

Day 70

深度學習理論與實作

# 深度神經網路的基礎知識



出題教練

陳明佑 / 聯成數網

# 大綱

---

- 深度學習網路歷史發展
- 為什麼需要深度學習網路
- 深度學習網路架構
- 深度學習網路優缺點

# 深度學習網路歷史

- 最早一開始發展為

單層感知器

Perceptron



多層感知器

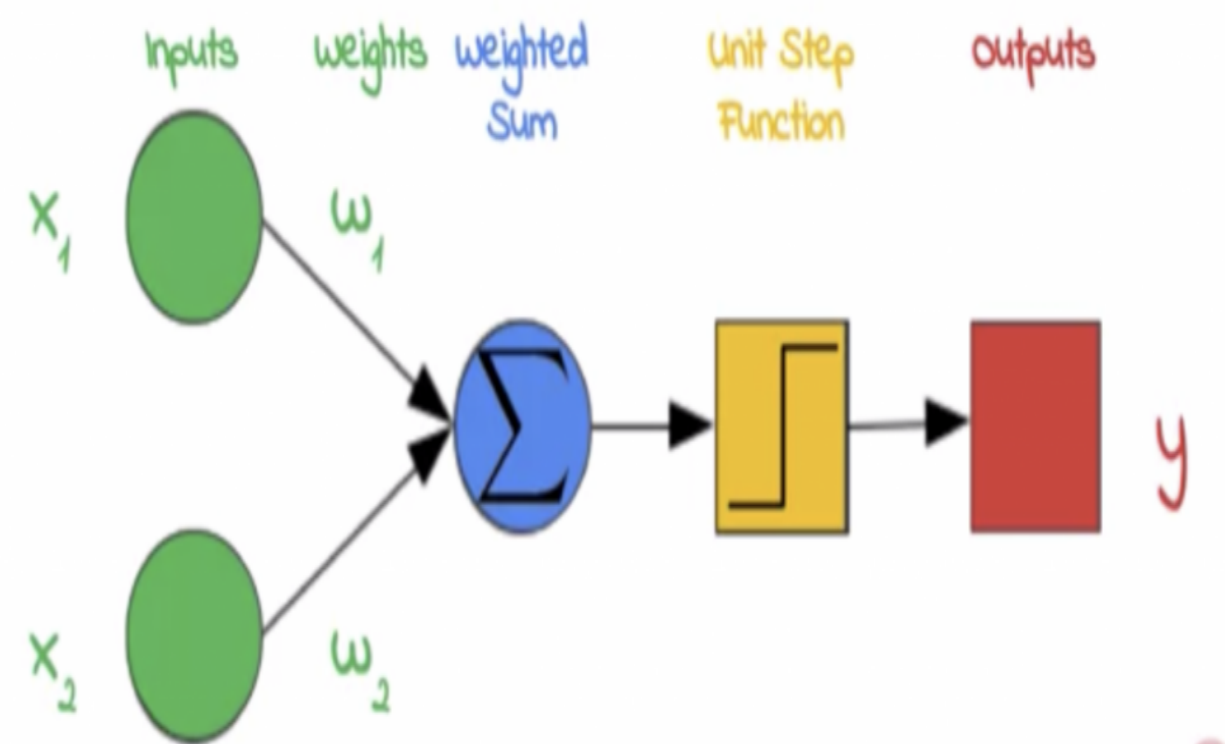
Multi layer Perceptron



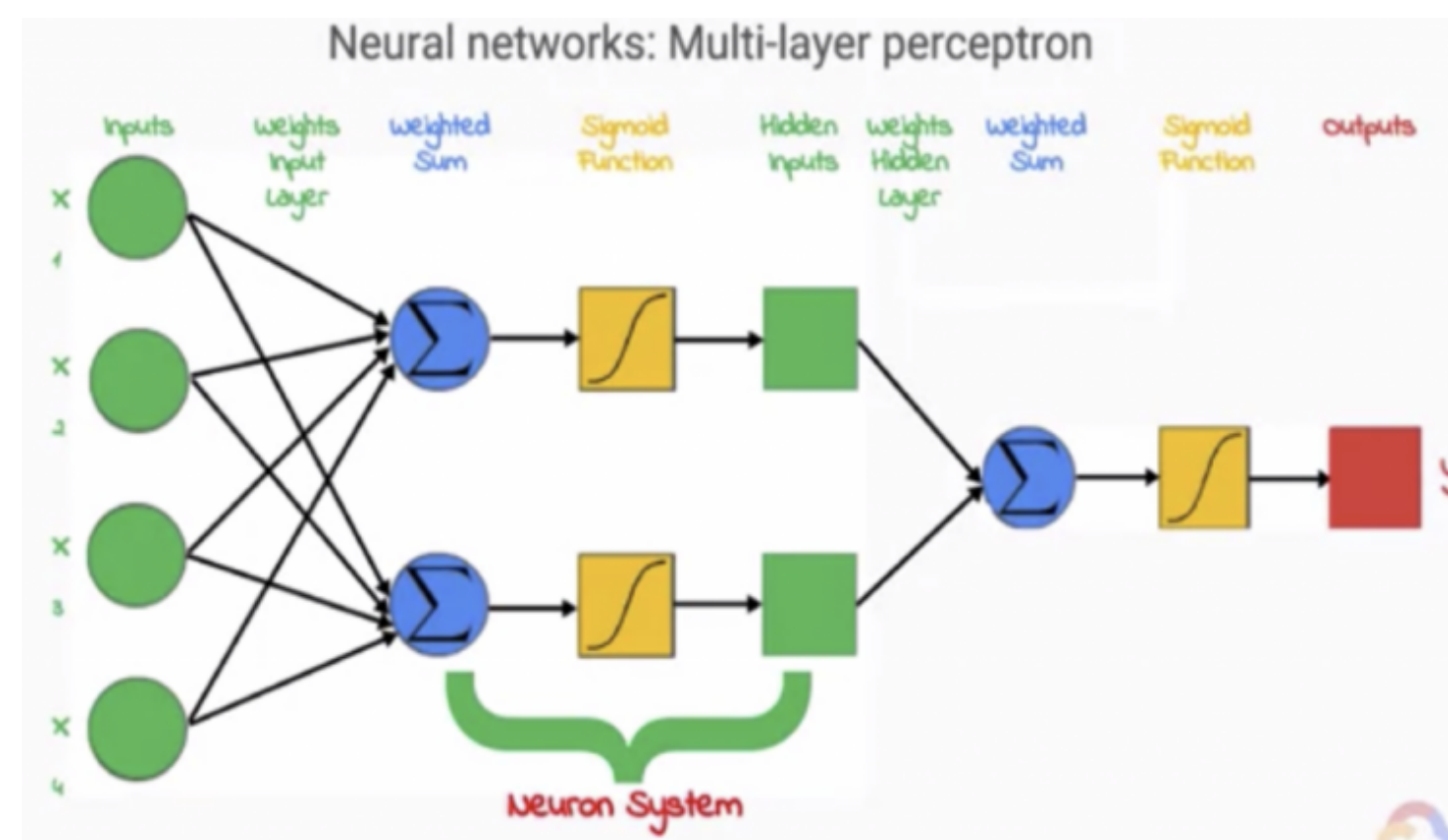
深度學習網路

DNN

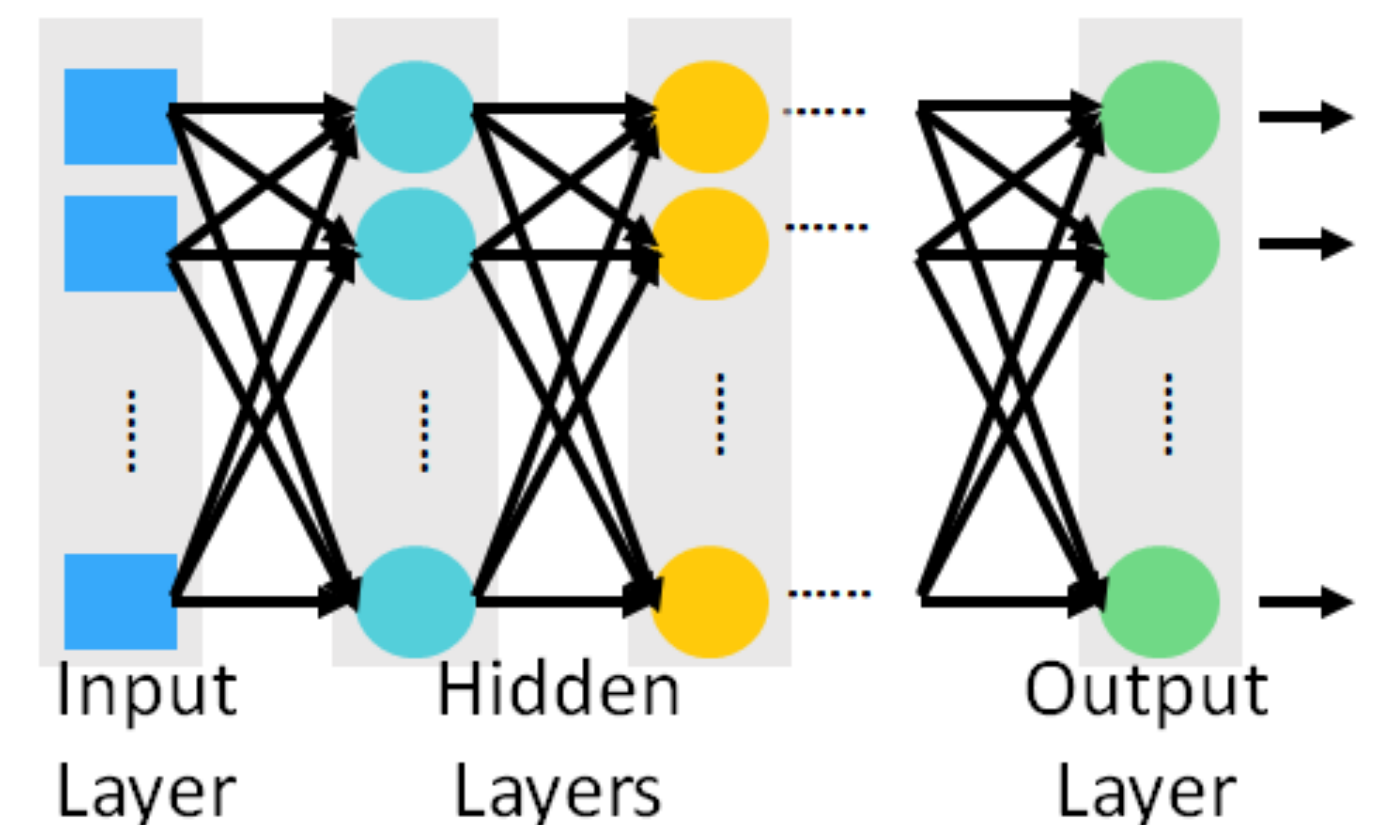
- 而多層感知器與深度學習網路主要差異為可加入多層隱藏層建構出一個具有深度的神經



