解密AI黑盒子

主題二：感知器與邏輯迴歸

單元1

自然指數e
和
自然對數ln

自然指數e(exponentia)

- 定義下面數列的極限值為自然指數e(exponentia)：$e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n \approx 2.7183$
- 意義：為自然增長的極限值。
- 自然指數的導函數，設y=f(x)=e^x

$$\frac{dy}{dx} = \frac{de^x}{dx} = e^x$$

●動腦時間●

求出下列自然指數函數的導函數。

(1) $y = f(x) = e^{3x^2+1}$　　(2) $y = f(x) = \frac{1}{e^{-x}+1}$

自然對數ln(nature log)

- 一般以10為底對數函數是$log_{10}x$，簡寫為logx，用來求出x是10的多少次方。
  例：x=10^a⇨ logx=a

- 若是以自然指數為底的對數是$log_e x = ln\, x$，用來求出x是e的多少次方，這就是自然對數。
  例：x=e^a⇨$ln\, x$=a

- 自然對數的導函數，設y=f(x)=lnx $\quad \dfrac{\mathrm{dy}}{\mathrm{dx}} = \dfrac{\mathrm{d}ln\, x}{\mathrm{dx}} = \dfrac{1}{x}$

●動腦時間●

求出下列自然對數函數的導函數。

(1)$y = f(x) = \ln(x^2 + 3x + 1)$ (2) $y = f(x) = \ln \frac{1}{e^{-x}+1}$

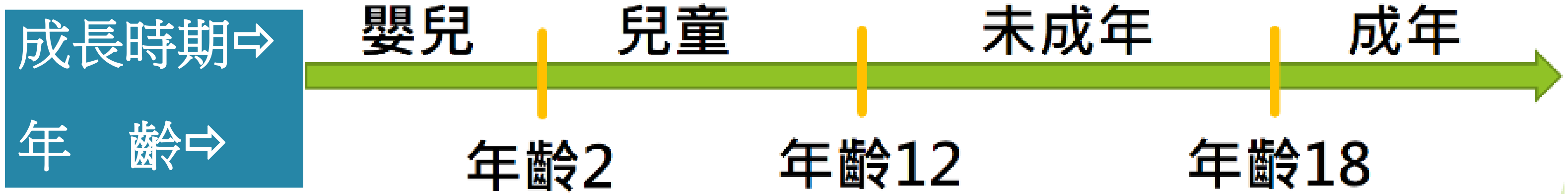
分類的概念
和
可視化分析

單元2

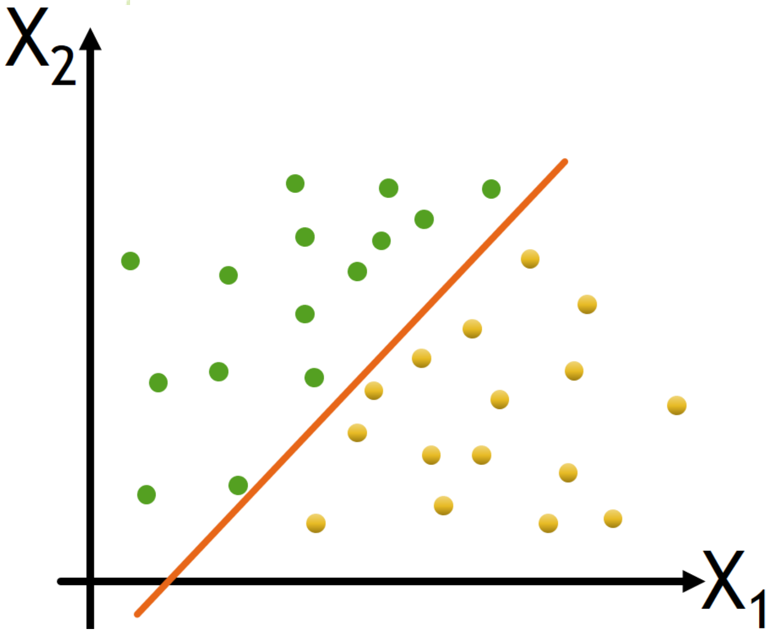
分類（Classification）

- 給定一個樣本特徵 x，我們希望預測其對應的屬性值 y，如果 y 是離散的，那麼這就是一個分類問題，如果 y 是連續的實數，這就是一個迴歸問題。

例：依據年齡特徵，可以將人的成長時期分為不同階段(離散資料)。



例：依據花的花瓣長x_1和花瓣寬x_2兩種特徵，分為不同品種。



- 機器學習在分類的重點是如何根據樣本特徵及對應的屬性值，去找出最佳的分類決策邊界。

離散資料如何做編碼

■ 人工智慧的處理流程中，運算處理的資料都是數值，所以必須將資料特徵和分類標示改用數值資料來編碼，才能送進神經網路中做運算處理。

例：下面表格中的原始資料欄位『學級』、『性別』和『等級』，必須經過編碼後才能輸入機器學習的模型作訓練。

學級	性別	分數	等級
國小	男	60	丁
國中	女	75	丙
高中	女	94	甲
國小	男	83	乙

Label encoding和One hot encoding

編碼	Label encoding	One hot encoding
方法	把每個類別對應到某個整數，不會增加新欄位。	為每個類別新增一個欄位，用 0 或 1 表示是否為這個分類
適用	原始資料是有序離散值	原始資料是無序離散值

舉例

Label encoding:

學級	性別	分數	等級
1	1	60	4
2	0	75	3
3	0	94	1
1	1	83	2

One hot encoding:

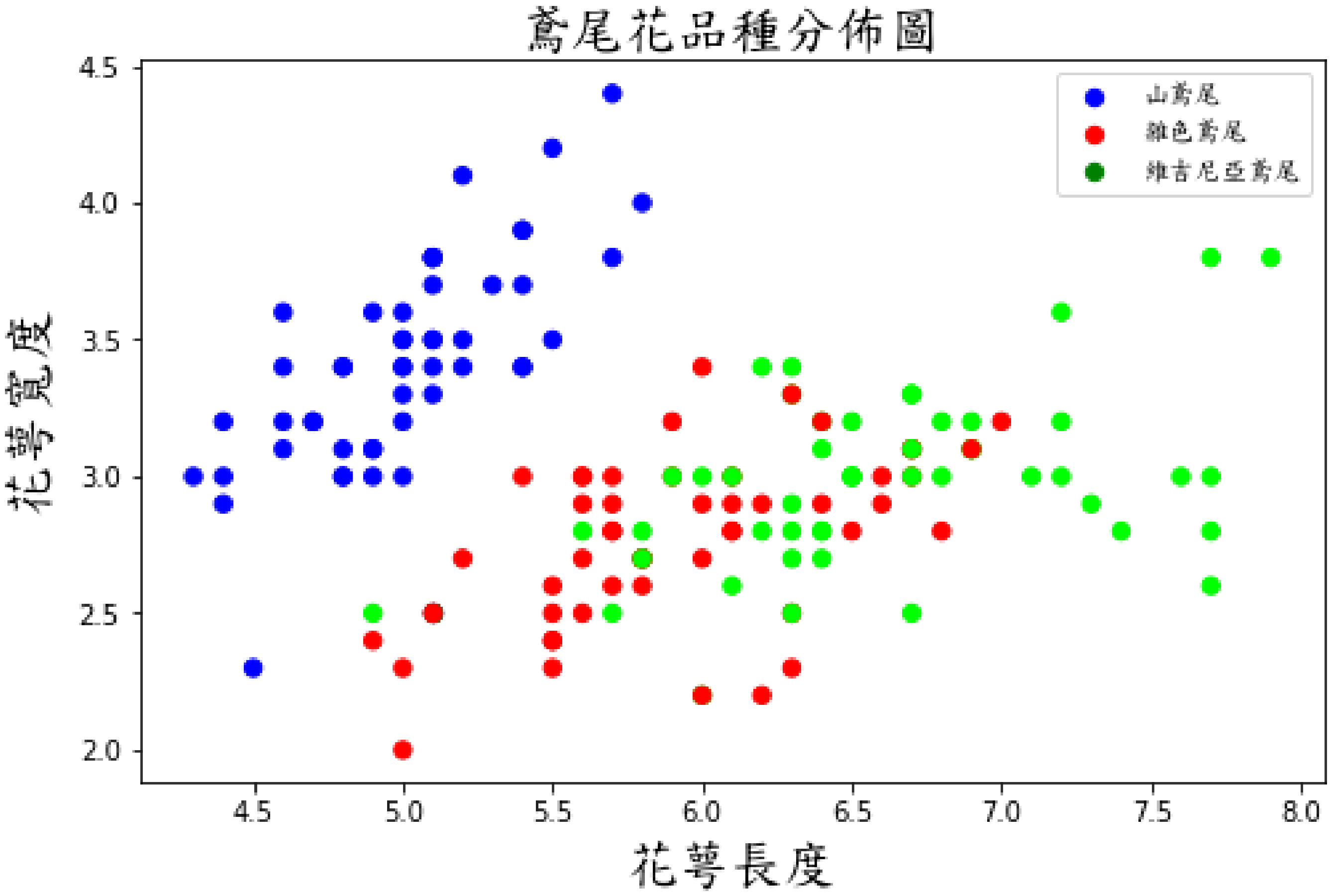
國小	國中	高中	男	女	分數	甲	乙	丙	丁
1	0	0	1	0	60	0	0	0	1
0	1	0	0	1	75	0	0	1	0
0	0	1	0	1	94	1	0	0	0
1	0	0	1	0	83	0	1	0	0

實作2-1 觀察鳶尾花的特徵分佈圖

- Iris dataset 由英國統計學家 Ronald Fisher 爵士在1936年時，對加斯帕半島上的鳶尾屬花朵調查所得150資料：
- 在每個資料包含四個屬性:花萼長度(Sepal length)，花萼寬度(Sepal width)，花瓣長度(Petal length)，花瓣寬度(Petal width)。
- 可通過這四個屬性預測鳶尾花卉屬於山鳶尾(setosa)、雜色鳶尾(versicolor)、維吉尼亞鳶尾(virginica)三類的哪一類示。

實作2-1 觀察鳶尾花的特徵分佈圖

- 取花萼長度(Sepal length)，花萼寬度(Sepal width)兩種特徵為坐標軸，畫出鳶尾花分為山鳶尾(setosa)、雜色鳶尾(versicolor)兩類的分佈圖(以不同顏色表示)，如右圖。
- 交錯選取不同的兩種特徵，觀察那些特徵比較容易能用來作分類？

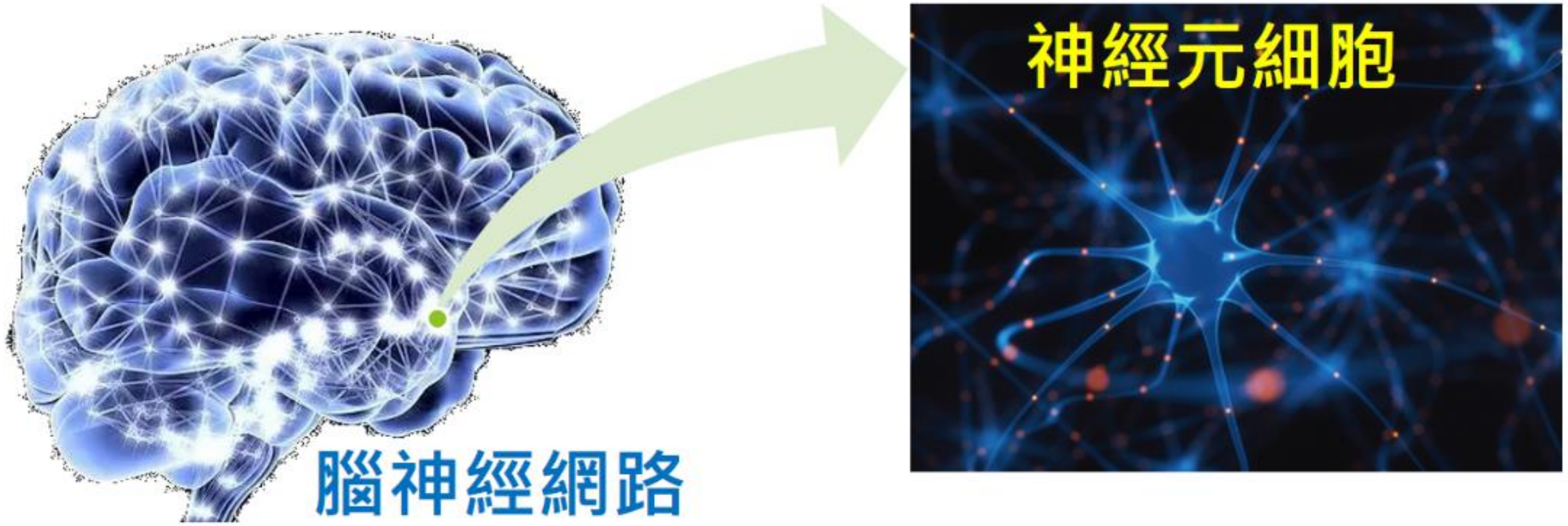


單元3

感知器 Perceptrons

人腦智慧-強大的神經網路

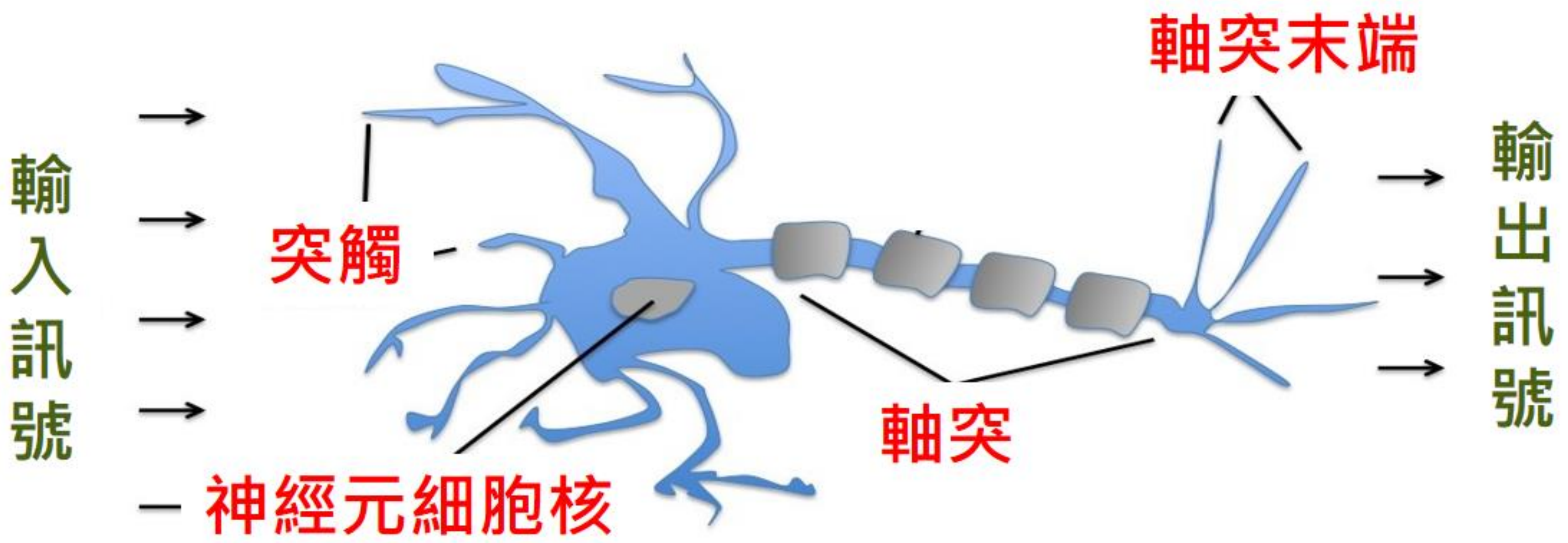
- 人腦智慧：能計算、記憶、邏輯推理、判斷，做出決策，更重要的是能靠學習、經驗累積創造智慧。
- 人腦有強大的腦神經網路，約一千億個神經元互相連結以傳遞訊息。
- 要把大腦神經元互相的連結存起來，最少大約要10^{22}Byte的記憶體，是目前全世界最大的超級電腦一千萬倍。



