

# IEEM535300

## 智慧化企業整合

### Activation functions

王禹晴

# Outline

- 1 線性系統與非線性系統
- 2 激活函數(Activation functions)之介紹
- 3 常用的激活函數(Activation functions)
- 4 使用Python構建激活函數(Activation functions)

# Outline

- 1 線性系統與非線性系統
- 2 激活函數(Activation functions)之介紹
- 3 常用的激活函數(Activation functions)
- 4 使用Python構建激活函數(Activation functions)

# 線性系統

Input data



Model

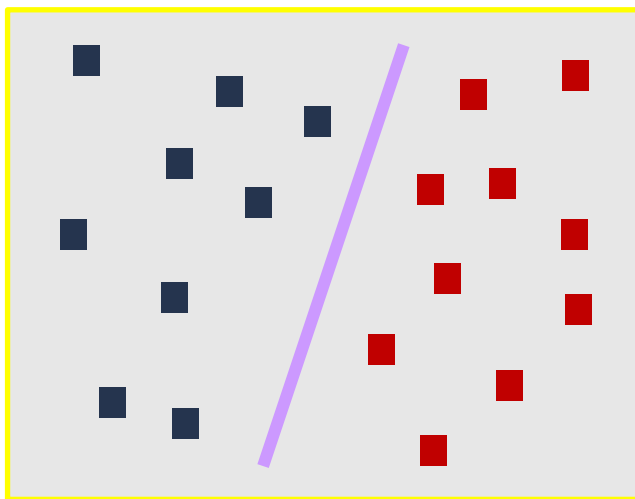
$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Outputs

