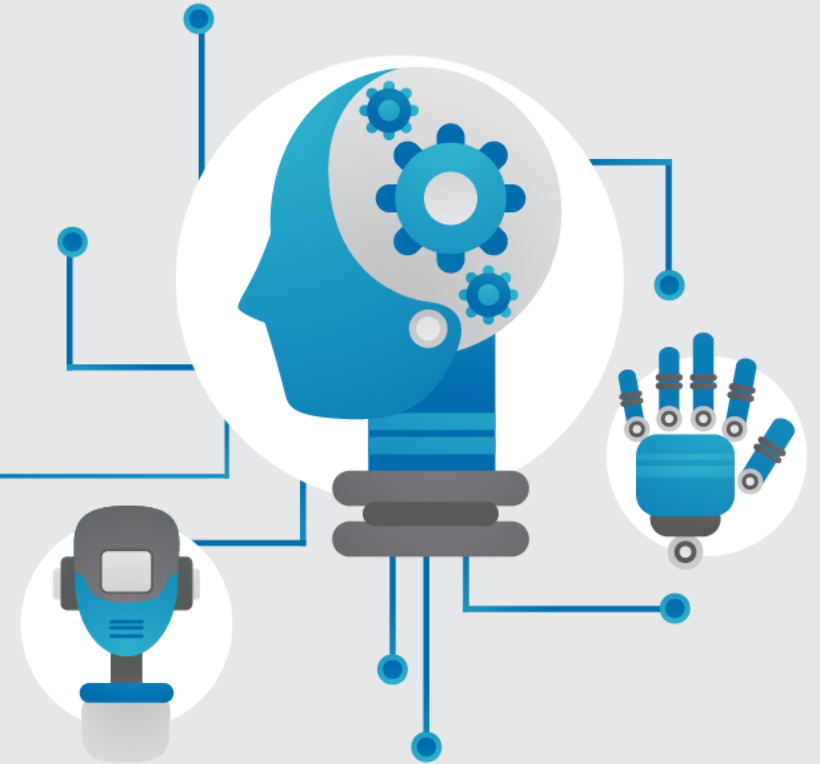


# 神經元結構

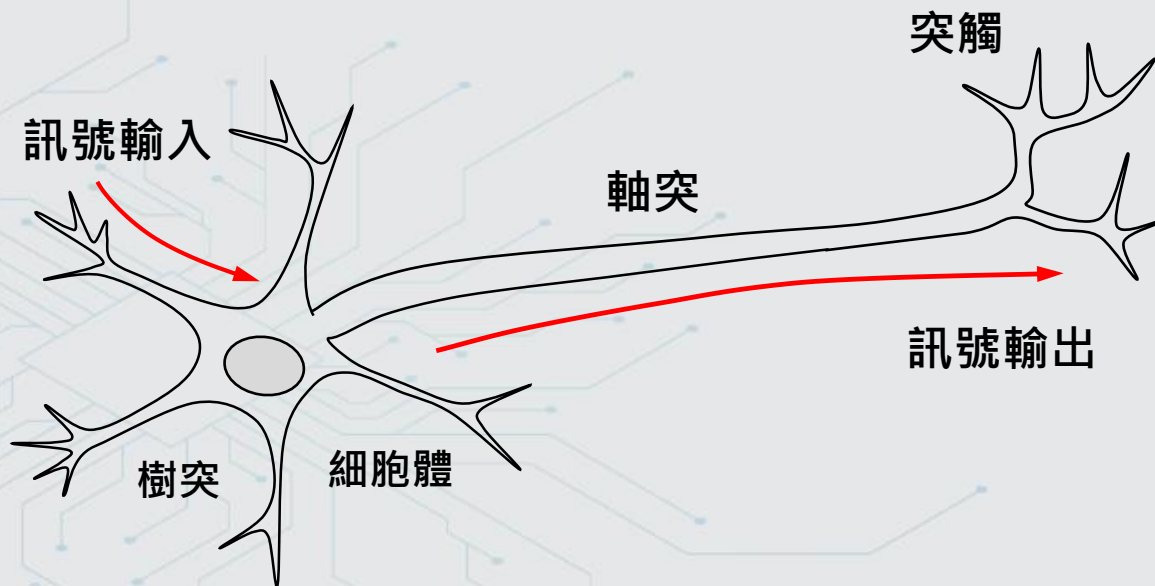




# 神經元細胞



神經元細胞本身會長出像樹枝形狀的**樹突**（dendrite），右邊伸出去的是神經細胞的**軸突**（axon）。  
**樹突**負責**接收訊號**，**軸突**負責把訊號**傳遞出去**，  
兩個神經元的軸突和樹突之間，會有神經**突觸**（synapse）。