Perceptron



Multi layer Perceptron



DNN



# 深度學習網路歷史

---

- 深度學習本身是機器學習領域下的一個分支，核心概念為：  
「從資料中尋找一組最適合解決某種特定問題的函式」

## 語音辨識

輸入一段語音，機器能辨識出語音對應的文字

$$f(\text{[Audio Waveform]}) = \text{"How are you"}$$

## 圖像辨識

輸入一張貓咪圖片，機器能分辨出這張圖裡有貓

$$f(\text{[Cat Image]}) = \text{"Cat"}$$

# 深度學習網路歷史

---

## 棋藝競賽

輸入圍棋的落子狀況，機器能算出勝率最高的下個落子點

$$f(\text{圍棋落子狀況}) = \text{“5-5” (next move)}$$



## 對話系統

在對話系統中，機器接受到 Hi，它能作出一個適當的回應，例如Hello

$$f(\text{“Hi” (what the user said)}) = \text{“Hello” (system response)}$$

# 為什麼需要深度學習網路

---

- 手寫辨識照片利用單一層隱藏層神經網路能達到97%以上的正確率，但近年來不斷訓練僅能達到接近97%的正確率
- 主要因為神經網路同時須考慮到照片大小、照片裡出現的圖像、像素及RGB等問題，因此需要較大的神經網路才能學習到更多照片裡的資訊
- 在一個語音辨識的測驗中，無論是淺的或深度神經網路，辨識率都隨著神經元數目的增加而成長  
在相同數目的神經元時，深度神經網路的表現總是比較好