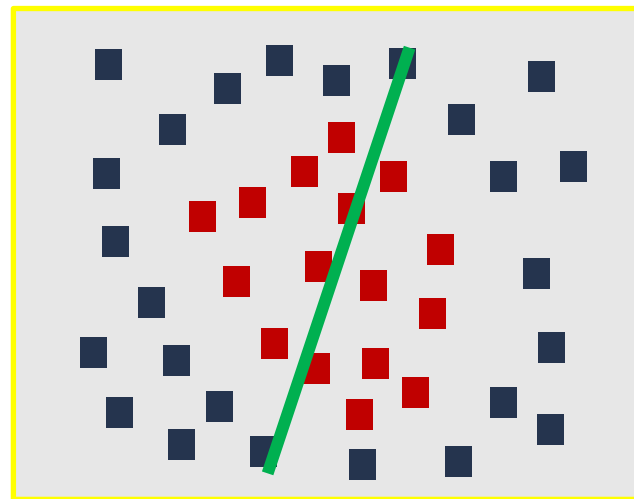
250

# 線性 V.S. 非線性系統

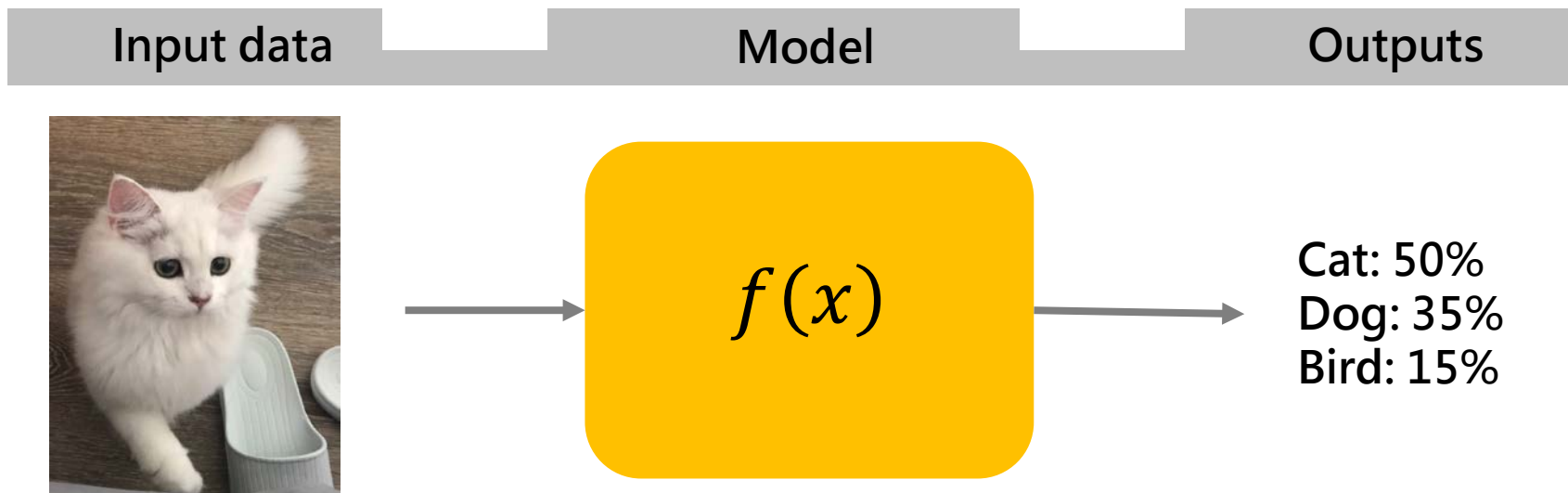
Separable for linear system



Inseparable for linear system



# Deep Learning

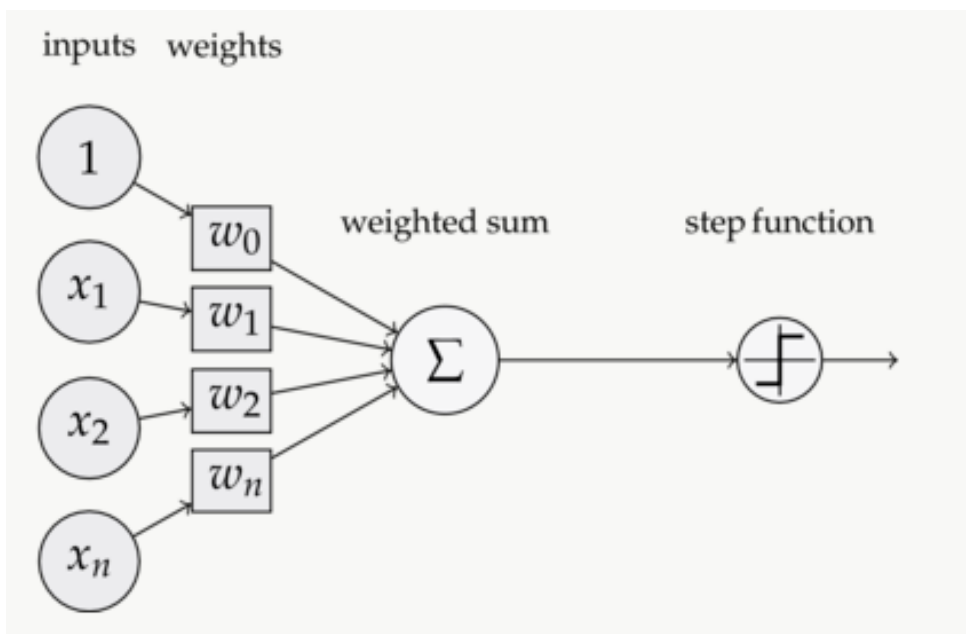


# Outline

- 1 線性系統與非線性系統
- 2 激活函數(Activation functions)之介紹
- 3 常用的激活函數(Activation functions)
- 4 使用Python構建激活函數(Activation functions)