神經元(Neuron)

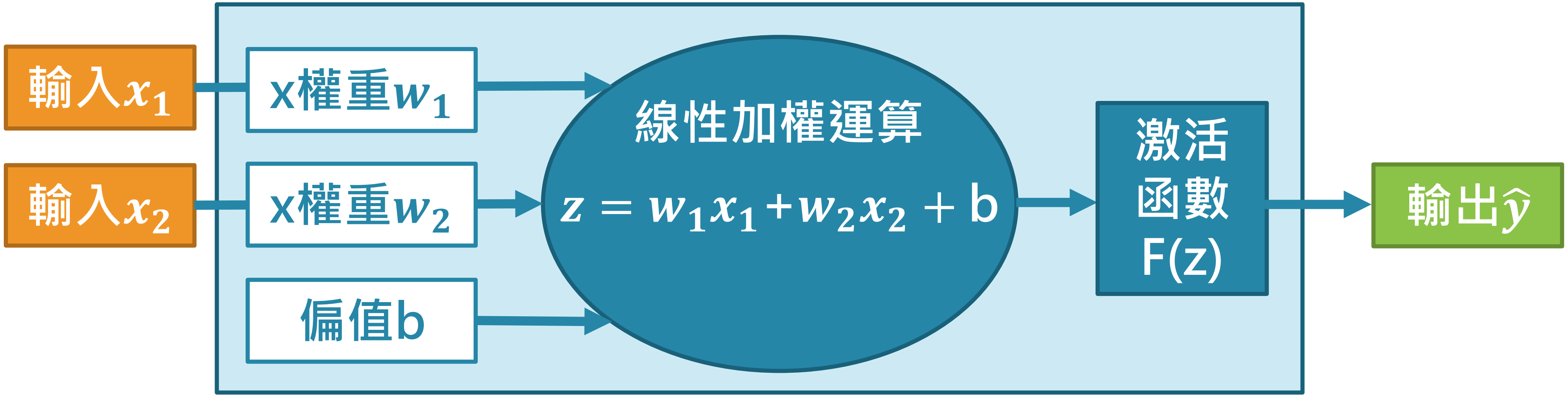
■ 組成大腦最基本的單位是神經細胞或「神經元」(neuron)

- 突觸負責接收訊號。
- 細胞核負責訊號的處理。
- 軸突末端與下一層神經元的突觸相連，軸突負責把訊號輸出。

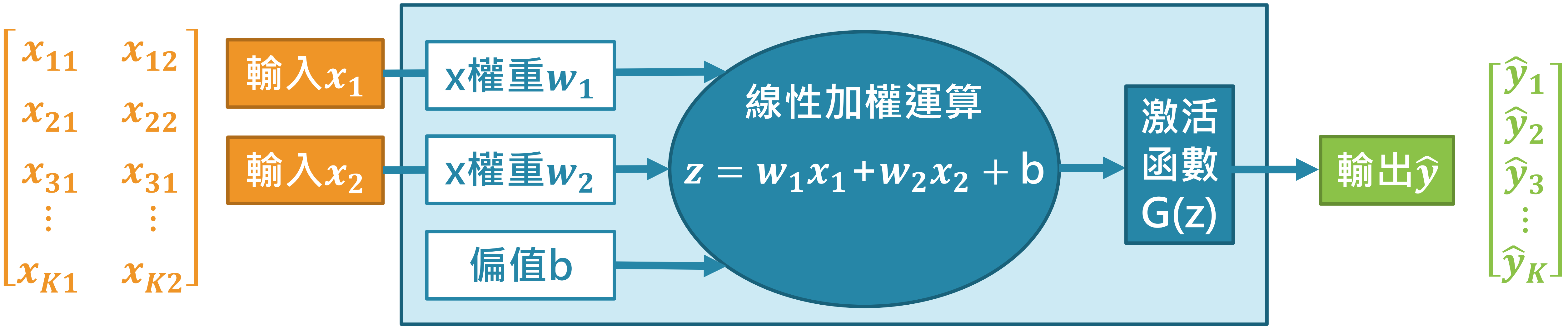


感知器（Perceptrons）

- 感知器：1950年美國神經科學家Frank Rosenblatt提出模擬神經元細胞，具有學習能力的最簡單處理單元。
  - 感知器將兩個輸入端的值乘以權重並與偏值加總：$z = w_1x_1 + w_2x_2 + \mathrm{b}$
  - 若輸出為離散資料，或非線性關係，則須再經過激活函數的處理。



感知器運算處理流程



$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \\ \vdots & \vdots \\ x_{K1} & x_{K2} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + b = \begin{bmatrix} w_1x_{11} + w_2x_{12} + b \\ w_1x_{21} + w_2x_{22} + b \\ w_1x_{31} + w_2x_{32} + b \\ \vdots \\ w_1x_{K1} + w_2x_{K2} + b \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_K \end{bmatrix}$$

$$\mathbf{F}\left(\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_K \end{bmatrix}\right) = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_K \end{bmatrix}$$

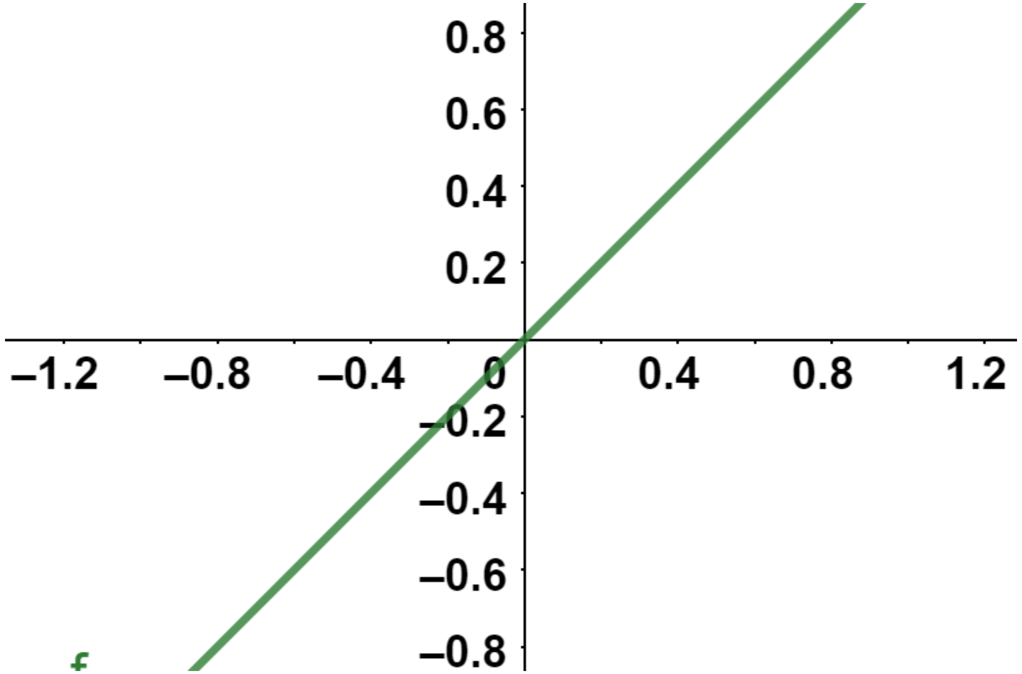
■ [預估值矩陣$\hat{Y}$] = G([**特徵矩陣**X] [權重矩陣$\boldsymbol{W}$] + [偏值$\boldsymbol{B}$])

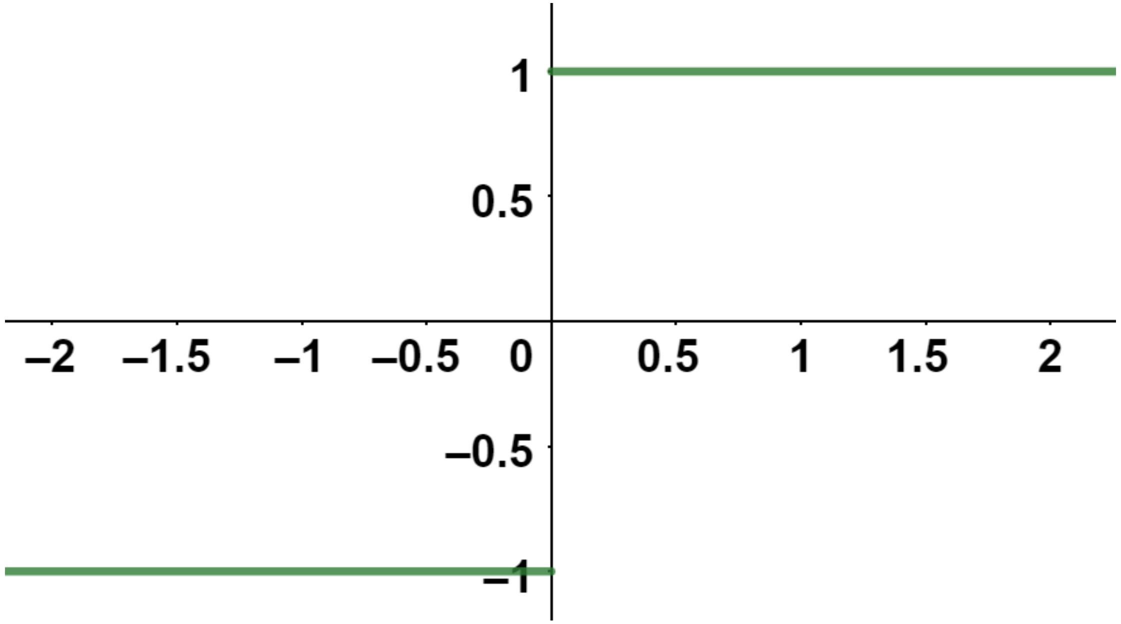
激活函數（Activation Function）

- 感知器可以依據用途選用特定的激活函數：
  - 若用在前一單元的迴歸，則可選用恆等函數。
  - 1950年Frank Rosenblatt用在二元分類上，則可選用階梯函數。

- 恆等函數： $G(z) = z$

- Step Function $\begin{cases} 當 z \geq 0 \Rightarrow G(z) = +1 \\ 當 z < 0 \Rightarrow G(z) = -1 \end{cases}$