# 深度學習網路概念

---

Define a set of Functions



Evaluate the Functions



Pick the best Function

最終目標

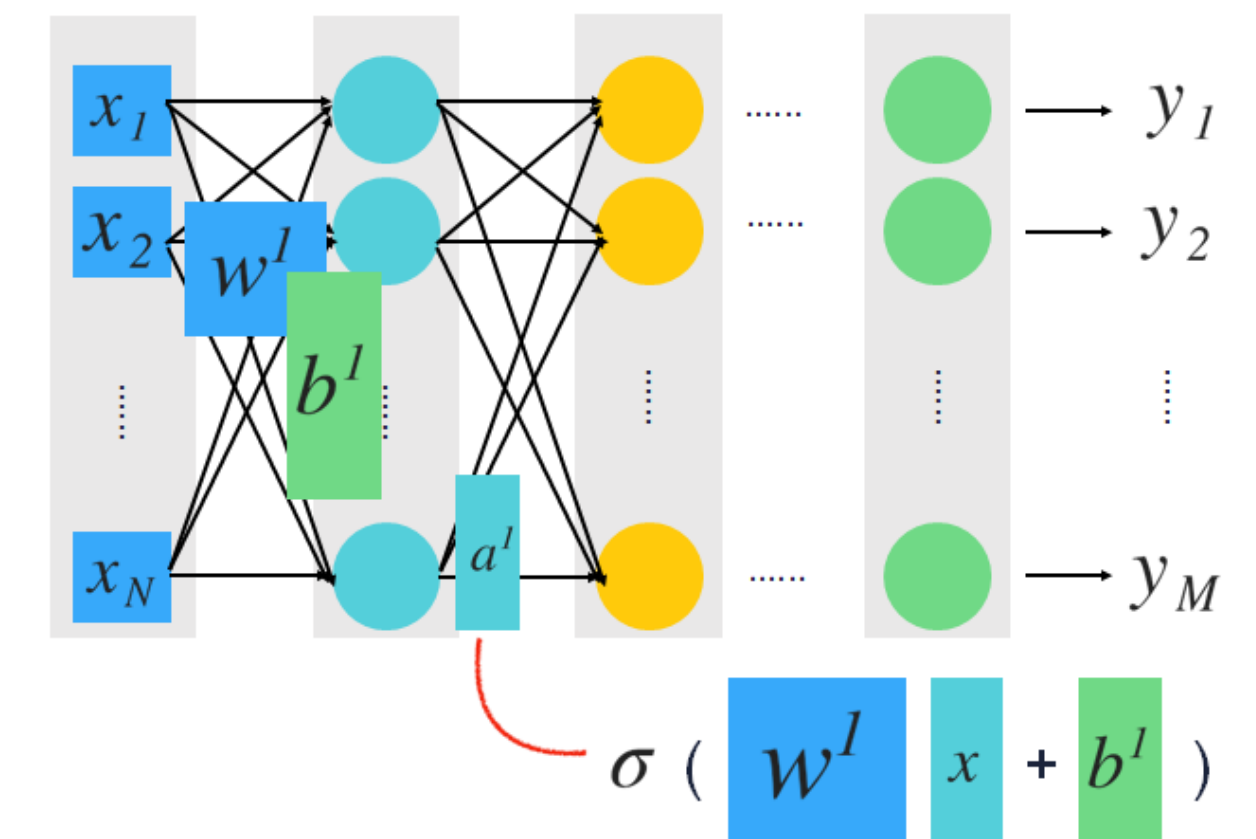
從資料中尋找一組 最適合解決某種特定問題的函數

# 深度學習網路步驟1-Define a set of Functions

- 函數定義方式:由這一層神經元接收上一層神經元的輸入，經過計算(  $W \cdot x + b$  )後可以得到一個輸出結果 $a^l$ ，而 $a^l$ 又會成為下一層的輸入
- $a^l$ 向量為一個函數，當函數只要 $x$ 數量有增減或是 $W$ (權重) /  $b$ (偏移項) 發生改變，整個神經網路的函數內容就會改變
- 當神經網路的神經元越多，則神經網路的候選函數就越多，對問題的解釋能力就越強