# 激活函數(Activation functions)之介紹

- 在神經元裡定義一些權重(weight)，經過計算之後，判斷結果是否超過一個閾值(threshold)，如果超過，神經元輸出為 1；反之，則輸出 0。

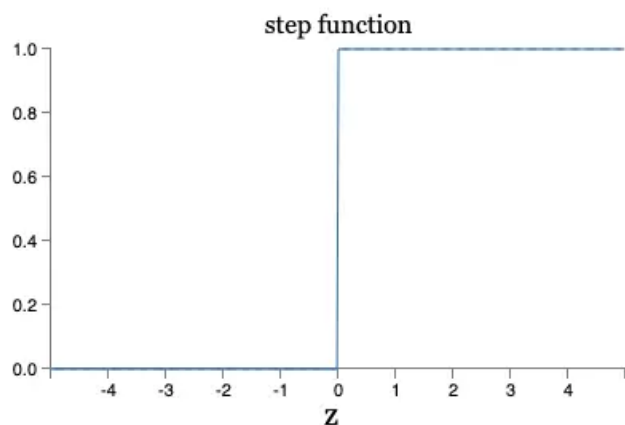


$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases}$$



# 激活函數(Activation functions)之介紹

- 當調整參數時，節點輸出的值在 0 和 1 之間躍遷，該輸出是一個階梯函數，這會對調參造成很大不便。



希望輸出的函數變得平滑一些

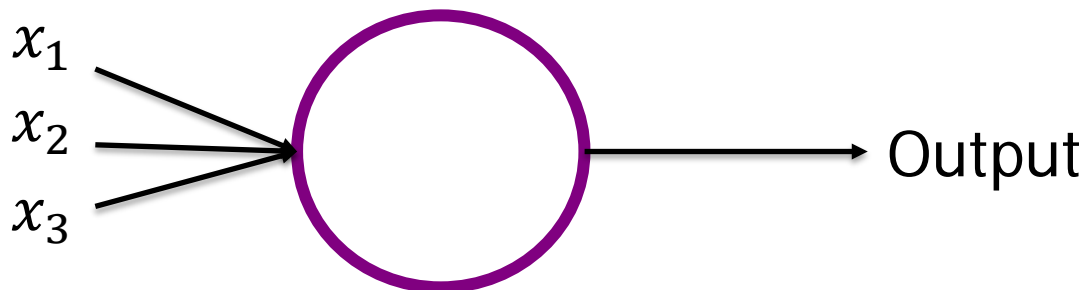


激活函數

- 如果不用激活函數，每一層output都是上層input的線性函數，無論神經網絡有多少層，輸出都是輸入的線性組合。

# 激活函數(Activation functions)之介紹

- 激活函數(Activation functions)：對於人工神經網絡模型去學習、理解非常複雜和非線性的函數來說具有十分重要的作用。
- ✓ 增加非線性程度。
- ✓ 將輸出限制在給定範圍內（模擬數據的概率分佈）

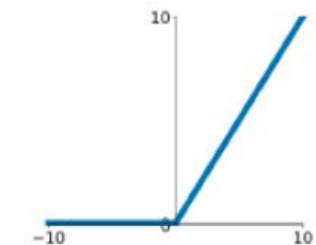
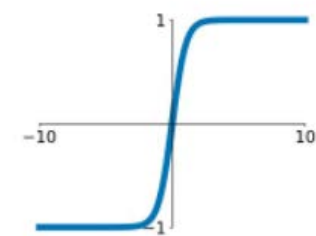
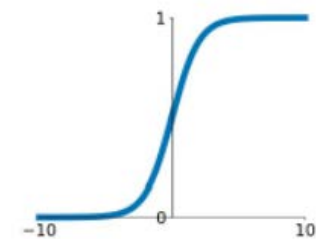


# Outline

- 1 線性系統與非線性系統
- 2 激活函數(Activation functions)之介紹
- 3 常用的激活函數(Activation functions)
- 4 使用Python構建激活函數(Activation functions)

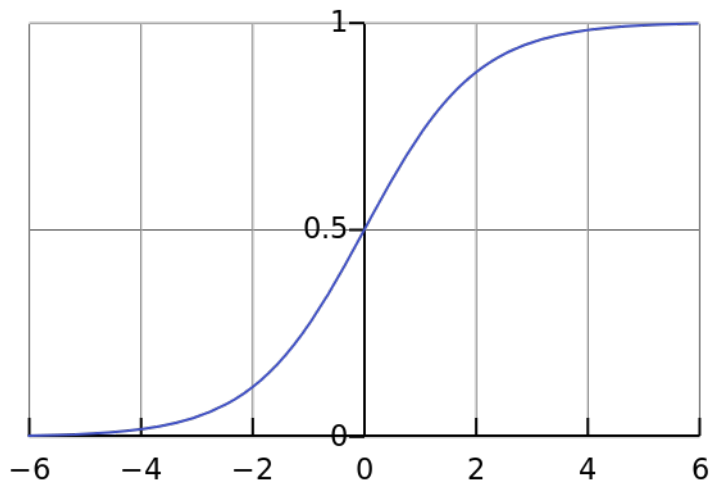
# Common Activation Functions

- Sigmoid
- Tanh
- ReLu
- Leaky Relu
- Softplus (see references)
- Softmax (see references)
- .....



