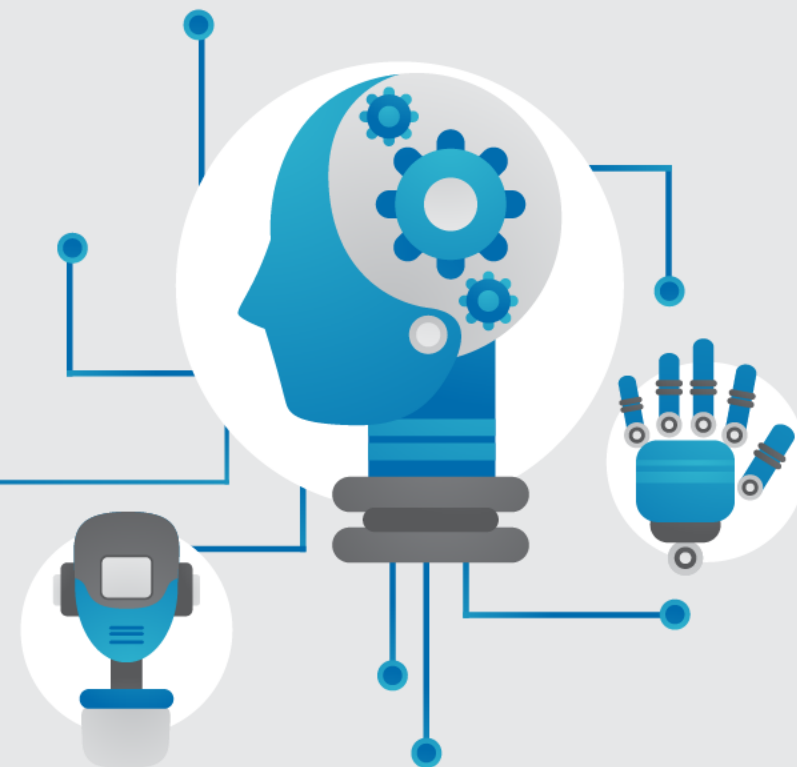


多層感知器網路概念(I)





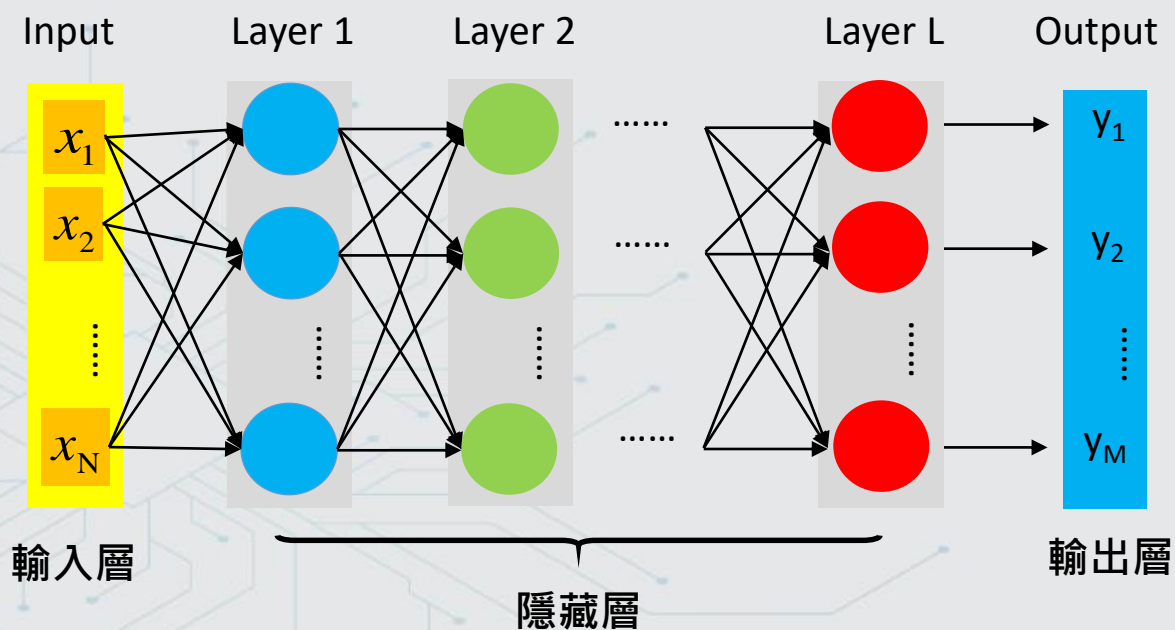
多層感知器模型 (Multi-Layer Perceptron)