## 核心概念

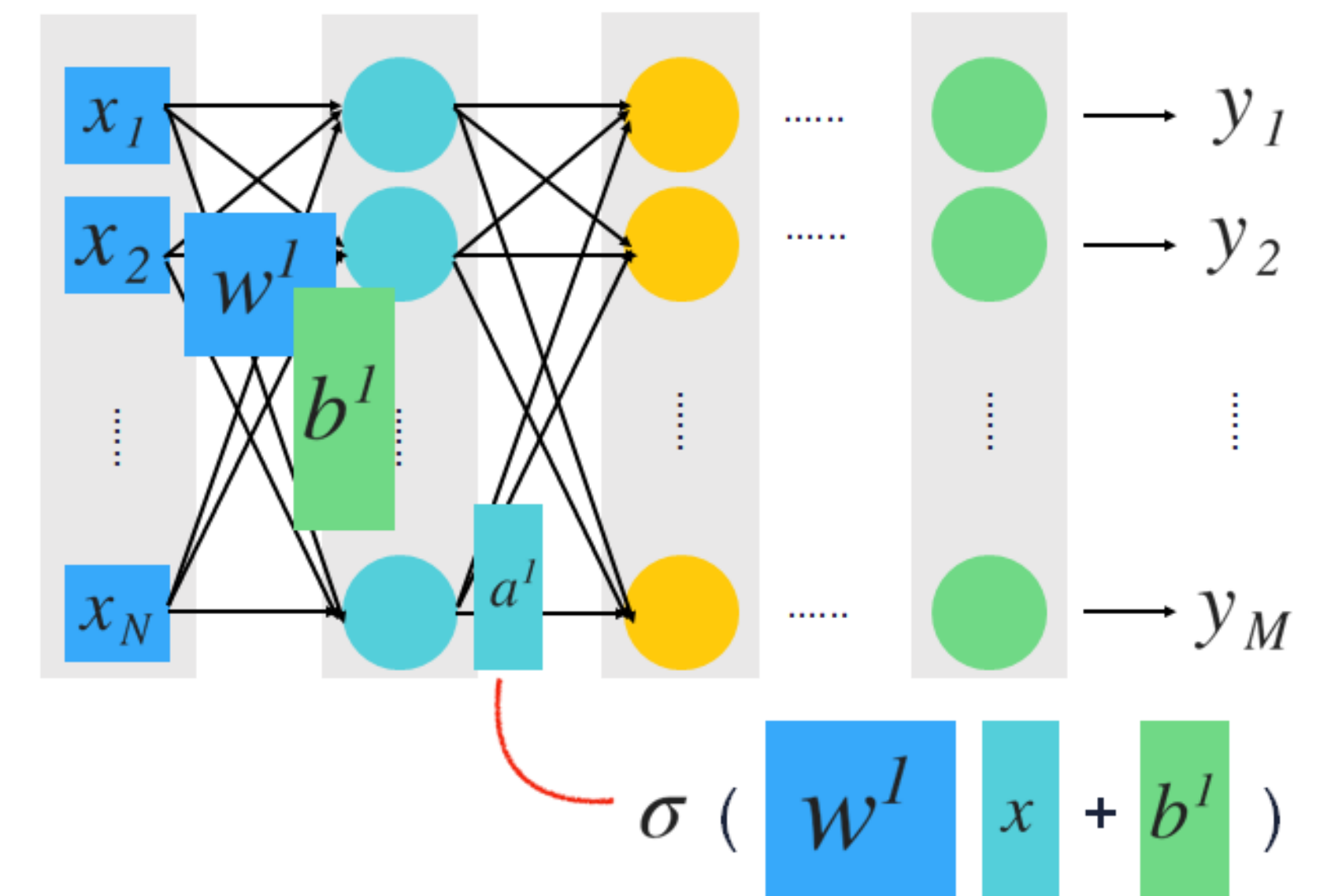
- 每一層的神經元數量或連接方式改動，就會產生不同的函式集合
- 每次權重一旦改動，就會產生一組新的函數