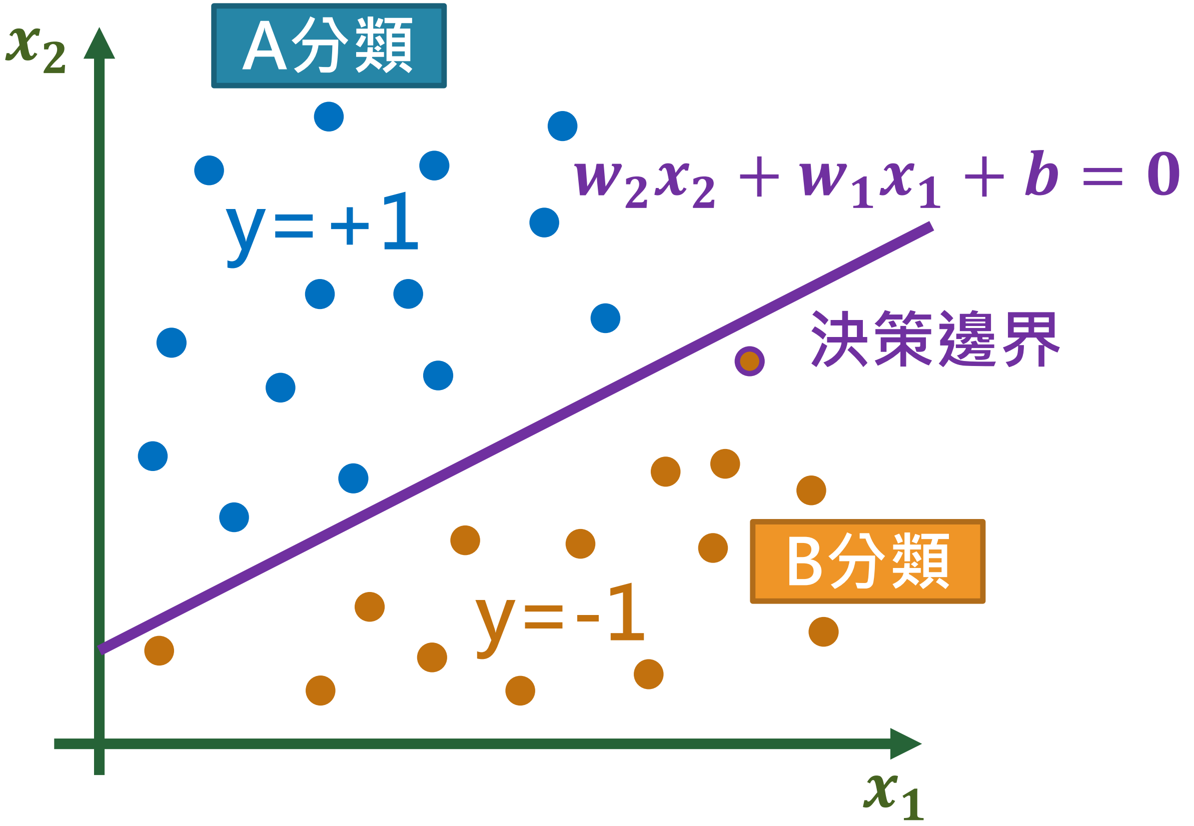
利用感知器處理二元分類

■ 線性二元分類的決策邊界是一條線性方程式：

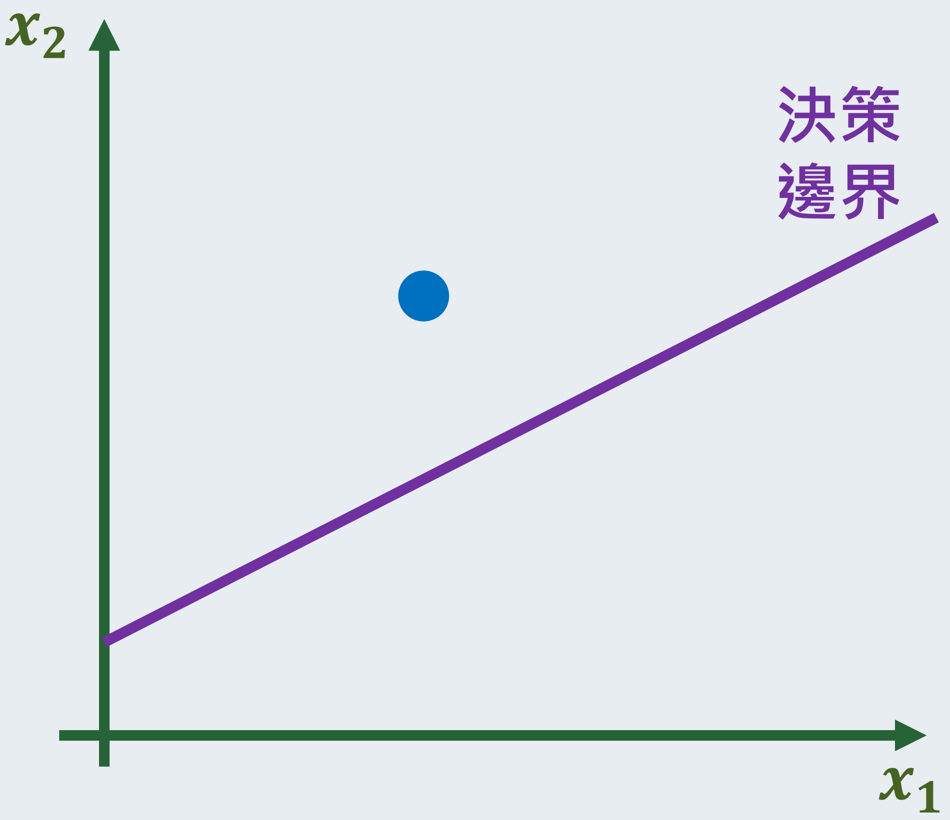
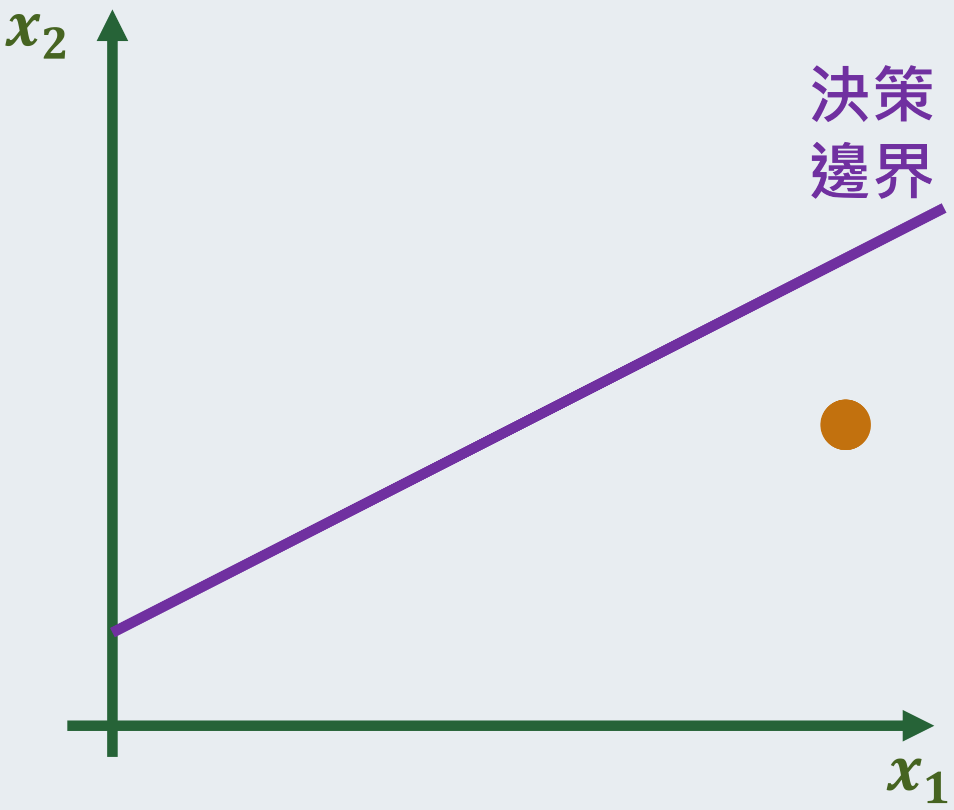
$$w_2x_2 + w_1x_1 + b = 0$$

- 若$w_2x_2 + w_1x_1 + b \geqq 0$：
  則分類屬於A

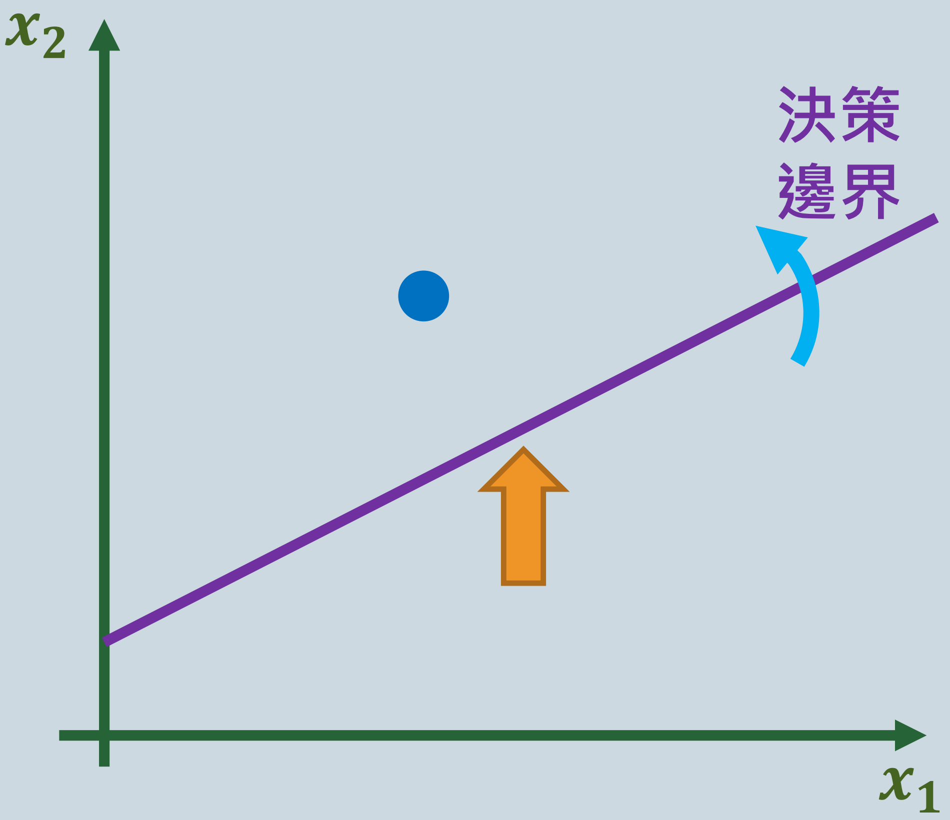
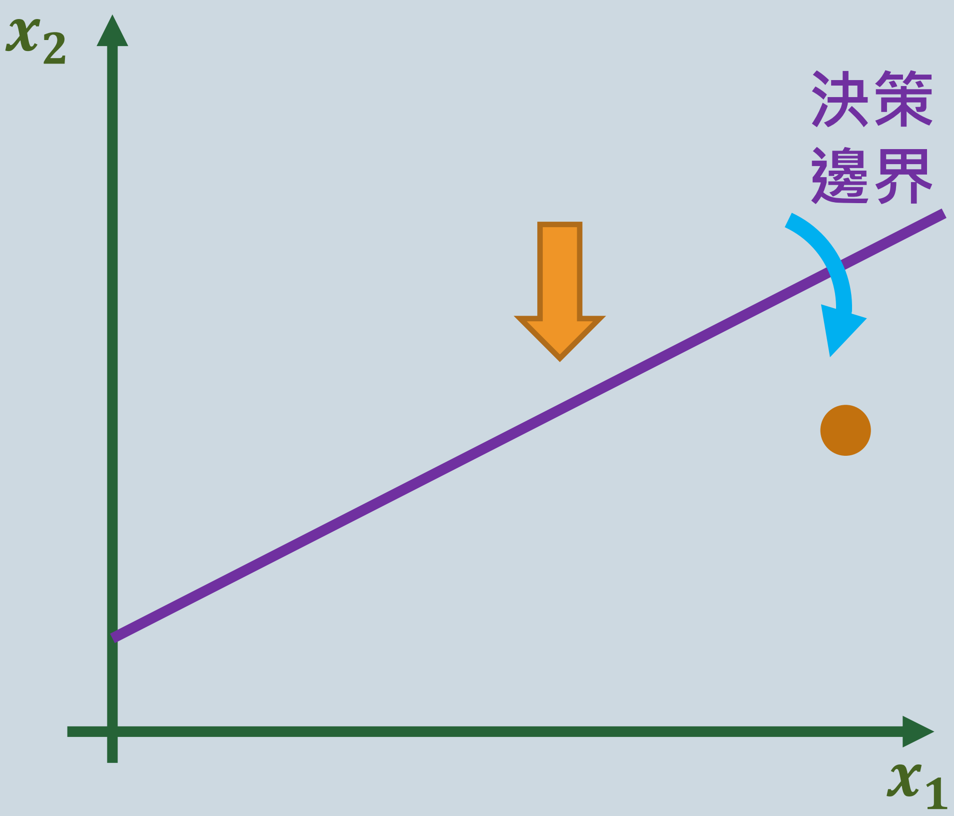
- 若$w_2x_2 + w_1x_1 + b < 0$：
  則分類屬於B



感知器如何機器學習？

情況	實際分類為B類(標示y=−1) 決策判斷為A類(標示$\hat{y}$=+1)	實際分類為A類(標示y=+1) 決策判斷為B類(標示$\hat{y}$=−1)
誤差	實際值y - 判斷值$\hat{y}$ = - 2	實際值y - 判斷值$\hat{y}$ = + 2
圖示		

感知器如何機器學習？

調整方式	(1)假設學習率為η (2)將決策邊界向上移：調降b (3)將決策邊界逆時針旋轉	(1)假設學習率為η (2)將決策邊界向下移：調升b (3)將決策邊界順時針旋轉
圖示		