# 常用的激活函數(Activation functions)

## Sigmoid ?



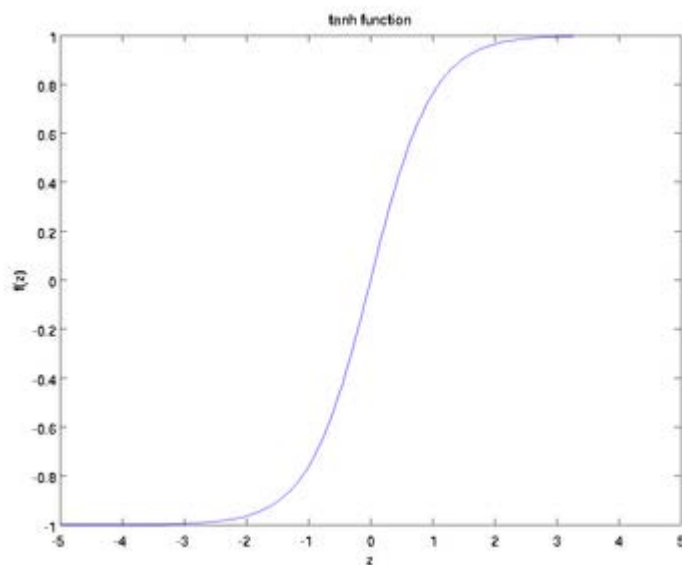
$$A = \frac{1}{1+e^{-x}}$$

- 非線性功能
- 輸出：( 0,1 )
- 處理極端值
- 常用於隱藏層和輸出層
- 常用於二元分類
- 梯度消失問題

# 常用的激活函數(Activation functions)

## Tanh ?

$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

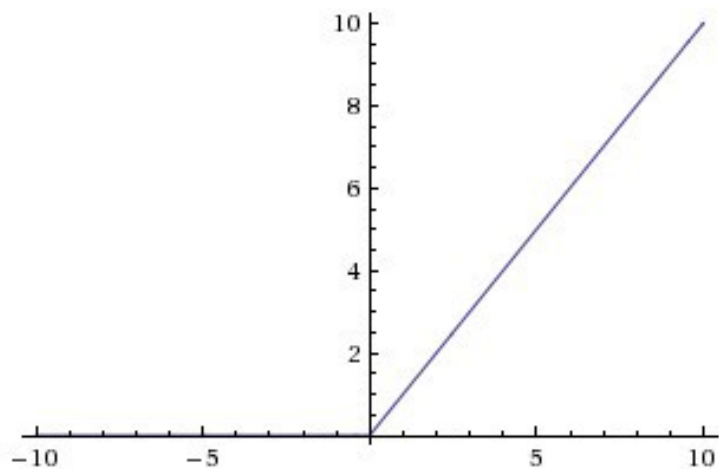


- 非線性功能
- 輸出：( -1,1 )
- 處理極端值
- 常用於隱藏層
- 梯度消失問題

# 常用的激活函數(Activation functions)

## ReLU ?

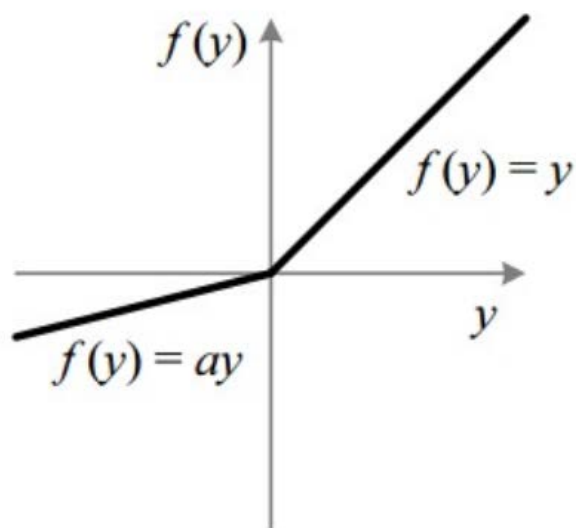
$$A(x) = \max(0, x)$$



- 非線性功能
- 輸出：  $(0, \infty)$
- 捨棄一些信息
- 常用於隱藏層
- **梯度爆炸問題**

# 常用的激活函數(Activation functions)

## Leaky ReLu ?



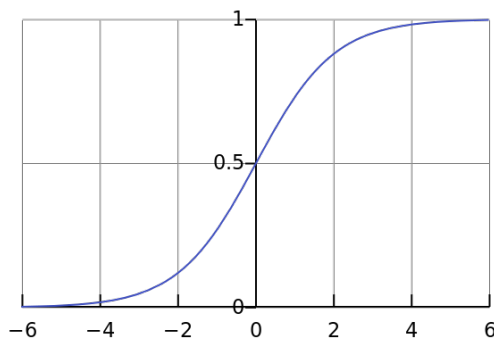
- ReLu的修改版
- 輸入  $< 0$  時保留信息
- 常用於隱藏層
- **梯度爆炸問題**



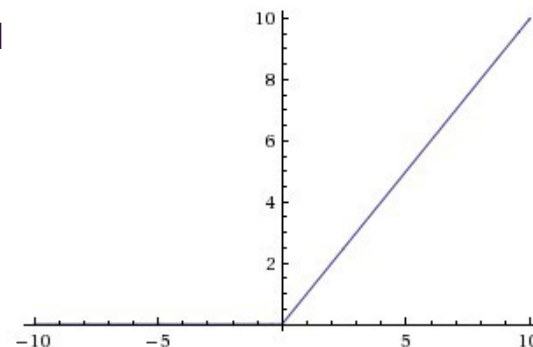
# 常用的激活函數(Activation functions)

## Sigmoid、Tanh、ReLU、Leaky ReLU

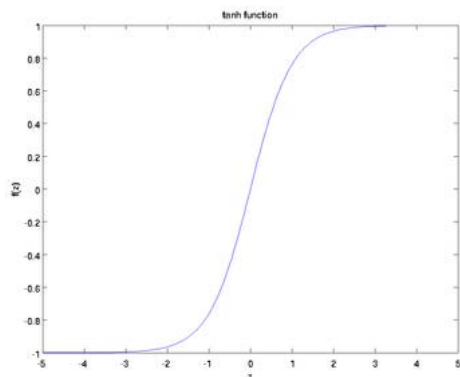
Sigmoid



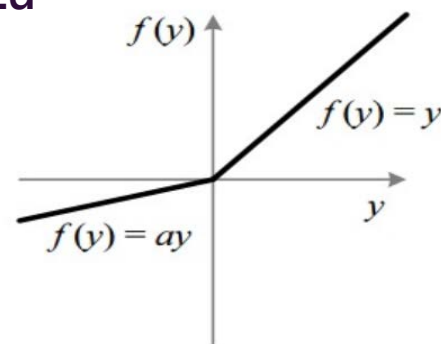
ReLU



Tanh



Leaky ReLU



# Outline

- 1 線性系統與非線性系統
- 2 激活函數(Activation functions)之介紹
- 3 常用的激活函數(Activation functions)
- 4 使用Python構建激活函數(Activation functions)

# 使用Python構建激活函數(Activation functions)

## Define function in Python ?

```
In [1]: def your_function(x):  
        #do something here  
        #do something here  
        #do something here  
  
        y = x**5 + 2*x + 1  
        return y #the output of this function
```

```
In [2]: result = your_function(5678) #call self-defined function  
        print(result)
```

5901697440255953725

# 使用Python構建激活函數(Activation functions)

## Define Tanh ?

```
In [3]: import numpy as np #import Numpy package and renamed it as np
import math #import math package
import matplotlib.pyplot as plt #import the pyplot.py in Matplotlib package,
#and renamed it as plt
```

```
In [4]: def tanh(x):
y = (2/(1+math.exp(-2*x))) - 1 #the formula of hyper tangent function
return y #output the result
```