機器學習實務



多層感知器

是屬於前饋網路 (feed forward) 的一種，網路每一層中可包含許多各自獨立的神經元，位於同一層的神經元彼此間無任何連結，每一層都全連接到下一層。

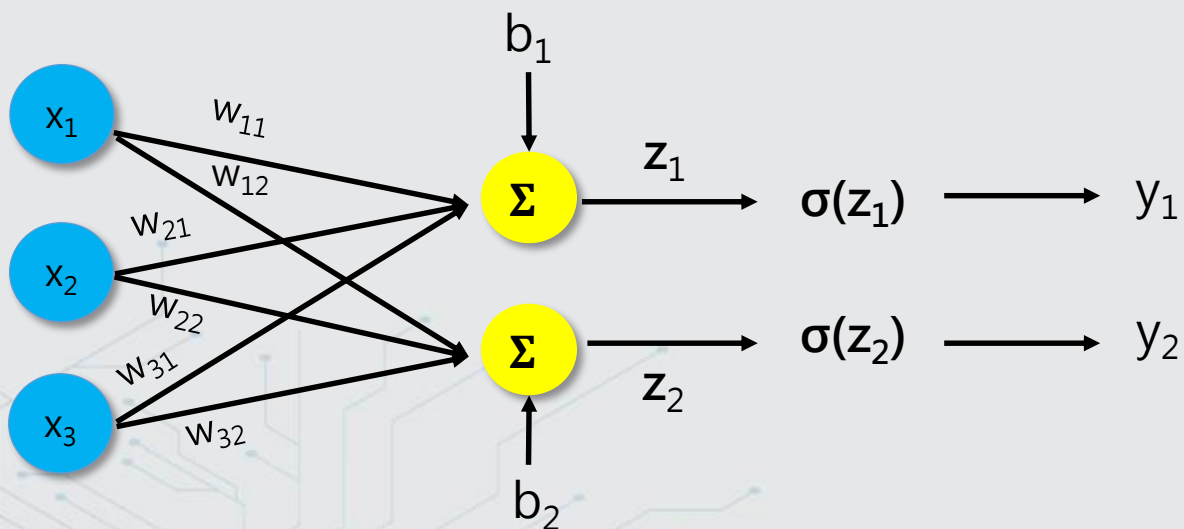




多層感知器運算



用矩陣運算模擬多層感知器運算



$$X = [x_1, x_2, x_3]$$

$$Y = [y_1, y_2]$$

$$B = [b_1, b_2]$$

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix}$$

$$Y = \sigma(X * W + B)$$



激活函數 (Activation function)



⚙️ 使用激活函數的目的

激活函數一般是**非線性函數**，作用是能夠給類神經網路加入非線性因素，使類神經網路解決更為複雜的問題。

⚙️ 常用的激活函數

- Sigmoid
- tanh
- Rectified Linear Unit
- Softmax

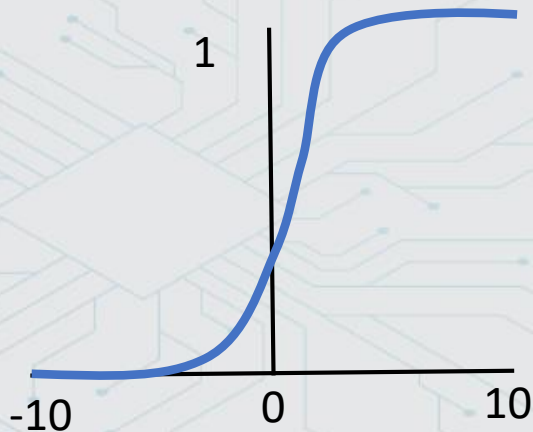


常用激活函數 - Sigmoid



Sigmoid

- 函數的輸出介於[0,1]，適用於輸出機率的模型。
- 在使用反向傳播 (Backpropagation) 調整權重時，會有梯度消失的問題 (Vanishing gradient problem)，無法有效地學習。
- 特別不適用於層數多的網路模型。