感知器的問題

■ 只有模型預測分類錯誤的時候，才會產生修正。

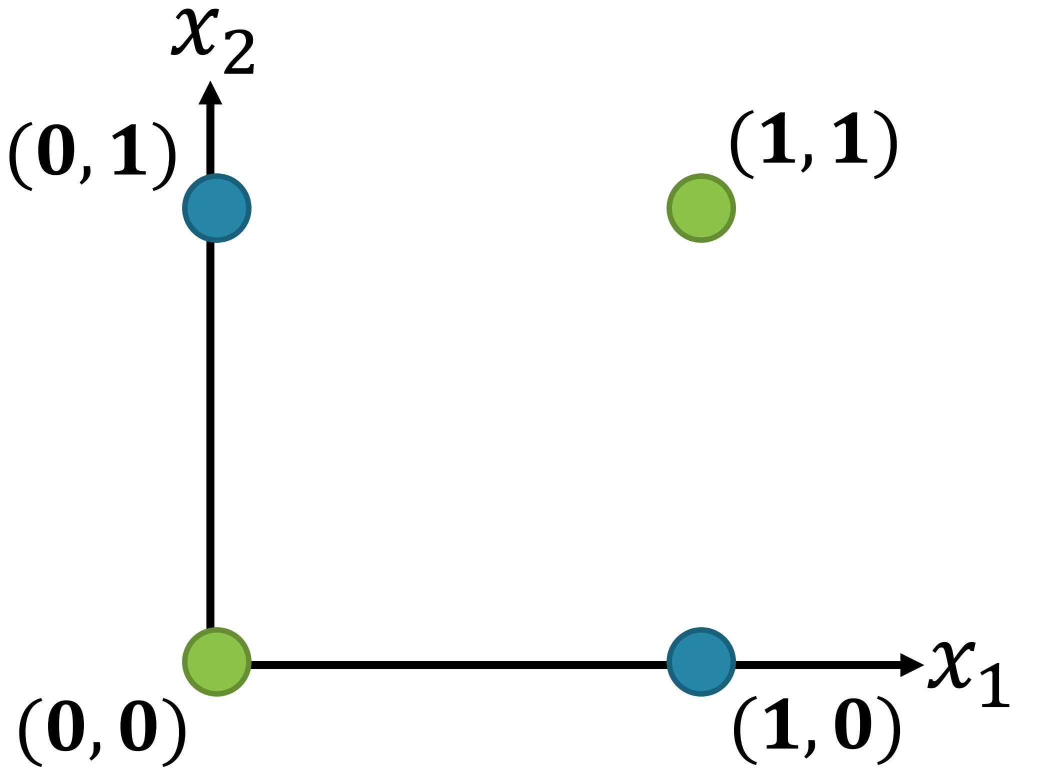
■ 沒有評估模型好壞的損失函數，無法讓模型取得最佳狀態。

■ 學習率的大小和資料的順序會影響到模型的訓練結果。

■ 只能處理線性分類，無法處理非線性分類，對於簡單的XOR問題也無法處理。

■ 感知器是非常單純的模型，只能處理較為簡單的分類問題，但卻是神經網路的源頭。

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

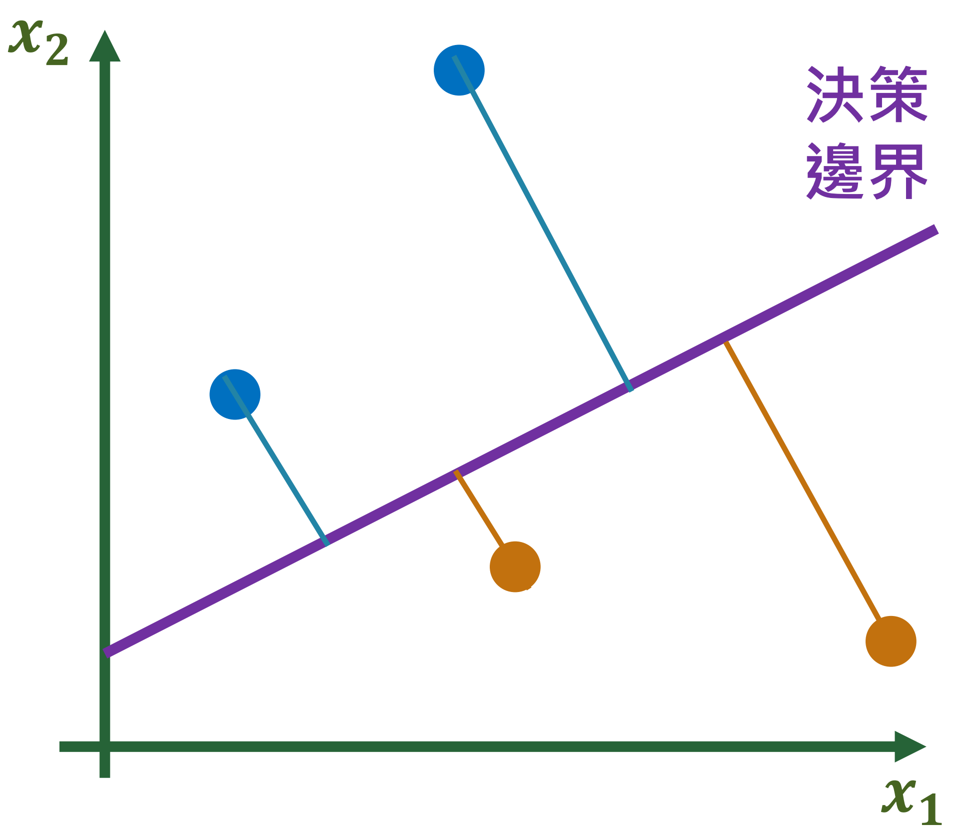


單元4

邏輯迴歸
Logistic regression

分類好壞如何評估？

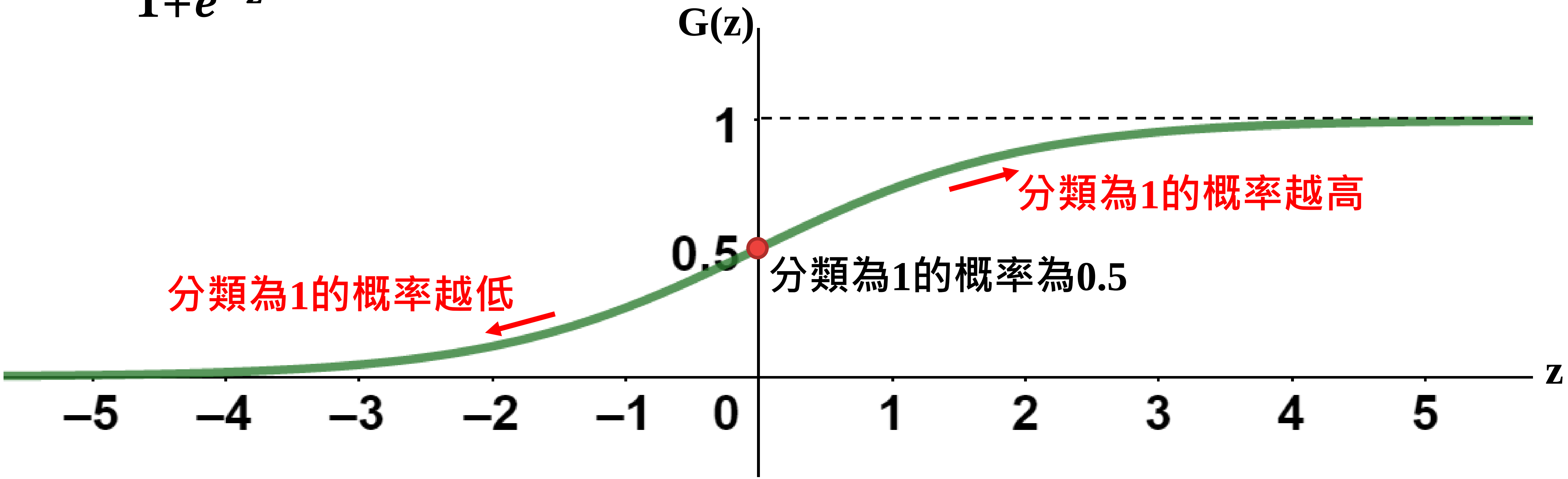
- 剛好在決策邊界上特徵資料，分類為A分類和B分類的機率均為0.5。
- 在決策邊界上方(z>0)的特徵資料，離決策邊界越遠的，分類為A的機率越高，分類效果越好。
- 在決策邊界下方(z<0)的特徵資料，離決策邊界越遠的，分類為B的機率越高，分類效果越好。



感知器的改進

- 感知器使用的激活函數改為Sigmoid Function：

$$G(z)=\frac{1}{1+e^{-z}}=\text{輸出分類為1的機率p}$$

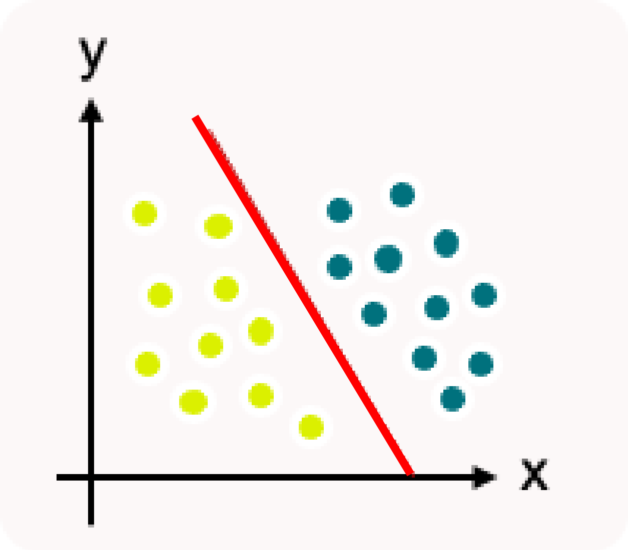


- 連續可微分： 可梯度下降修正權重參數。
- 有分類概率：產生損失函數(Loss Function)，用來評估模型好壞。

邏輯迴歸Logistic regression

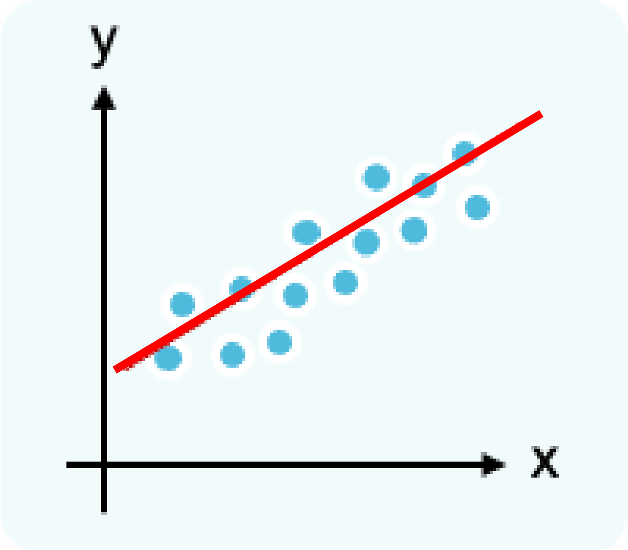
- 邏輯迴歸 (Logistic regression) 是由線性迴歸變化而來的，它是一種分類的模型。
- 邏輯迴歸的目標是要找出一條直線能夠將所有數據清楚地分開並做分類，我們又可以稱迴歸的線性分類器。

Logistic Regression



找到能將資料分開的最佳決策邊界直線

Linear Regression



找到跟資料最佳擬合的直線

邏輯迴歸損失函數

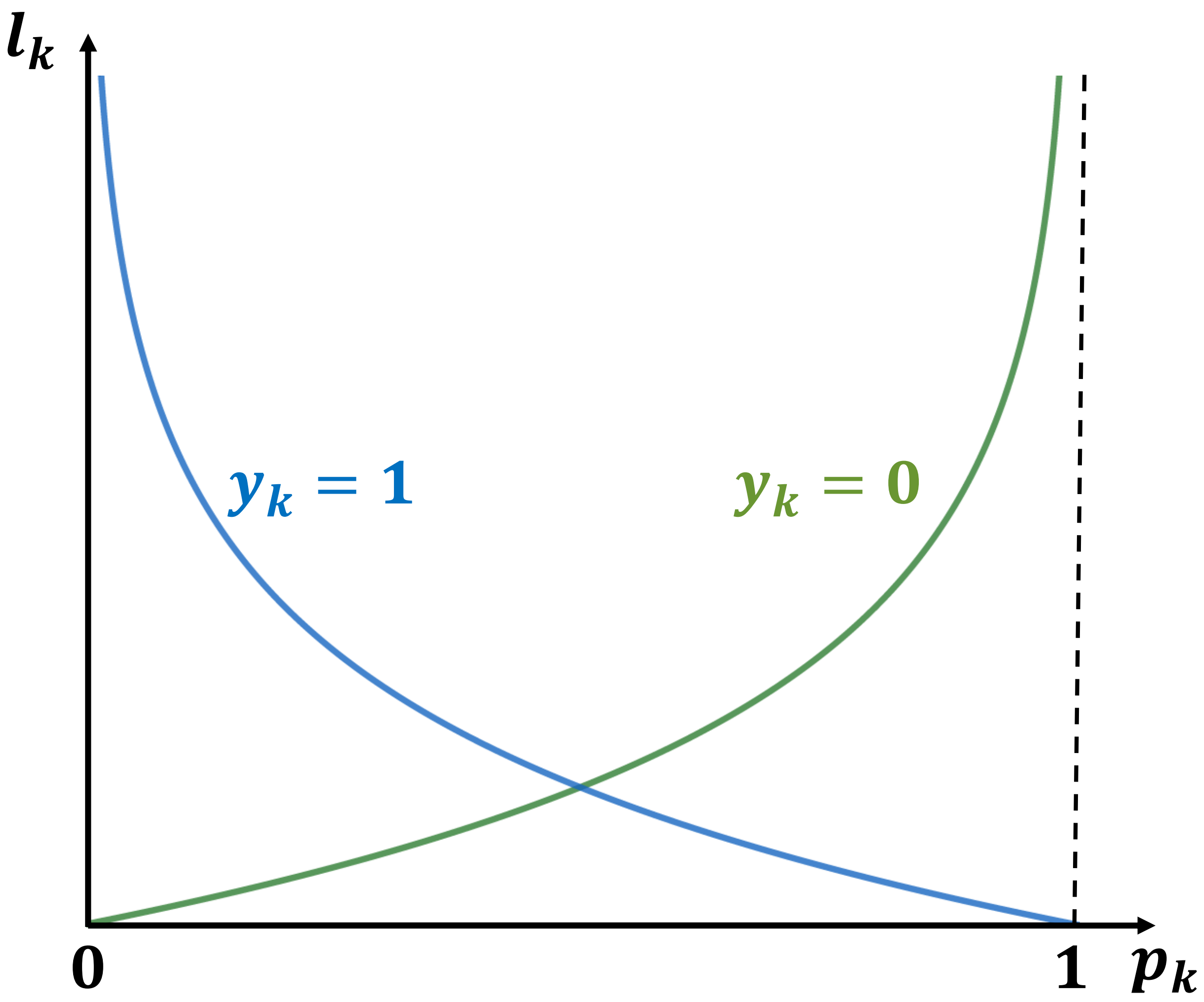
- 第k筆樣本分類為1的機率$p_k=\dfrac{1}{1+e^{-z_k}}$

- 預測損失值l_k=

$$-[y_k \times ln(p_k) + (1-y_k) \times ln(1-p_k)]$$

- 當實際值y_k=1：(只看上式的第1項)
  - 若預測值p_k=1，l_k=0
  - 若預測值p_k=0，l_k=∞

- 當實際值y_k=0：(只看上式的第2項)
  - 若預測值p_k=1，l_k=∞
  - 若預測值p_k=0，l_k=0



邏輯迴歸損失函數

■ 將所有資料的預測損失值加總(稱為交叉熵cross-entropy)，可以用來作為損失函數，判斷分類模型的好壞。

$$\text{損失函數L}=\sum_{k=1}^{K} l_k = -\sum_{k=1}^{K}\left[y_k \times ln(p_k) + (1-y_k)\times ln(1-p_k)\right]$$

$p_k=\frac{1}{1+e^{-z_k}}$ 代入

$$=\sum_{k=1}^{K}\left[ln(1+e^{z_k}) - y_k \times z_k\right]$$

梯度下降法更新迴歸係數

$b,\ w_1,\ w_2 \rightarrow z_k \rightarrow L$

$$z_k = b + w_1 x_{k1} + w_2 x_{k2}$$

$$L = \sum_{k=1}^{K} [\ln(1 + e^{z_k}) - y_k \times z_k]$$

$$\frac{\partial z_k}{\partial b} = 1 \text{、} \frac{\partial z_k}{\partial w_1} = x_{k1} \text{、} \frac{\partial z_k}{\partial w_2} = x_{k2}$$

$$\frac{\partial L}{\partial z_k} = \sum_{k=1}^{K} \left(\frac{1}{1+e^{z_k}} e^{z_k} - y_k\right)$$

$$= \sum_{k=1}^{K} \left(\frac{1}{1+e^{-z_k}} - y_k\right)$$

$$= \sum_{k=1}^{K} (\hat{y}_k - y_k)$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial z_k}\frac{\partial z_k}{\partial b} = \sum_{k=1}^{K} (\hat{y}_k - y_k)$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial z_k}\frac{\partial z_k}{\partial w_1} = \sum_{k=1}^{K} (\hat{y}_k - y_k) x_{k1}$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial z_k}\frac{\partial z_k}{\partial w_2} = \sum_{k=1}^{K} (\hat{y}_k - y_k) x_{k2}$$

梯度下降法更新迴歸係數

■ w_0和w_1和w_2的修正方程式為：

$$b = b - \eta \frac{\partial L}{\partial b} = b - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k)$$

$$w_1 = w_1 - \eta \frac{\partial L}{\partial w_1} = w_1 - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k) x_{k1}$$

$$w_2 = w_2 - \eta \frac{\partial L}{\partial w_2} = w_2 - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k) x_{k2}$$