# 使用Python構建激活函數(Activation functions)

## Generate input & output data ?

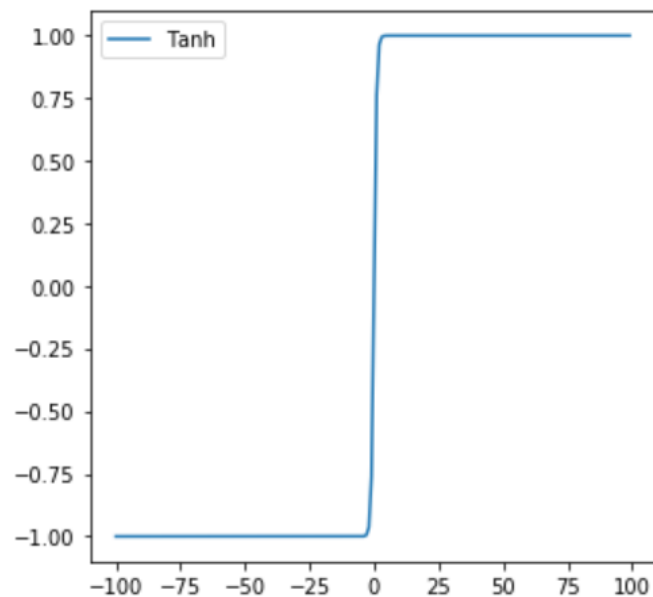
```
In [9]: x = list(range(-100,100,1))  
y = []  
for element in x:  
    y.append(tanh(element))
```

# 使用Python構建激活函數(Activation functions)

## Plot the figure ?

```
In [10]: plt.figure(figsize=(5,5))  
plt.plot(x,y,label='Tanh')  
plt.legend(loc='upper left')
```

```
Out[10]: <matplotlib.legend.Legend at 0x2e1e38d56d8>
```

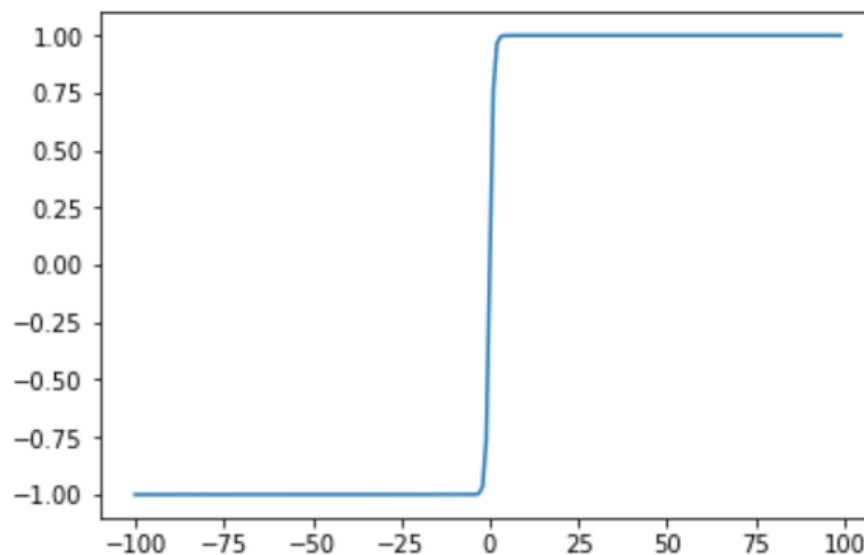


# 使用Python構建激活函數(Activation functions)

## Another way to present ?

```
In [13]: x = np.arange(-100,100,1)
         y = np.tanh(x)
         plt.plot(x,y)
```

```
Out[13]: [<matplotlib.lines.Line2D at 0x2e1e3a5b080>]
```



# References

- [Defining functions in Python](#)
- [Visualization in Python using Matplotlib](#)
- [Understanding activation functions](#)
- [Various activation functions](#)



# Activation functions之課程練習

- 課程練習

1. 請嘗試列出將-5到5等分成100份的一組數據

$$X = \{x, x \in [-5,5]\}$$

2. 使用此數據  $X$  畫出sigmoid function ( $X$ )的圖。

- 今日上課過後解答會上傳至eLearn。

# Activation functions之回家作業

- 回家作業
  1. 請使用「[data\\_Hw2](#)」中的參數來生成9張Leaky ReLu的函數圖。
  2. 必須使用subplot function來生成圖形。
- 請以“學號\_組別\_姓名\_Activation functions.ipynb”命名並上傳至eLearn。

# End