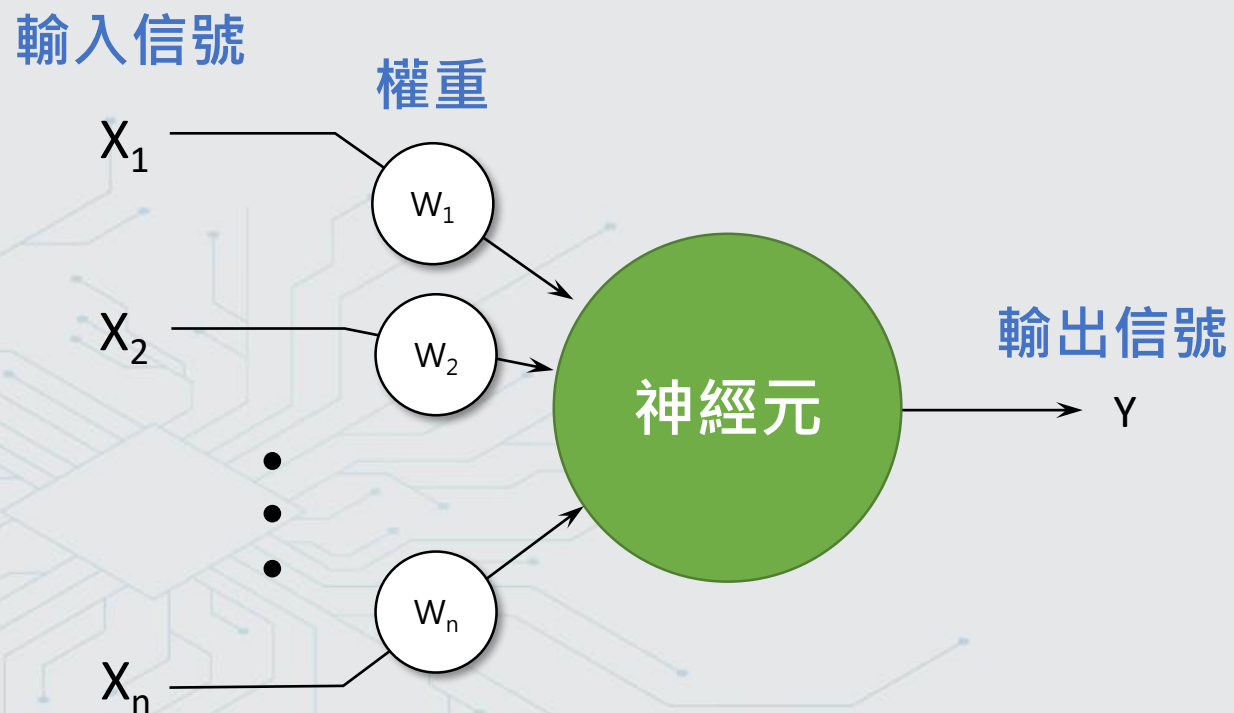




# 神經元運算模型



1943年，Warren McCulloch和Walter Pitts提出模仿神經元的運算模型。



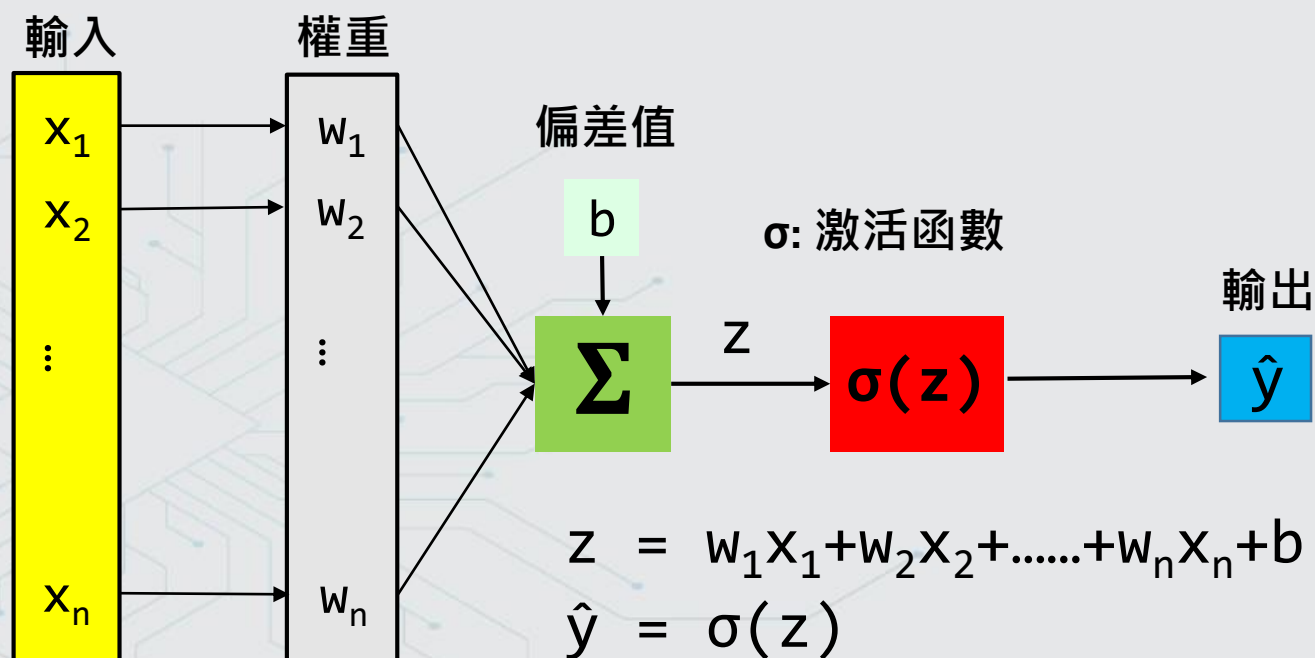


# 感知器



1957年，Frank Rosenblatt提出**感知器**（Perceptron）結構實現神經元模型。

感知器從前端接收訊息，將這些訊息依據**權重**（weight）加權後加總，再加上**偏差值**（bias），透過**激活函數**（activation function）轉換成新訊息傳送出去。





# 感知器學習演算法



➡ 目的：透過資料訓練，調整權重使誤差最少。

➡ 學習演算法的步驟

1. **初始值**：設定權重 $w_1, w_2, \dots, w_n$ 、偏差值 $b$ 和閾值 $\theta$ 的初始值。