梯度下降法更新迴歸係數

■ 梯度下降更新迴歸係數以矩陣運算：

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} - \eta \begin{bmatrix} (\hat{y}_1 - y_1)x_{11} + (\hat{y}_2 - y_2)x_{21} + (\hat{y}_3 - y_3)x_{31} + 、、+ (\hat{y}_3 - y_3)x_{k1} \\ (\hat{y}_1 - y_1)x_{12} + (\hat{y}_2 - y_2)x_{22} + (\hat{y}_3 - y_3)x_{32} + 、、+ (\hat{y}_3 - y_3)x_{k2} \end{bmatrix}$$

$$= [w] - \eta \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \\ \vdots & \vdots \\ x_{K1} & x_{K2} \end{bmatrix}^T \times \left(\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_K \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_K \end{bmatrix} \right)$$

■ [矩陣W] = [矩陣W] − η[矩陣X]^T × ([矩陣Ŷ] −[矩陣Y])

實作2-2 利用邏輯迴歸做鳶尾花的二元分類

- 實作要求：
  - 取鳶尾花的花萼長和花萼寬兩個特徵，將山鳶尾和變色鳶尾做二元分類。
  - 紀錄並觀察學習過程損失函數的變化。
  - 畫圖呈現出分類決策邊界線的效果。
  - 以驗證集資料測試其準確度。

曲線的決策邊界

單元5

邏輯迴歸-曲線決策邊界

- 曲線決策邊界方程式：x_2為x_1的n次方函數
  $b+w_1x_1+w_2x_1^2+$、、、$+w_nx_1^n+w_{n+1}x_2=0$

- 偏值和權重有n+2個：b、w_1、w_2、w_3、、、w_{n+1}

- 第k筆資料的特徵值(x_{k1}，x_{k2})，加權運算後
  $z_k=b+w_1x_{k1}+w_2x_{k1}^2+$、、、$+w_nx_{k1}^n+w_{n+1}x_{k2}$

用矩陣運算模型的預估值

$$\hat{Y}=\begin{bmatrix}\hat{y}_1\\ \hat{y}_2\\ \hat{y}_3\\ \vdots\\ \hat{y}_K\end{bmatrix}=G\left(\begin{bmatrix} b+w_1x_{11}+w_2x_{11}^2+、、、+w_nx_{11}^n+w_{n+1}x_{12}\\ b+w_1x_{21}+w_2x_{21}^2+、、、+w_nx_{21}^n+w_{n+1}x_{22}\\ b+w_1x_{31}+w_2x_{31}^2+、、、+w_nx_{31}^n+w_{n+1}x_{32}\\ \vdots\\ b+w_1x_{K1}+w_2x_{K1}^2+、、、+w_nx_{K1}^n+w_{n+1}x_{K2}\end{bmatrix}\right)$$

$$\begin{array}{cc} & \text{特徵數量 n+1} \\ \text{樣本數K} & \end{array}$$

$$=G\left(\begin{bmatrix} x_{11}^1 & x_{11}^2 & x_{11}^3 & \cdots & x_{11}^n & x_{12}\\ x_{21}^1 & x_{21}^2 & x_{21}^3 & \cdots & x_{21}^n & x_{22}\\ x_{31}^1 & x_{31}^2 & x_{31}^3 & \cdots & x_{31}^n & x_{32}\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ x_{K1}^1 & x_{K1}^2 & x_{K1}^3 & \cdots & x_{K1}^n & x_{K2}\end{bmatrix}\begin{bmatrix} w_1\\ w_2\\ w_3\\ \vdots\\ w_n\\ w_{n+1}\end{bmatrix}+b\right)$$

[預估值矩陣$\hat{Y}$] =G([特徵資料矩陣X] [權重矩陣W]+偏值b)

梯度下降法更新迴歸係數

b

w_1

w_2

w_{n+1}

$$z_k = b + w_1 x_{k1} + w_2 x_{k1}^2 、、 + w_n x_{k1}^n + w_{n+1} x_{k2}$$

$$\text{L} = \sum_{k=1}^{K} [ln(1 + e^{z_k}) - y_k \times z_k]$$

z_k → L

$$\frac{\partial z_k}{\partial b} = 1 \text{ 、 } \frac{\partial z_k}{\partial w_1} = x_{k1} \text{ 、 } \frac{\partial z_k}{\partial w_2} = x_{k1}^2$$

$$\frac{\partial z_k}{\partial w_n} = x_{k1}^n \text{ 、 } \frac{\partial z_k}{\partial w_{n+1}} = x_{k2}$$

$$\frac{\partial L}{\partial z_k} = \sum_{k=1}^{K} (\frac{1}{1 + e^{z_k}} e^{z_k} - y_k)$$

$$= \sum_{k=1}^{K} (\frac{1}{1 + e^{-z_k}} - y_k)$$

$$= \sum_{k=1}^{K} (\hat{y}_k - y_k)$$

梯度下降法更新迴歸係數

■ b、w_1、w_2、、、和w_n、w_{n+1}的梯度下降修正方程式為：

$$b = b - \eta \frac{\partial L}{\partial b} = b - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k)$$

、

$$w_1 = w_1 - \eta \frac{\partial L}{\partial w_1} = w_1 - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k) x_{k1}$$

$$w_2 = w_2 - \eta \frac{\partial L}{\partial w_2} = w_2 - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k)\, x_{k1}^2$$

、、、

$$w_n = w_n - \eta \frac{\partial L}{\partial w_n} = w_n - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k)\, x_{k1}^n$$

$$w_{n+1} = w_{n+1} - \eta \frac{\partial L}{\partial w_{n+1}} = w_{n+1} - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k)\, x_{k2}$$

梯度下降法更新迴歸係數

■ 梯度下降更新迴歸係數以矩陣運算：

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \\ w_{n+1} \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \\ w_{n+1} \end{bmatrix} - \eta \begin{bmatrix} (\hat{y}_1 - y_1)x_{11} + (\hat{y}_2 - y_2)x_{21} + (\hat{y}_3 - y_3)x_{31} + 、、 + (\hat{y}_K - y_K)x_{K1} \\ (\hat{y}_1 - y_1)x_{11}^2 + (\hat{y}_2 - y_2)x_{21}^2 + (\hat{y}_3 - y_3)x_{31}^2 + 、、 + (\hat{y}_K - y_K)x_{K1}^2 \\ (\hat{y}_1 - y_1)x_{11}^3 + (\hat{y}_2 - y_2)x_{21}^3 + (\hat{y}_3 - y_3)x_{31}^3 + 、、 + (\hat{y}_K - y_K)x_{K1}^3 \\ \vdots \\ (\hat{y}_1 - y_1)x_{11}^n + (\hat{y}_2 - y_2)x_{21}^n + (\hat{y}_3 - y_3)x_{31}^n + 、、 + (\hat{y}_K - y_K)x_{K1}^n \\ (\hat{y}_1 - y_1)x_{12} + (\hat{y}_2 - y_2)x_{22} + (\hat{y}_3 - y_3)x_{32} + 、、 + (\hat{y}_K - y_K)x_{K2} \end{bmatrix}$$

■ $= \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \\ w_{n+1} \end{bmatrix} - \eta \begin{bmatrix} x_{11}^1 & x_{11}^2 & x_{11}^3 & \cdots & x_{11}^n & x_{12} \\ x_{21}^1 & x_{21}^2 & x_{21}^3 & \cdots & x_{21}^n & x_{22} \\ x_{31}^1 & x_{31}^2 & x_{31}^3 & \cdots & x_{31}^n & x_{32} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{K1}^1 & x_{K1}^2 & x_{K1}^3 & \cdots & x_{K1}^n & x_{K2} \end{bmatrix}^T \times \left(\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_K \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_K \end{bmatrix} \right)$

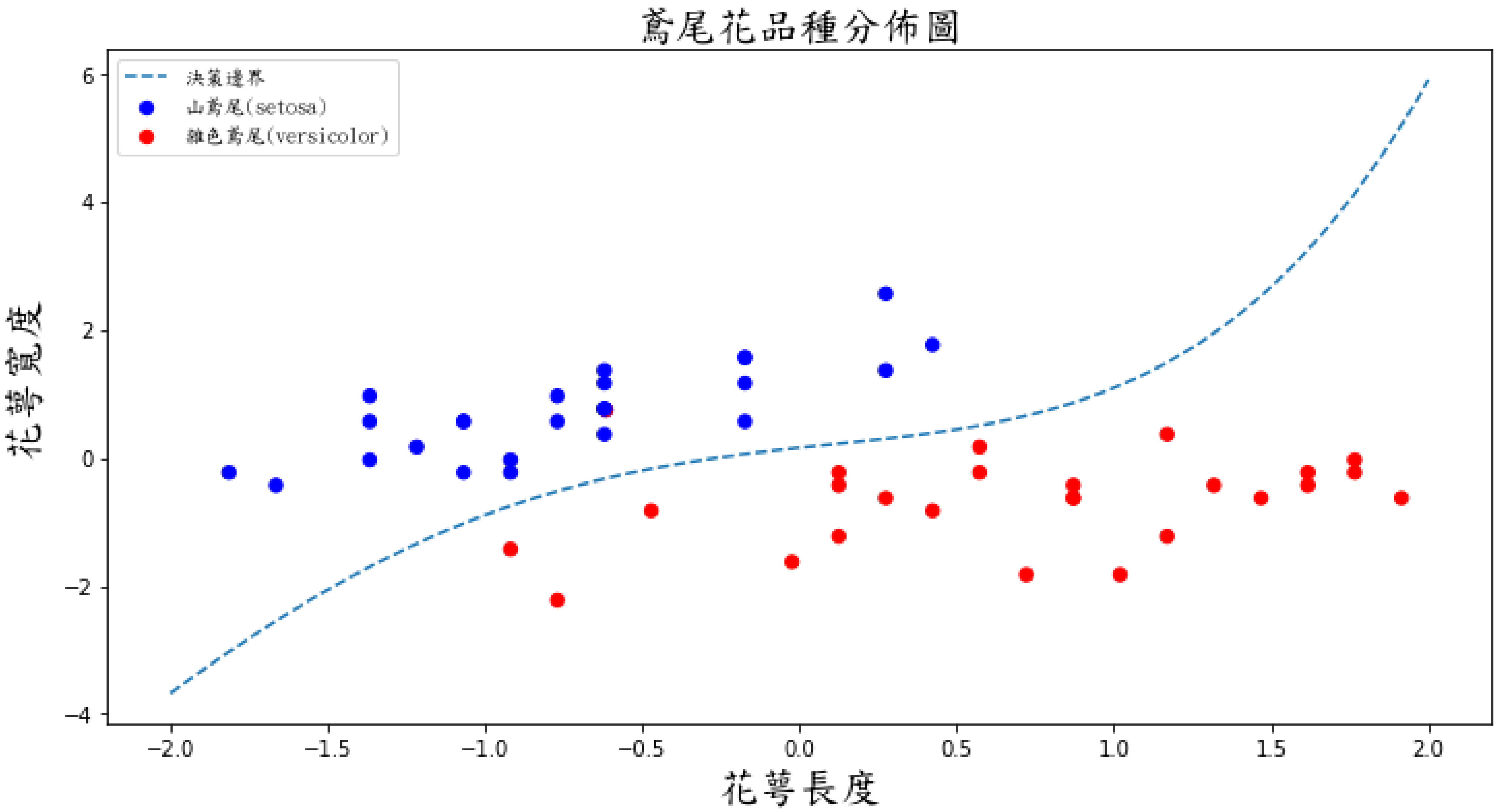
■ [矩陣W] = [矩陣W] − η[矩陣X]^T × ([矩陣Ŷ] −[矩陣Y])

利用邏輯迴歸做二元曲線決策邊界

- 實驗目的：以不同次方的曲線做分類的決策邊界。
- 觀察重點：
  - 逐漸調高迴歸模型函數的次方：
    - 觀察迴歸函數圖和訓練集資料、驗證集資料間的分佈變化。
    - 比較由迴歸函數算出來訓練集資料、驗證集資料的平均損失值的差距。
  - 逐漸增加模型訓練的次數：
    - 觀察迴歸函數圖和訓練集資料、驗證集資料間的分佈變化。
    - 比較由迴歸函數算出來訓練集資料、驗證集資料的平均損失值的差距。