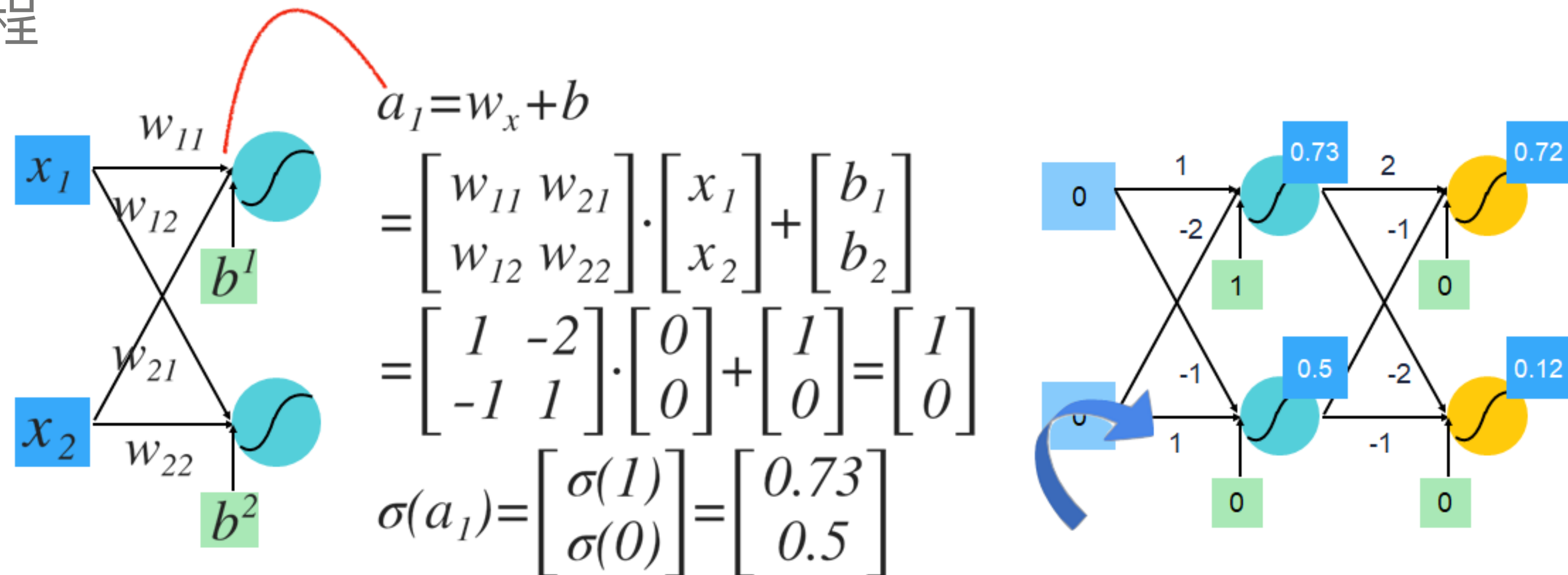
# 神經元傳輸過程(1)

- $[x_1, x_2, \dots, x_n]$  是資料輸入層，神經元接受來自  $[X_1 \dots X_n]$  輸入層的資訊，經過  $W$  權重相乘後加總，加上 Bias 偏移項之後再經過激活函數輸出  $Y$
- 整個運算過程都是數值運算，所以傳入的內容必需要轉化為數值
- 每一層的輸出內容，皆來自前一層權重矩陣與輸入值向量與偏移量的計算結果 -> 輸出向量  $a^1$  為一個函數型式