2. **計算輸出值**：根據輸入值 $x_1, x_2, \dots, x_n$ 、權重 $w_1, w_2, \dots, w_n$ 和閾值 $\theta$ 計算感知器的輸出值  $y$ 。
3. **調整權重**：
$$w_i^{t+1} = w_i^t + \Delta w_i^t$$
$$\Delta w_i^t = \alpha \times x_i^t \times (\hat{y} - y) \quad (\alpha : \text{學習率常數})$$
4. **重複步驟2至3**。



# 感知器範圍說明



我們用一個感知器實作AND運算：

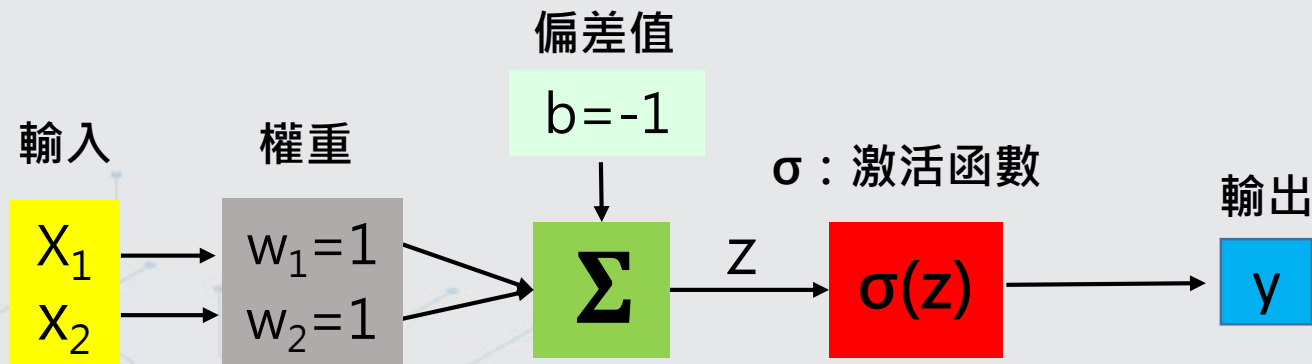
輸入端		輸出端
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