實作2-3

承實作2，利用邏輯迴歸做二元曲線決策邊界



單元6

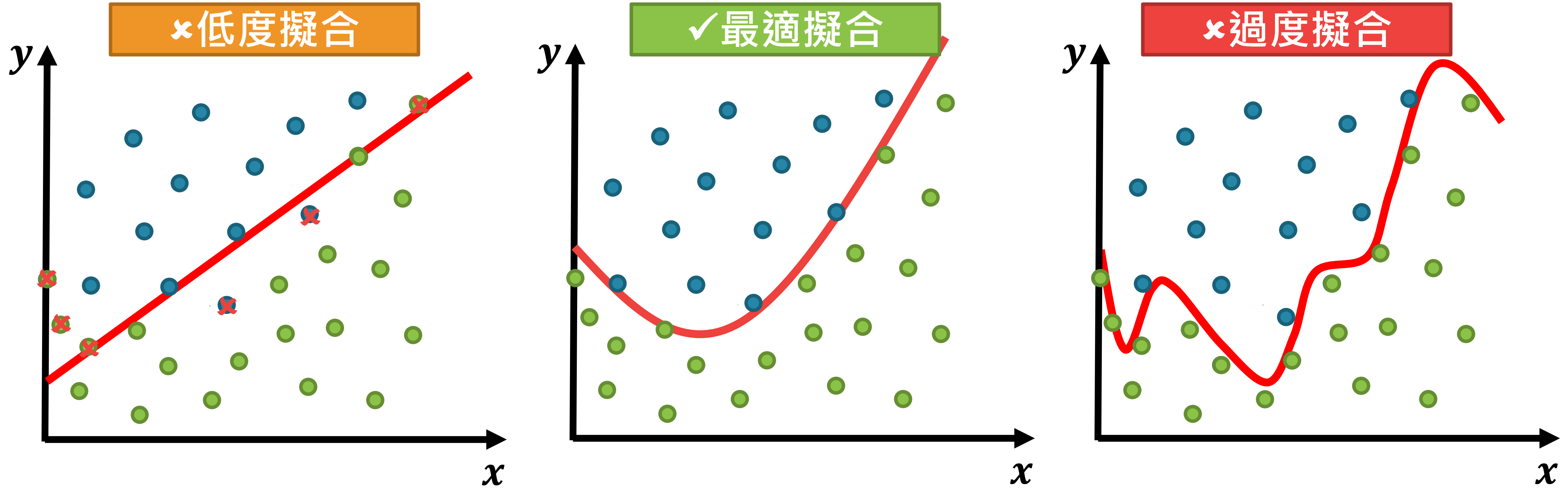
分類的過度擬合

過度擬合Over fitting

多項式函數的特性是次方越高，轉折點(微分為零的位置)越多，越容易擬合訓練數據的分佈，損失函數會變很小。

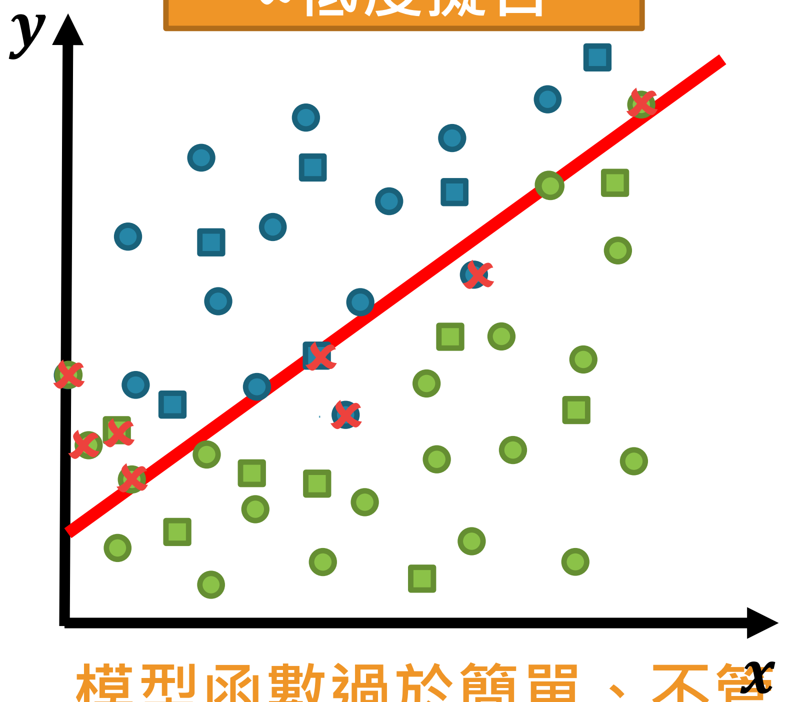
例：3次方的多項式函數，導函數為2次方，其函數圖最多有2個轉折點。

例：4次方的多項式函數，導函數為3次方，其函數圖最多有3個轉折點。



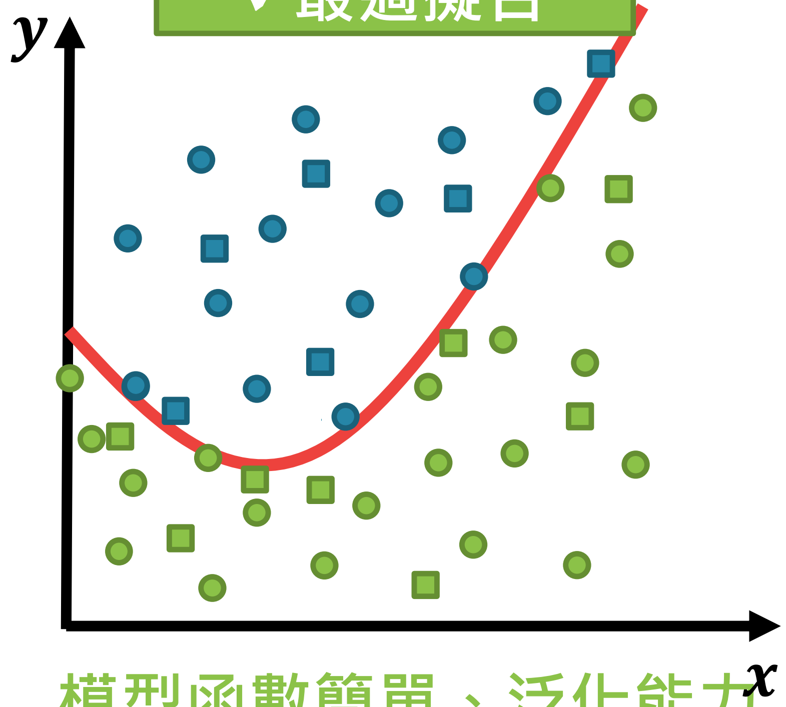
過度擬合Over fitting

✘低度擬合



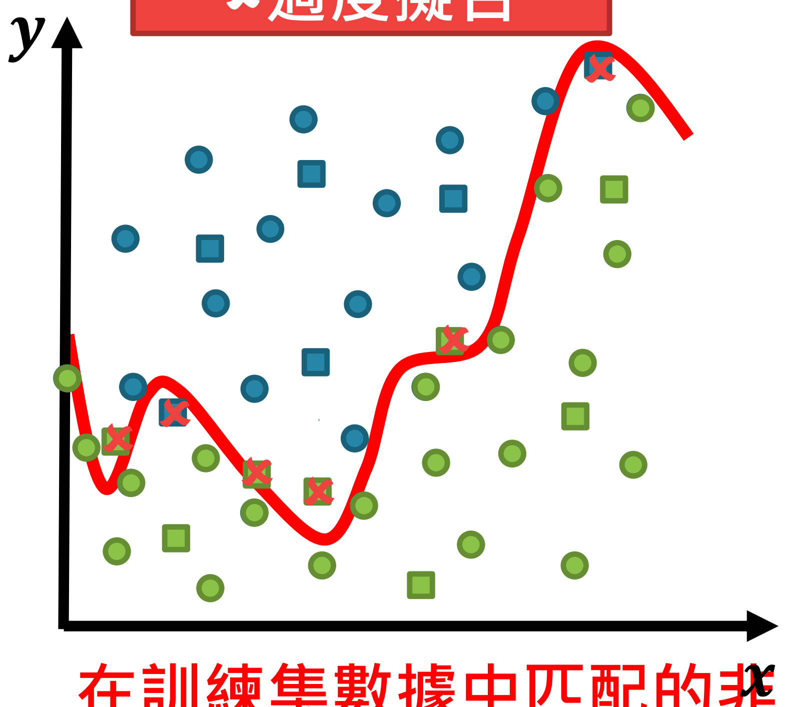
模型函數過於簡單、不管用訓練集數據或驗證集數據測試，發現模型結果差距都非常大。

✔最適擬合



模型函數簡單、泛化能力好，不管用訓練集數據或驗證集數據測試，發現模型結果差距都非常小。

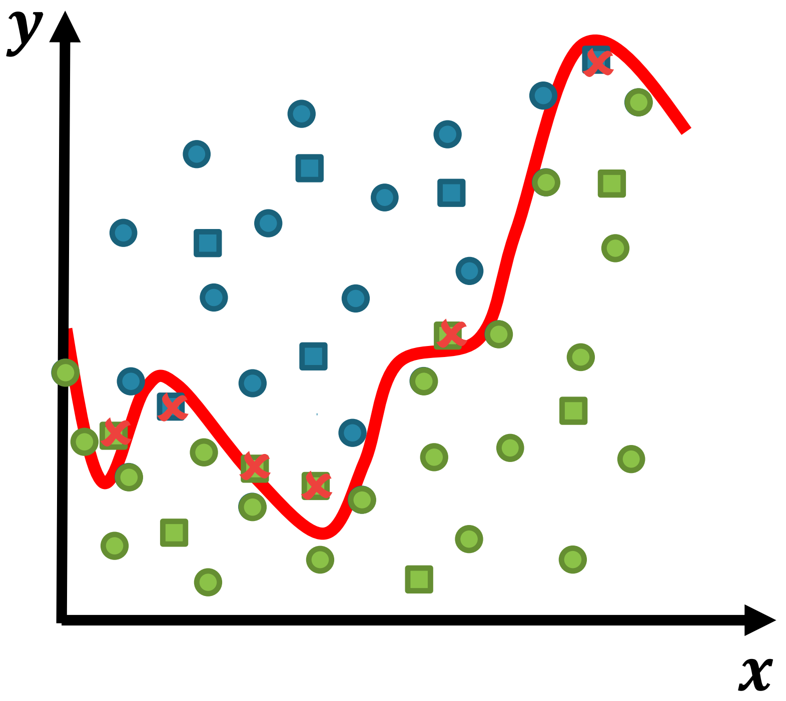
✘過度擬合



在訓練集數據中匹配的非常完美；但在測試集數據中偏差嚴重。

過度擬合Over fitting

- 過度擬合的解決方法：
  - 增加樣本數量
  - 減少模型複雜度
  - 減少訓練次數



單元7

多特徵(3個以上)的分類

3個以上多特徵邏輯迴歸

- 假設要用n個特徵資料(x_1、x_2、x_3、、、x_n)來做二元分類，分類的決策邊界方程式：$b+w_1x_1+w_2x_2+w_3x_3$、、、$+w_nx_n=0$
  □偏值和權重有n+1個：b、w_1、w_2、w_3、、、w_n

- 第k筆資料的特徵資料(x_{k1}、x_{k2}、x_{k3}、、、、x_{kn})
  □線性加權運算後$z_k=b+w_1x_{k1}+w_2x_{k2}+w_3x_{k3}$、、、$+w_nx_{kn}$
  □預測值$F(z_k)=\frac{1}{1+e^{-z_k}}=p_k$(分類標示為1的機率)。

- 損失函數

$$L=\sum-[y_k\times lnp_k+(1-y_k)\times ln(1-p_k)]=\sum[ln(1+e^{z_k})-y_k\times z_k]$$

用矩陣運算模型的預估值

$$\hat{Y}=\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_K \end{bmatrix} = G\left(\begin{bmatrix} b + w_1x_{11} + w_2x_{12} + w_3x_{13} + \text{、 、 、} + w_nx_{1n} \\ b + w_1x_{21} + w_2x_{22} + w_3x_{23} + \text{、 、 、} + w_nx_{2n} \\ b + w_1x_{31} + w_2x_{32} + w_3x_{33} + \text{、 、 、} + w_nx_{3n} \\ \vdots \\ b + w_1x_{K1} + w_2x_{K2} + w_3x_{K3} + \text{、 、 、} + w_nx_{Kn} \end{bmatrix} \right)$$

$$= G\left(\begin{array}{cc} & \xrightarrow{\text{特徵數量 n}} \\ \text{樣本數K}\big\downarrow & \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{K1} & x_{K2} & x_{K3} & \cdots & x_{Kn} \end{bmatrix} \end{array} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} + b \right)$$

[預估值矩陣$\hat{Y}$] =G([特徵資料矩陣X] [權重矩陣W]+偏值b)

梯度下降法更新迴歸係數

b, w_1, w_2, 、、、 w_n $\rightarrow z_k \rightarrow L$

$$z_k = b + w_1 x_{k1} + w_2 x_{k2} + w_3 x_{k3} \text{、、、} + w_n x_{kn}$$

$$L = \sum_{k=1}^{K} [ln(1 + e^{z_k}) - y_k \times z_k]$$

$$\frac{\partial z_k}{\partial b} = 1 \text{ 、 } \frac{\partial z_k}{\partial w_1} = x_{k1} \text{ 、 } \frac{\partial z_k}{\partial w_2} = x_{k2}$$

$$\text{、 、 、 } \frac{\partial z_k}{\partial w_n} = x_{kn}$$

$$\frac{\partial L}{\partial z_k} = \sum_{k=1}^{K} \left(\frac{1}{1 + e^{z_k}} e^{z_k} - y_k\right)$$

$$= \sum_{k=1}^{K} \left(\frac{1}{1 + e^{-z_k}} - y_k\right)$$

$$= \sum_{k=1}^{K} (\hat{y}_k - y_k)$$

梯度下降法更新迴歸係數

■ b、w_1、w_2、、、和w_n的梯度下降修正方程式為：

$$b = b - \eta \frac{\partial L}{\partial b} = b - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k) \quad 、 \quad w_1 = w_1 - \eta \frac{\partial L}{\partial w_1} = w_1 - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k) x_{k1}$$

$$w_2 = w_2 - \eta \frac{\partial L}{\partial w_2} = w_2 - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k)\, x_{k2}$$

$$w_3 = w_3 - \eta \frac{\partial L}{\partial w_3} = w_3 - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k)\, x_{k3} \quad 、、、$$

$$w_n = w_n - \eta \frac{\partial L}{\partial w_n} = w_n - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k)\, x_{kn}$$

梯度下降法更新迴歸係數

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} - \eta \begin{bmatrix} (\hat{y}_1 - y_1)x_{11} + (\hat{y}_2 - y_2)x_{21} + (\hat{y}_3 - y_3)x_{31} + 、、 + (\hat{y}_3 - y_3)x_{K1} \\ (\hat{y}_1 - y_1)x_{12} + (\hat{y}_2 - y_2)x_{22} + (\hat{y}_3 - y_3)x_{32} + 、、 + (\hat{y}_3 - y_3)x_{K2} \\ (\hat{y}_1 - y_1)x_{13} + (\hat{y}_2 - y_2)x_{23} + (\hat{y}_3 - y_3)x_{33} + 、、 + (\hat{y}_3 - y_3)x_{K3} \\ \vdots \\ (\hat{y}_1 - y_1)x_{1n} + (\hat{y}_2 - y_2)x_{2n} + (\hat{y}_3 - y_3)x_{3n} + 、、 + (\hat{y}_3 - y_3)x_{Kn} \end{bmatrix}$$