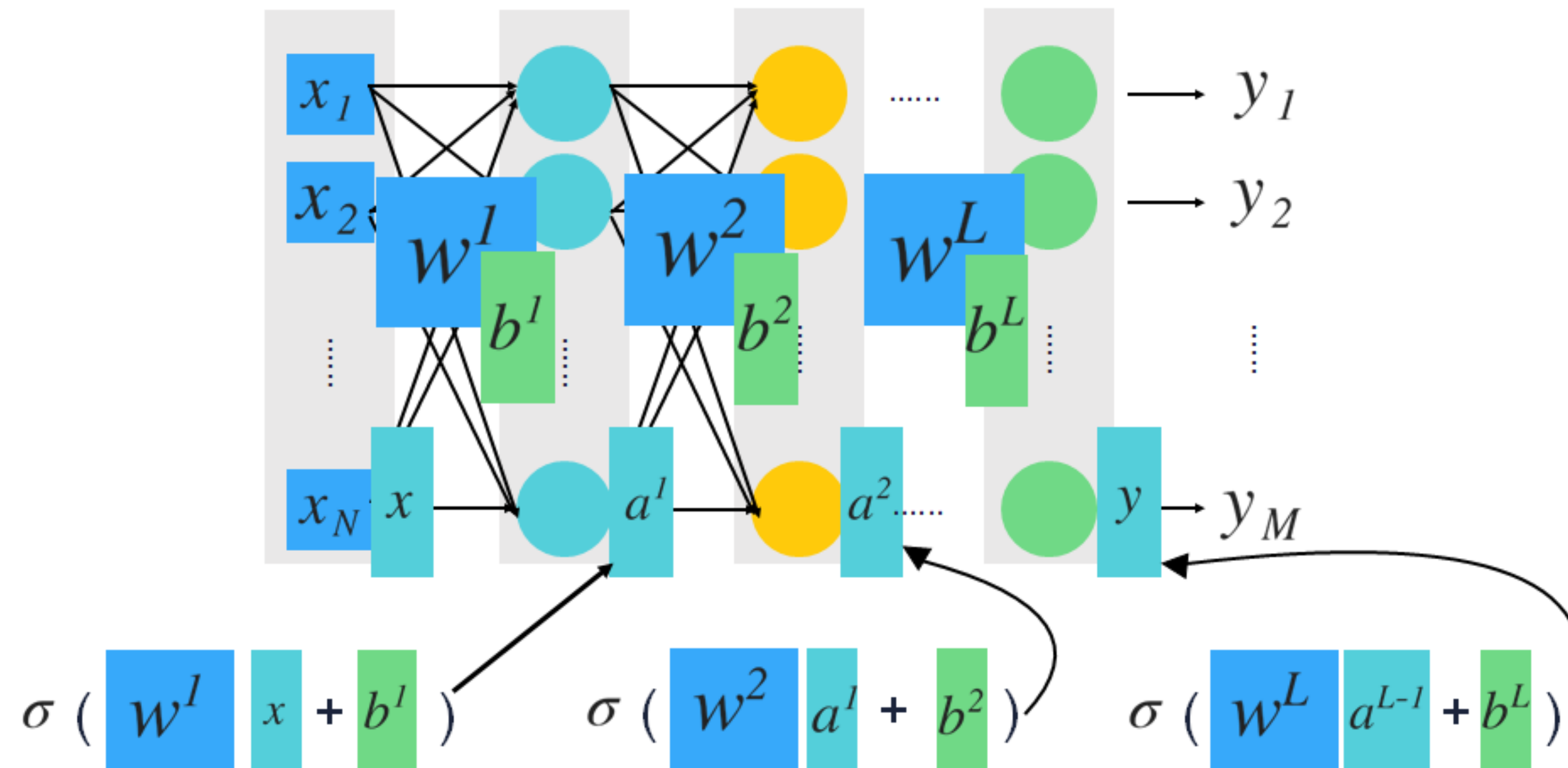
# 神經元傳輸過程(1)

- $[x_1, x_2, \dots, x_n]$  為輸入向量
- $[w_1, w_2, \dots, w_n]$  為權重矩陣
- 利用所有權重( $w$ )、輸入參數( $x$ )與偏移項( $b$ )的計算過程用矩陣完整呈現神經元傳輸過程