# 感知器範例說明 - 初始值



設定權重 $w_1=1$ ,  $w_2=1$ 、偏差值 $b = -1$ 和閾值 $\theta = 0$



$$z = 1x_1 + 1x_2 - 1$$

$$y = \sigma(z) = \begin{cases} 1 & \text{if } z \geq \theta = 0 \\ 0 & \text{if } z < 0 \end{cases}$$



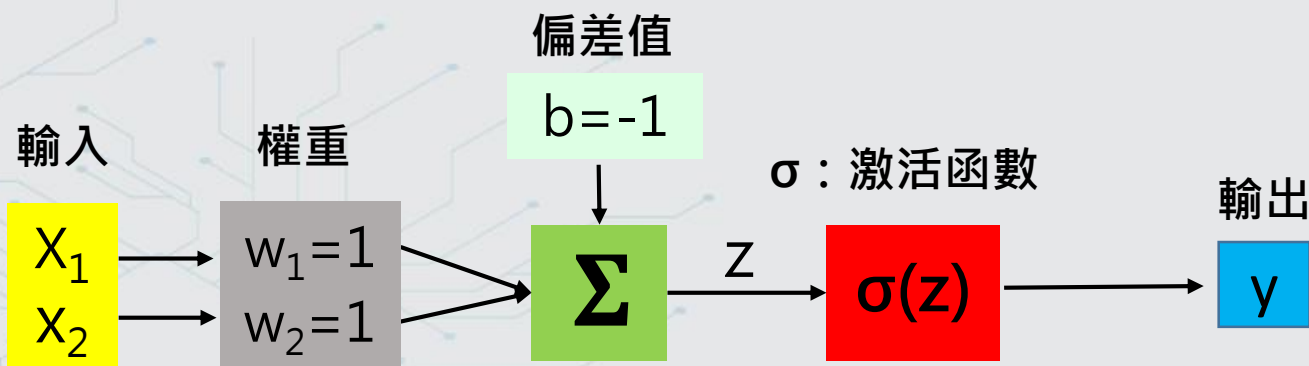
# 感知器範例說明 - 計算輸出值

機器學習實務



➔ 輸入訓練資料

Epoch	輸入		期望輸出	初始權重		實際輸出	誤差
	$x_1$	$x_2$	$\hat{y}$	$w_1$	$w_2$	$y$	$e$
1	0	0	0	1	1	0	0



$$z = 1x_1 + 1x_2 - 1$$
$$y = \sigma(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$





# 感知器範例說明 - 調整權重

機器學習實務



➔ 學習率常數  $\alpha = 0.1$

Epoch	輸入		期望輸出	初始權重		實際輸出	誤差	最終權重	
	$x_1$	$x_2$	$\hat{y}$	$w_1$	$w_2$	$y$	$e$	$w_1$	$w_2$
1	0	0	0	1	1	0	0	1	1

$$w_1^1 = w_1^1 + \alpha \times x_1^1 \times (\hat{y} - y) = 1 + 0.1 \times 0 \times 0 = 1$$

$$w_2^1 = w_2^1 + \alpha \times x_2^1 \times (\hat{y} - y) = 1 + 0.1 \times 0 \times 0 = 1$$