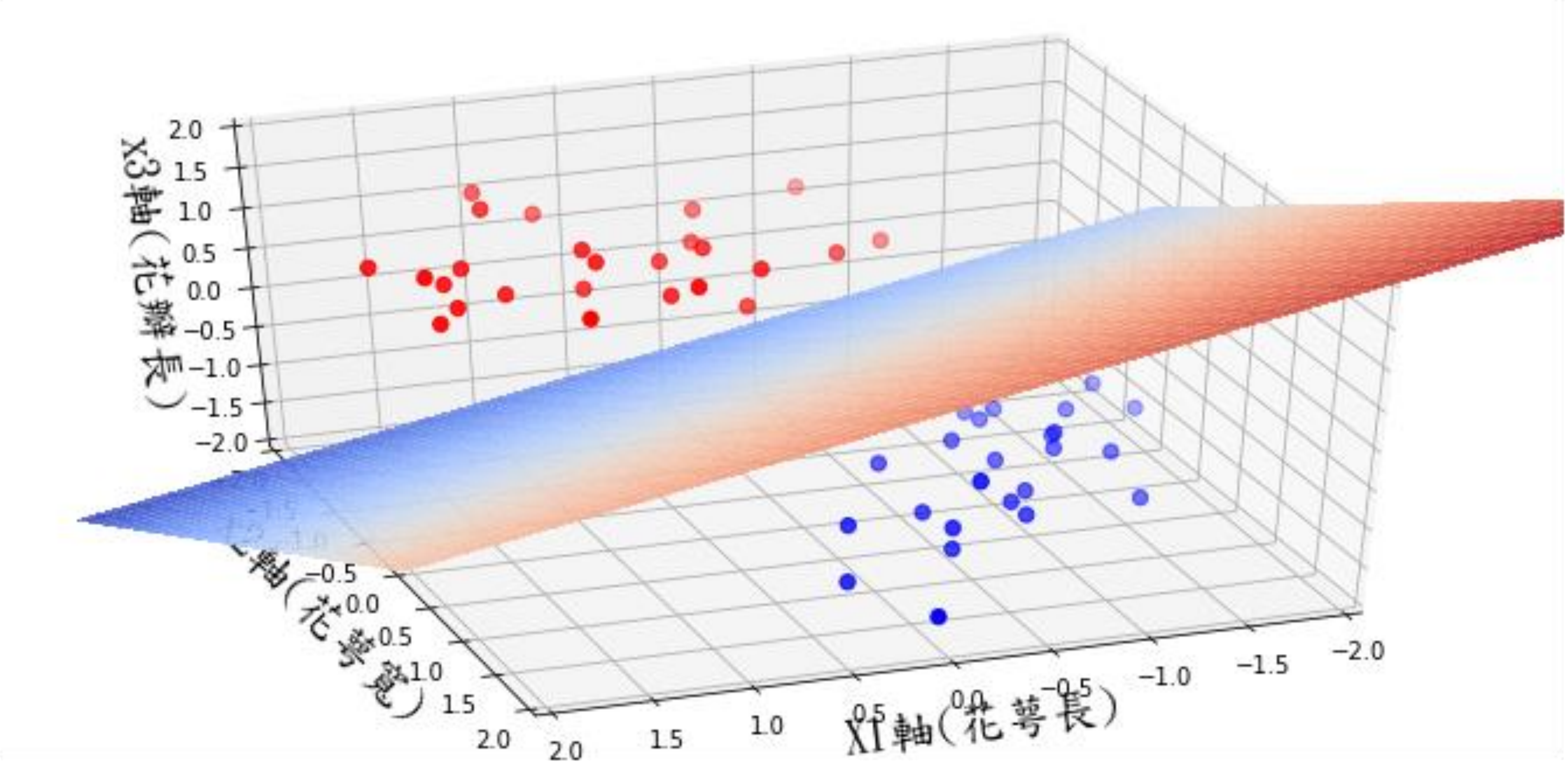
$$= \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} - \eta \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{K1} & x_{K2} & x_{K3} & \cdots & x_{Kn} \end{bmatrix}^T \times \left(\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_K \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_K \end{bmatrix} \right)$$

[矩陣W] = [矩陣W] − η[矩陣X]^T × ([矩陣Ŷ] −[矩陣Y])

實作2-4 利用邏輯迴歸做三個特徵的二元分類

- 取花萼長度(Sepal length)，花萼寬度(Sepal width)，花瓣長度(Petal length)3個特徵利用邏輯迴歸，找出鳶尾花分為山鳶尾(setosa)、雜色鳶尾(versicolor)兩類的決策邊界。
- 實作要求：
  - 如下圖，以三個座標軸的立體圖，呈現出分類邊界平面的效果。
  - 以驗證集資料測試其準確度。

實作2-4 利用邏輯迴歸做三個特徵的二元分類

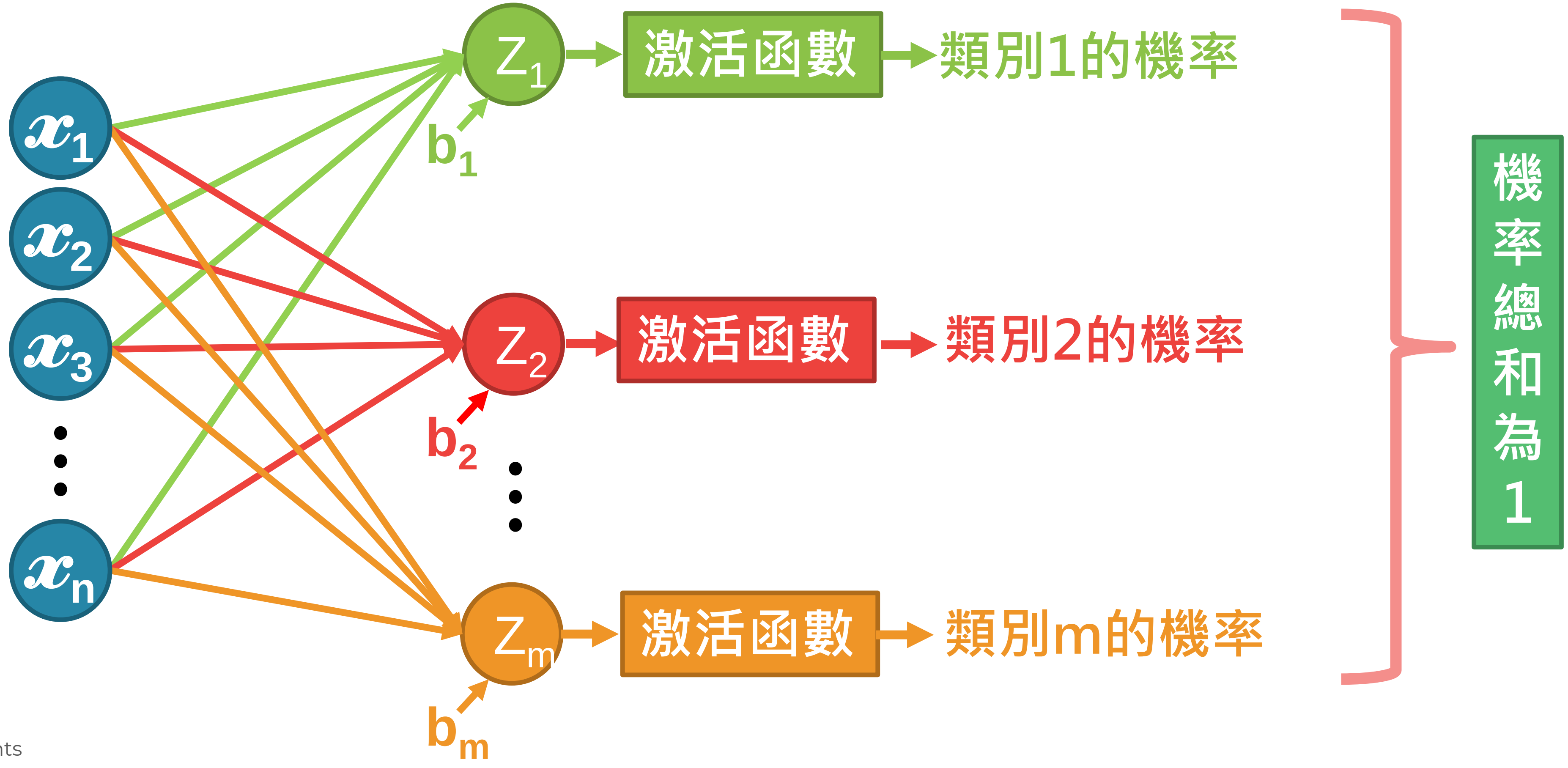


感知器處理多元分類

單元8

感知器如何做多元分類？

- 如果要利用樣本資料的n個特徵，將樣本分為m個類別，可以使用m個感知器為一組，每一個感知器分別計算屬於這個分類的機率高低。
- 但激活函數的部分必須修改，要符合所有類別的機率總和為1。



線性加權矩陣運算

輸出節點數m

輸入節點數n

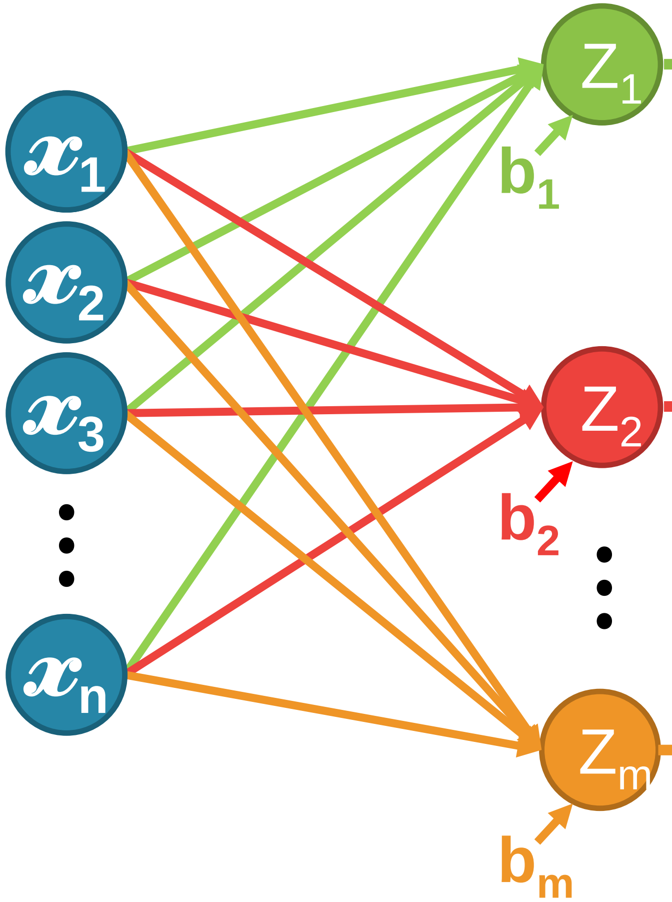
$$\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix}$$

$$\begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1m} \\ z_{21} & z_{22} & \cdots & z_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ z_{K1} & z_{K2} & \cdots & z_{Km} \end{bmatrix}$$

特徵數量n

樣本數K

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{K1} & x_{K2} & x_{K3} & \cdots & x_{Kn} \end{bmatrix}$$



$$z_{11} = x_{11}w_{11} + x_{12}w_{21} + .. + x_{1n}w_{n1} + b_1$$

$$z_{12} = x_{11}w_{12} + x_{12}w_{22} + .. + x_{1n}w_{n2} + b_2$$

$$z_{1m} = x_{11}w_{1m} + x_{12}w_{2m} + .. + x_{1n}w_{nm} + b_m$$

線性加權矩陣運算

輸出節點數m

輸入節點數 n

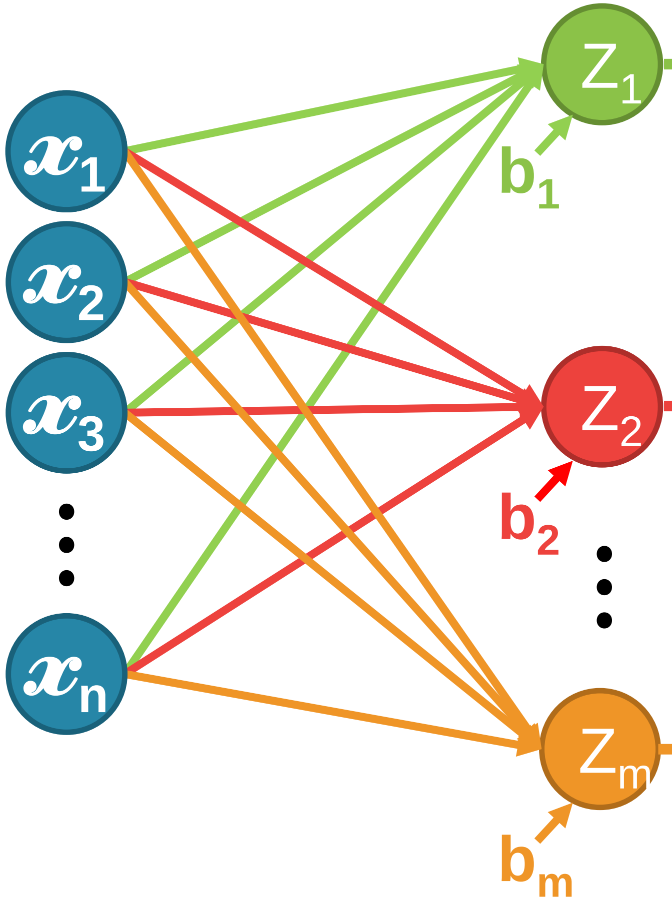
$$\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix}$$

$$\begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1m} \\ z_{21} & z_{22} & \cdots & z_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ z_{K1} & z_{K2} & \cdots & z_{Km} \end{bmatrix}$$

特徵數量 n

樣本數 K

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{K1} & x_{K2} & x_{K3} & \cdots & x_{Kn} \end{bmatrix}$$



$$z_{21} = x_{21}w_{11} + x_{22}w_{21} + .. + x_{2n}w_{n1} + b_1$$

$$z_{22} = x_{21}w_{12} + x_{22}w_{22} + .. + x_{2n}w_{n2} + b_2$$

$$z_{2m} = x_{21}w_{1m} + x_{22}w_{2m} + .. + x_{2n}w_{nm} + b_m$$

線性加權矩陣運算

輸出節點數m

輸入節點數n

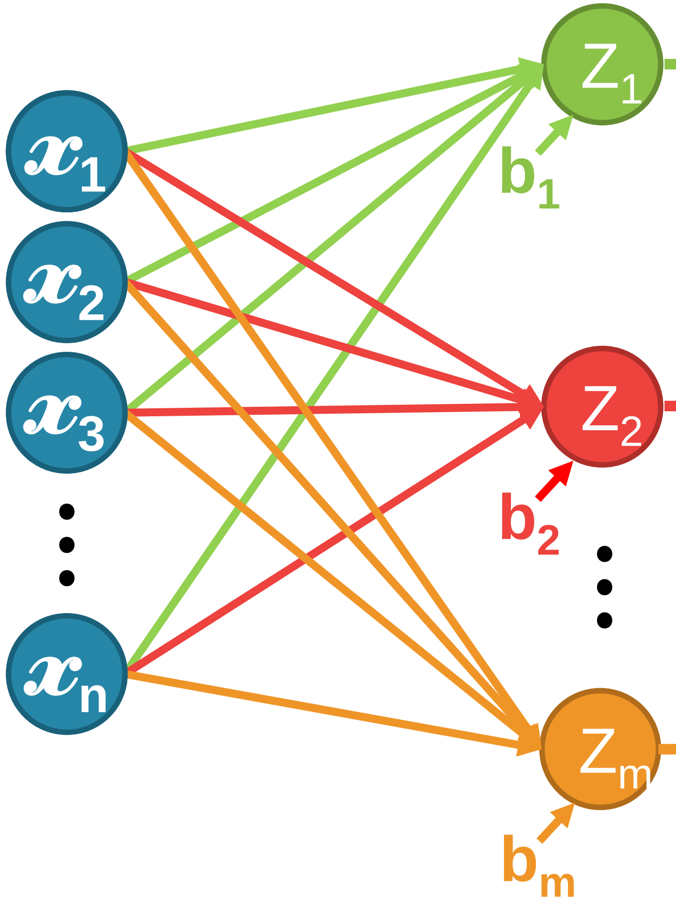
$$\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix}$$

$$\begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1m} \\ z_{21} & z_{22} & \cdots & z_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ z_{K1} & z_{K2} & \cdots & z_{Km} \end{bmatrix}$$