$$S(t) = \frac{1}{1 + e^{-t}}$$

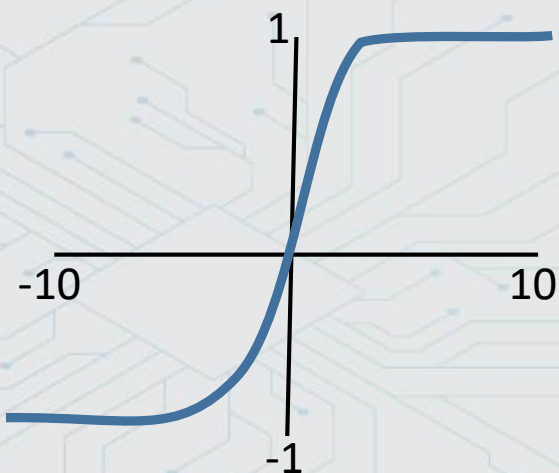


常用激活函數 - tanh



⚙️ 雙曲正切函數tanh

- 取值範圍是 $(-1, 1)$ ，曲線也呈「S」形。
- 主要用於分類。
- 會有梯度消失的問題。



$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

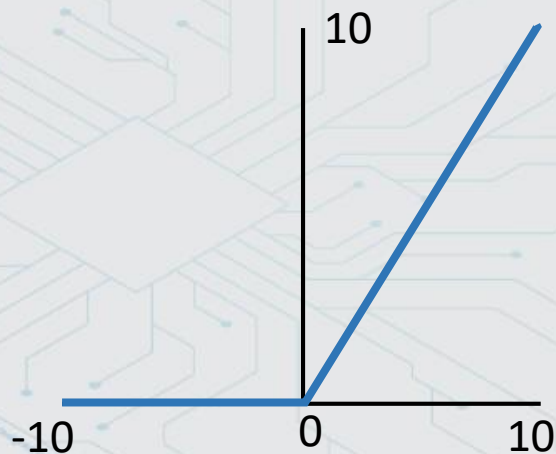


常用激活函數 - ReLU



⚙️ 線性整流函數 ReLU (Rectified Linear Unit)

- 在神經生理方面，當刺激超過某強度，才會引起神經衝動。ReLU捕捉了此生物神經元的特徵。
- 其分段線性性質能有效的克服梯度消失的問題。
- Dead ReLU problem：當輸入 x 都小於 0，就不再更新權重。