# 感知器範例說明 - 調整權重



➔ 學習率常數  $\alpha = 0.1$

Epoch	輸入		期望輸出	初始權重		實際輸出	誤差	最終權重	
	$x_1$	$x_2$	$\hat{y}$	$w_1$	$w_2$	$y$	$e$	$w_1$	$w_2$
1	0	0	0	1	1	0	0	1	1
	0	1	0	1	1	1	-1	1	0.9

$$w_1^2 = w_1^2 + \alpha \times x_1^2 \times (\hat{y} - y) = 1 + 0.1 \times 0 \times -1 = 1$$

$$w_2^2 = w_2^2 + \alpha \times x_2^2 \times (\hat{y} - y) = 1 + 0.1 \times 1 \times -1 = 0.9$$



# 感知器範例說明 - 調整權重

機器學習實務



➔ 學習率常數  $\alpha = 0.1$

Epoch	輸入		期望輸出	初始權重		實際輸出	誤差	最終權重	
	$x_1$	$x_2$	$\hat{y}$	$w_1$	$w_2$	$y$	$e$	$w_1$	$w_2$
1	0	0	0	1	1	0	0	1	1
	0	1	0	1	1	1	-1	1	0.9
	1	0	0	1	0.9	1	-1	0.9	0.9

$$w_1^3 = w_1^2 + \alpha \times x_1^2 \times (\hat{y} - y) = 1 + 0.1 \times 1 \times -1 = 0.9$$

$$w_2^3 = w_2^2 + \alpha \times x_2^2 \times (\hat{y} - y) = 0.9 + 0.1 \times 0 \times -1 = 0.9$$



# 感知器範例說明 - 調整權重



➡ 學習率常數  $\alpha = 0.1$

Epoch	輸入		期望輸出	初始權重		實際輸出	誤差	最終權重	
	$x_1$	$x_2$	$\hat{y}$	$w_1$	$w_2$	$y$	$e$	$w_1$	$w_2$
1	0	0	0	1	1	0	0	1	1
	0	1	0	1	1	1	-1	1	0.9
	1	0	0	1	0.9	1	-1	0.9	0.9
	1	1	1	0.9	0.9	1	0	0.9	0.9

$$w_1^4 = w_1^3 + \alpha \times x_1^3 \times (\hat{y} - y) = 0.9 + 0.1 \times 1 \times 0 = 0.9$$

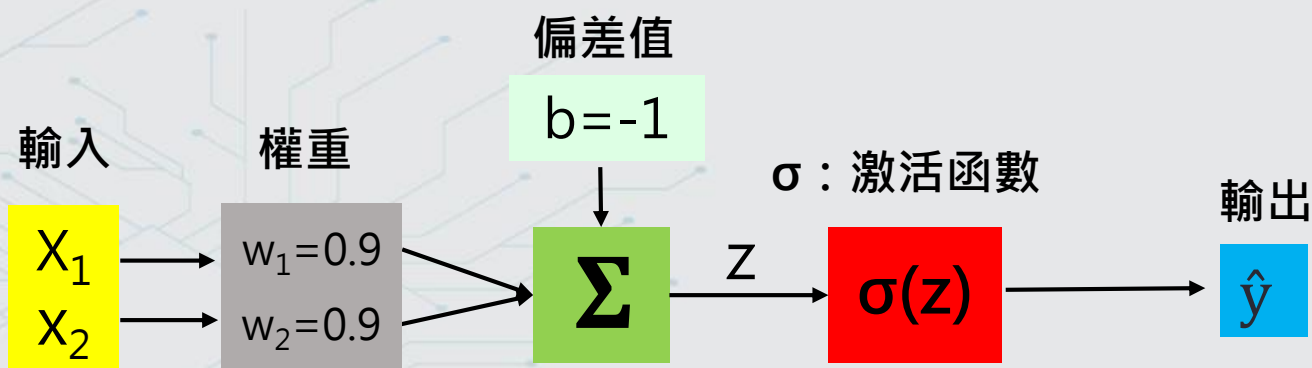
$$w_2^4 = w_2^3 + \alpha \times x_2^3 \times (\hat{y} - y) = 0.9 + 0.1 \times 1 \times 0 = 0.9$$



# 神經元範例說明



輸入端		輸出端
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



$$z = 0.9x_1 + 0.9x_2 - 1$$
$$\hat{y} = \sigma(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$



# 類神經網路 (Artificial Neural Network)

機器學習實務



很多個神經元連接成網路 (network)