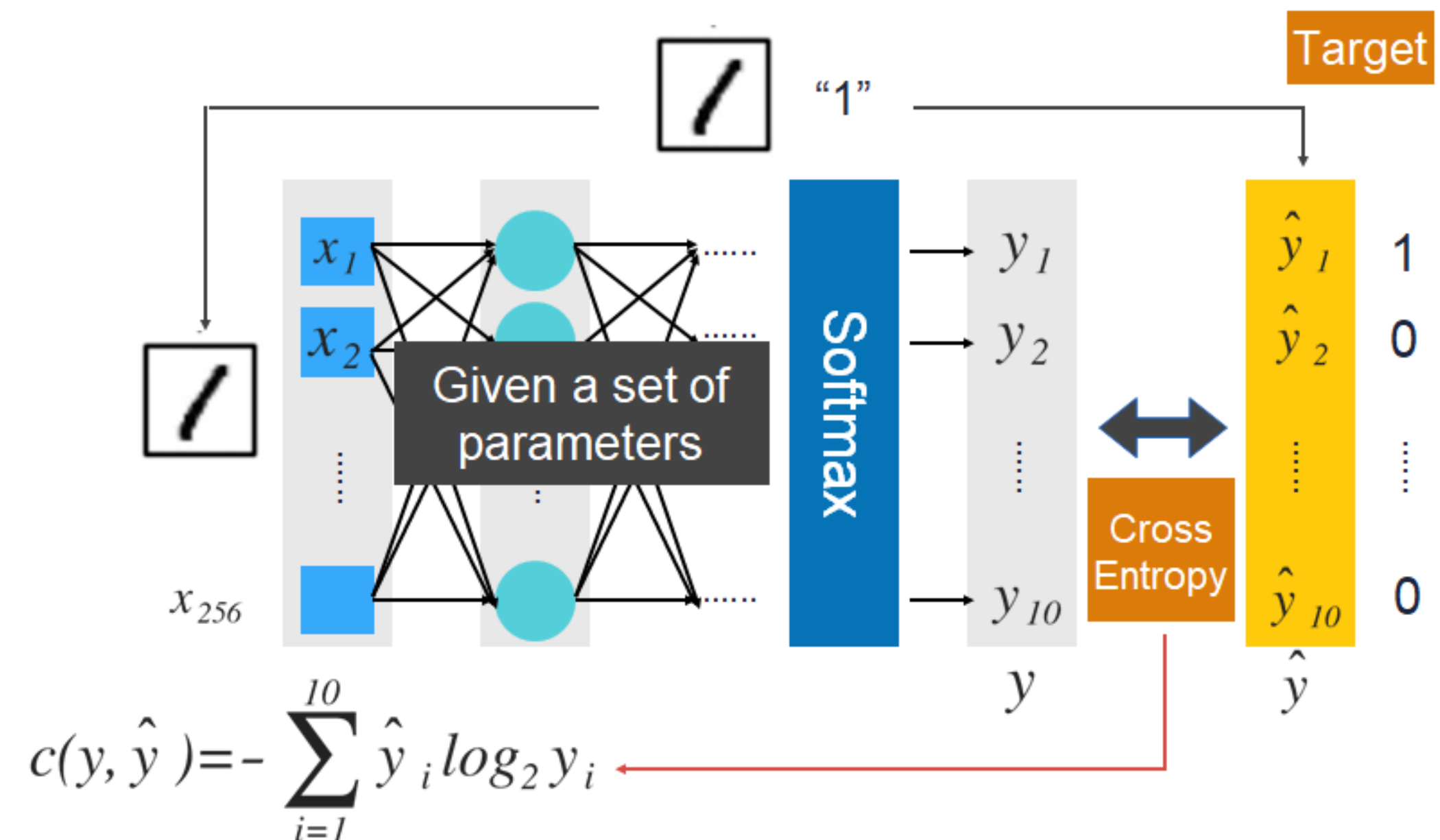
# 神經元傳輸過程(2)

- 每層的輸出結果函數，會成為下一層的輸入值
- $[w_1, w_2, \dots, w_n]$  為權重矩陣
- 把每層的輸出向量函數展開，就會看到每一層的神經元輸出的內容  
( $a_1, a_2 \dots$ )



# 深度學習網路步驟2-Evaluate the Functions

- 使用Loss Function作為評估整體神經網路的效能
- Cross-Entropy又稱K-L交叉熵，目的是用來評估模型預測結果與真實結果兩者之間的差距(損失數值)
- (黃色這條Y-hat是實際的結果，就是某個分類是1，其它是0，而模型預測的Y是一組機率值(加總為1))





# 深度學習網路步驟3-Pick the best Function

---

- 使用最佳化方法Gradient Descent(梯度下降法)修正神經網路中的權重( $W$ )，降低在訓練過程中產生的誤差

## Gradient Descent步驟:

1. Loss Function對網路中的各個 $W$ 進行偏微分，**讓各個 $W$ 權重逐步逼近最適合的權重**
2. 針對這 $y_1 \sim y_M$ 函數中(模型預測的結果)，各個權重進行偏微分，算出 $W$ 應該要修正的數值，並重複針對輸入下一批資料進行權重修正
3. 用Loss function針對各個 $W$ 權重作偏微分，調整各個權重( $W$ )的值，反覆進行，最後目標是 $Y_1 \sim Y_{10}$  的預測機率能逼近**損失數值(Loss)**最小的狀態，此時整個模型以及裡面的權重組合為Best Function

# 深度神經網路優缺點

---

## 優點

- 發展為多種架構應用於各領域中，如CNN、RNN、GAN
- 進行較深層的運算取出更多有用的資訊提升準確度

## 缺點

- 隱藏層增加更容易使得神經網路陷入區域最佳解，無法收斂
- 通過越多隱藏層產生的梯度變化就會越來越小，直到看不太出來變化，導致產生梯度消失問題
- 較深層的神經網路可能使模型陷入過擬合情形，需適當的使用正規化或是dropout做處理

# 解題時間 It's Your Turn

請跳出PDF至官網Sample Code & 作業  
開始解題