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



常用激活函數



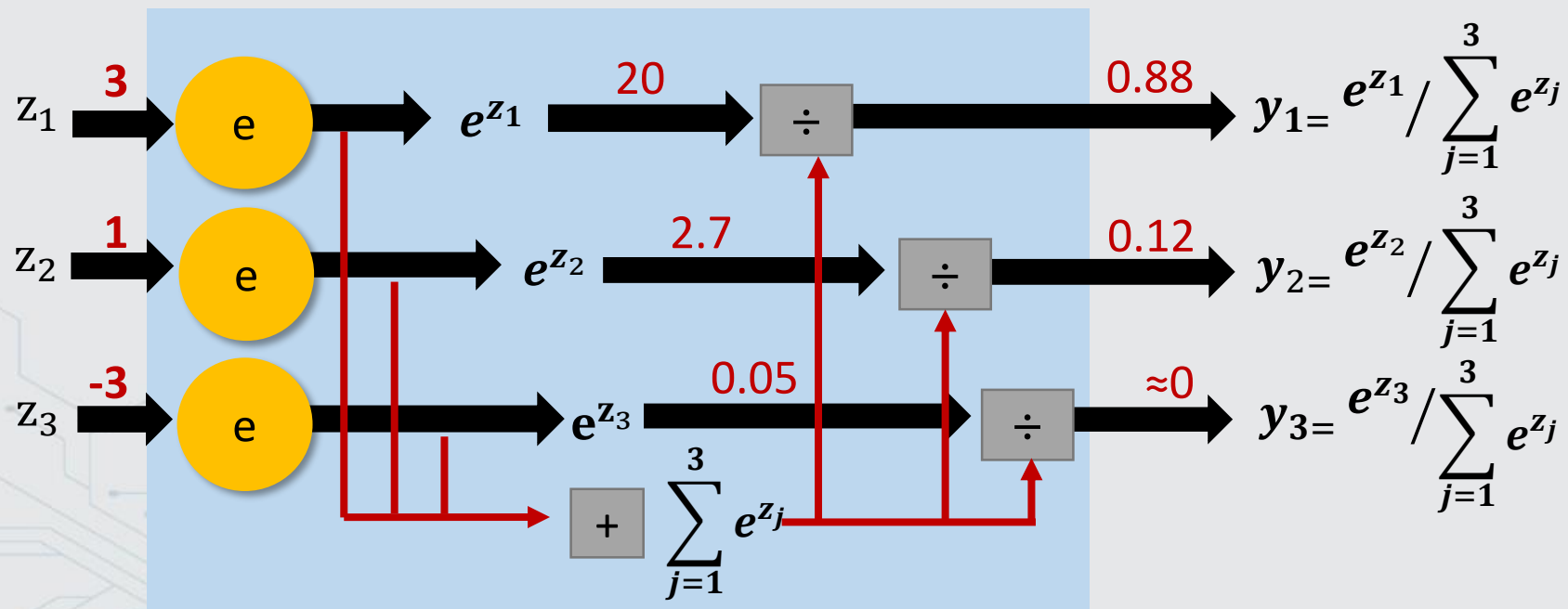
⚙️ Softmax (歸一化指數函數)

- 能將一個含任意實數的K維向量 z 「壓縮」到另一個K維實向量 z 中，使得每一個元素的範圍都在 $(0,1)$ 之間，並且所有元素的和為1。
- 可用於多分類的機率模型。

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{for } j=1,\dots,K$$



常用激活函數





損失函數



損失函數 (Loss function)

損失函數是用來計算預測值 (prediction)
與實際值 (actual values) 之間的偏差。偏差愈小，模型愈準確。



深度學習常用的 Loss function

- ✓ 均方誤差 (Mean_squared_error)
- ✓ 平均絕對誤差 (Mean_absolute_error)
- ✓ 平均絕對百分比誤差 (Mean_absolute_percentage_error)
- ✓ 均方對數誤差 (Mean_squared_logarithmic_error)
- ✓ 二分交叉熵 (Binary_crossentropy logloss)
- ✓ 分類交叉熵 (Categorical_crossentropy)



損失函數



Let Y be a vector of n predictions, and \hat{Y} be the vector of actual values.

- Mean squared error $MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$
- Mean absolute error $MAE = \frac{\sum_{i=1}^n |Y_i - \hat{Y}_i|}{n}$
- Mean absolute percentage error $MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{Y_i - \hat{Y}_i}{\hat{Y}_i} \right|$
- Mean squared logarithmic error

$$MSLE = \frac{100\%}{n} \sum_{i=1}^n (\log(Y_i) - \log(\hat{Y}_i))^2.$$



損失函數



$$LogLoss = -\frac{1}{N} \sum_{n=1}^N [y_n \log(\hat{y}_n) + (1 - y_n) \log(1 - \hat{y}_n)]$$



Binary cross-entropy loss

- ① 常用於**二分類**的問題，通常需要在網路的最後一層添加 sigmoid 配合使用。
- ② 期望輸出值 (target) 需要進行 one hot 編碼。



Categorical cross-entropy loss

- ① 適用於**多分類**的問題，並使用 softmax 作為輸出層的激活函數。
- ② 需要將類別的表示方法改成 one-hot encoding。



獨熱編碼 (Onehot Encoding)



獨熱編碼，又稱「一位有效編碼」

方法：使用N位狀態暫存器來對N個狀態進行編碼，每個狀態都有它獨立的暫存器位置，並且在任意時候，其中只有一位有效。

例如：

2進位編碼：

000,001,010,011,100,101

獨熱編碼：

000001,000010,000100,001000,010000,100000