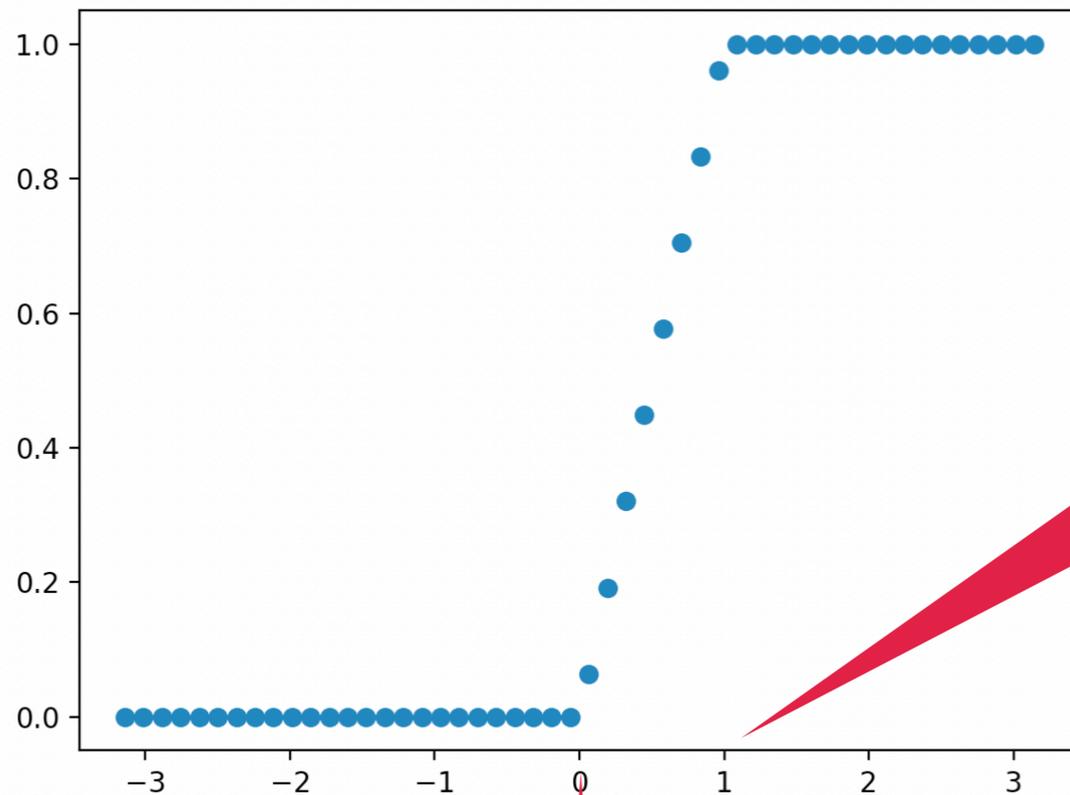
橫座標值大於門檻值
threshold，則函數值等於
輸入的橫座標值，否則函
數值為0

ReLU 提示

參考範例 slide 12

- 參考繪製門檻函數的指令
當橫座標值大於等於門檻值
時，將1附加在串列中
- 繪製ReLU函數需要改變附加值
- 將`pl.figure()`修改為`pl.figure(1)`

```
y = []  
for value in x:  
    if value >= threshold:  
        y.append(1)  
    else:  
        y.append(0)  
pl.figure()  
pl.scatter(x,y,marker='o')
```



BiLinear函數的上界門檻值，
threshold_h，高於此門檻，
函數值為1

BiLinear函數的下界門檻值，
threshold_l，低於此門檻，函數值為0

BiLinear 提示

參考範例 slide 12

- 設定threshold_l為0
設定threshold_h為1
- 使用if elif else
- 將pl.figure()修改為pl.figure(2)

```
threshold = 0
```

```
y = []
```

```
for value in x:
```

```
    if value >= threshold:
```

```
        y.append(1)
```

```
    else:
```

```
        y.append(0)
```

```
pl.figure()
```

```
pl.scatter(x,y,marker='o')
```

if elif else

如果cond1成立，執行
statement 1

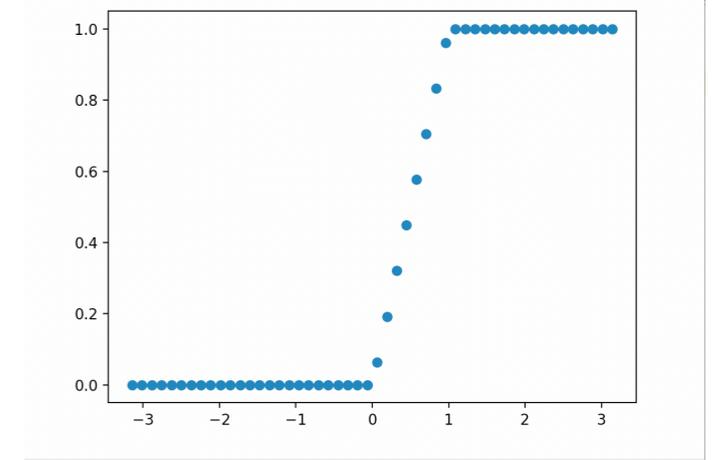
```
if cond1:  
    statement 1  
elif cond2:  
    statement 2  
else:  
    statement 3
```

否則如果cond2成立，執行
statement 2

否則，執行
statement 3

如果cond1成立，執行
y.append(1)

BiLinear 提示



比較value與
threshold_h，設
計cond1

比較value與
threshold_l，設
計cond2

改變串列y的附加
內容，設計
statement

```
if cond1:  
    y.append(1)  
elif cond2:  
    y.append(0)  
else:  
    statement
```

否則，如果
cond2成立，執
行y.append(0)

否則，執行
statement