特徵數量n

樣本數K

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{K1} & x_{K2} & x_{K3} & \cdots & x_{Kn} \end{bmatrix}$$



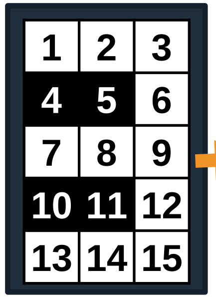
$$z_{K1} = x_{K1}w_{11} + x_{K2}w_{21}+..+x_{Kn}w_{n1} + b_1$$

$$z_{K2} = x_{K1}w_{12} + x_{K2}w_{22}+..+x_{Kn}w_{n2} + b_2$$

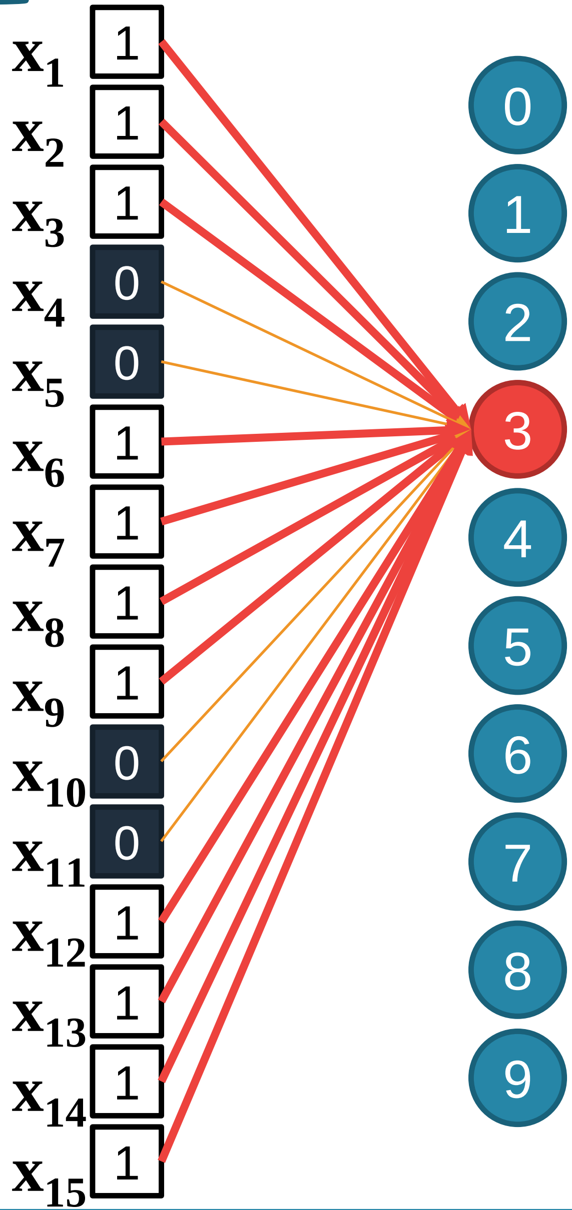
$$z_{Km} = x_{K1}w_{1m} + x_{K2}w_{2m}+..+x_{Kn}w_{nm} + b_m$$

如何利用特徵來做多元分類

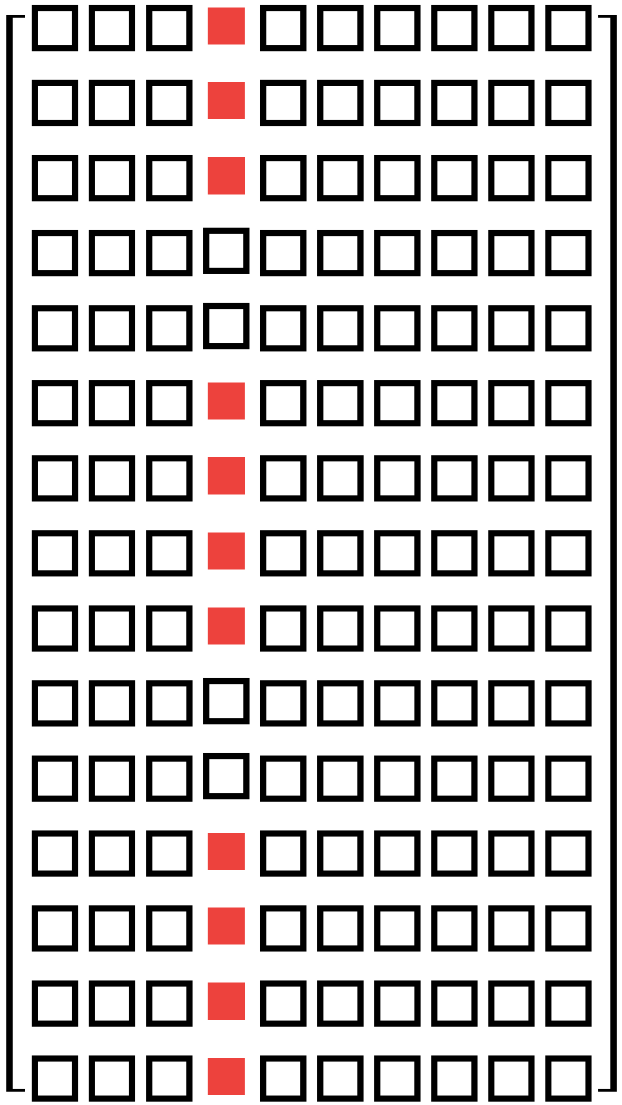
舉數字圖案做辨識為例子



每一個位置的像素亮度作為樣本特徵

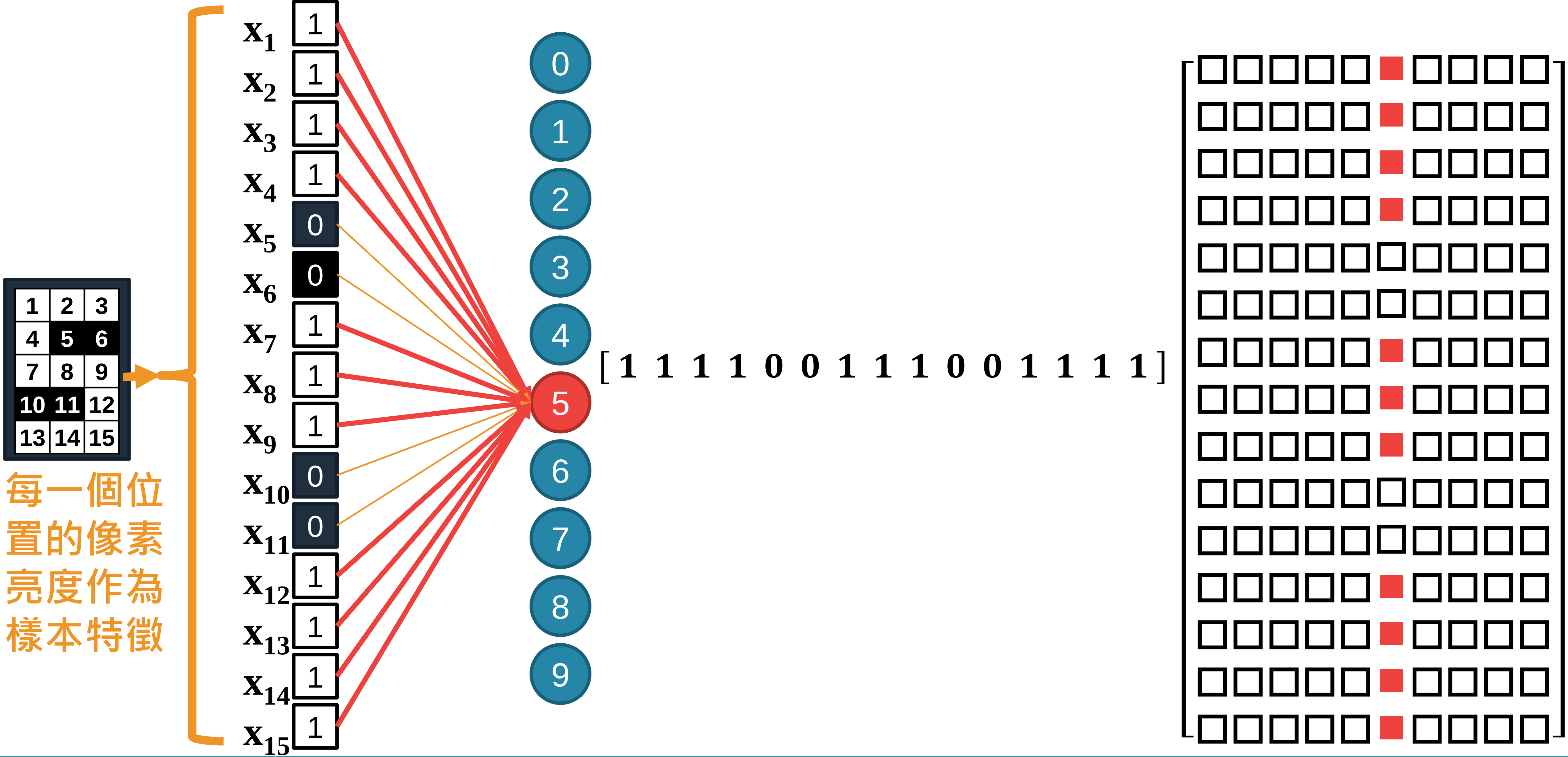


[1 1 1 0 0 1 1 1 1 0 0 1 1 1 1]



如何利用特徵來做多元分類

舉數字圖案做辨識為例子



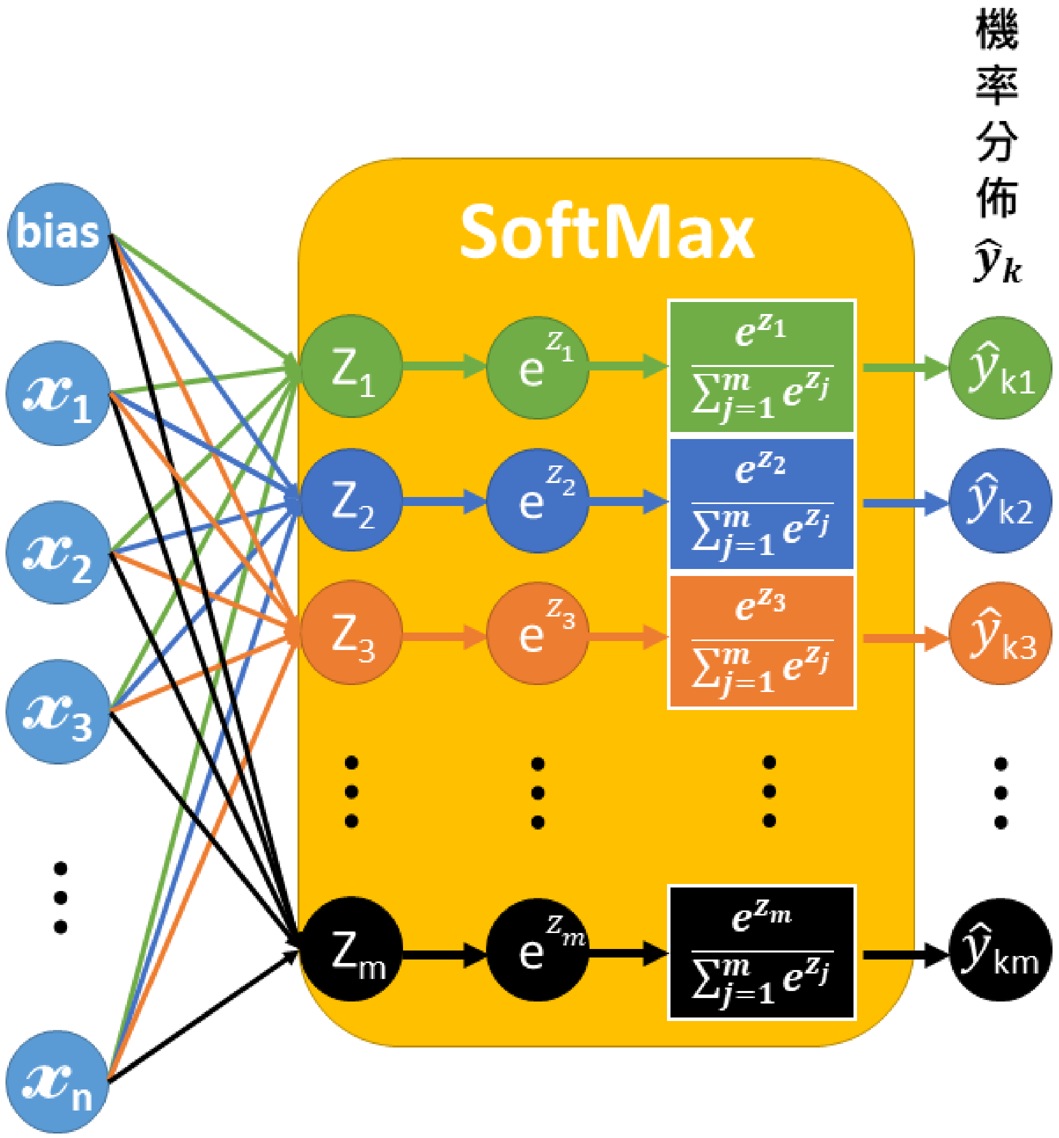
SoftMax激活函數

■ Softmax 將m個感知器的線性運算z值，先通過指數函數e^z處理。
口轉為全為正的數值。
口經過指數轉換拉大各感知器的數值差距。

■ 將指數總和$\sum_{j=1}^{m} e^{z_j}$作為分母，個別的輸出作為分子，得到m個小於1的數值，即為各類別的分類機率。

$$\hat{y}_{kj} = \frac{e^{z_j}}{\sum_{j=1}^{m} e^{z_j}}$$

■ Softmax迴歸模型是logistic迴歸模型在多元分類問題上的推廣，當分類數量等於 2 時，Softmax 迴歸與logistic迴歸是一致的。



多元分類的損失函數

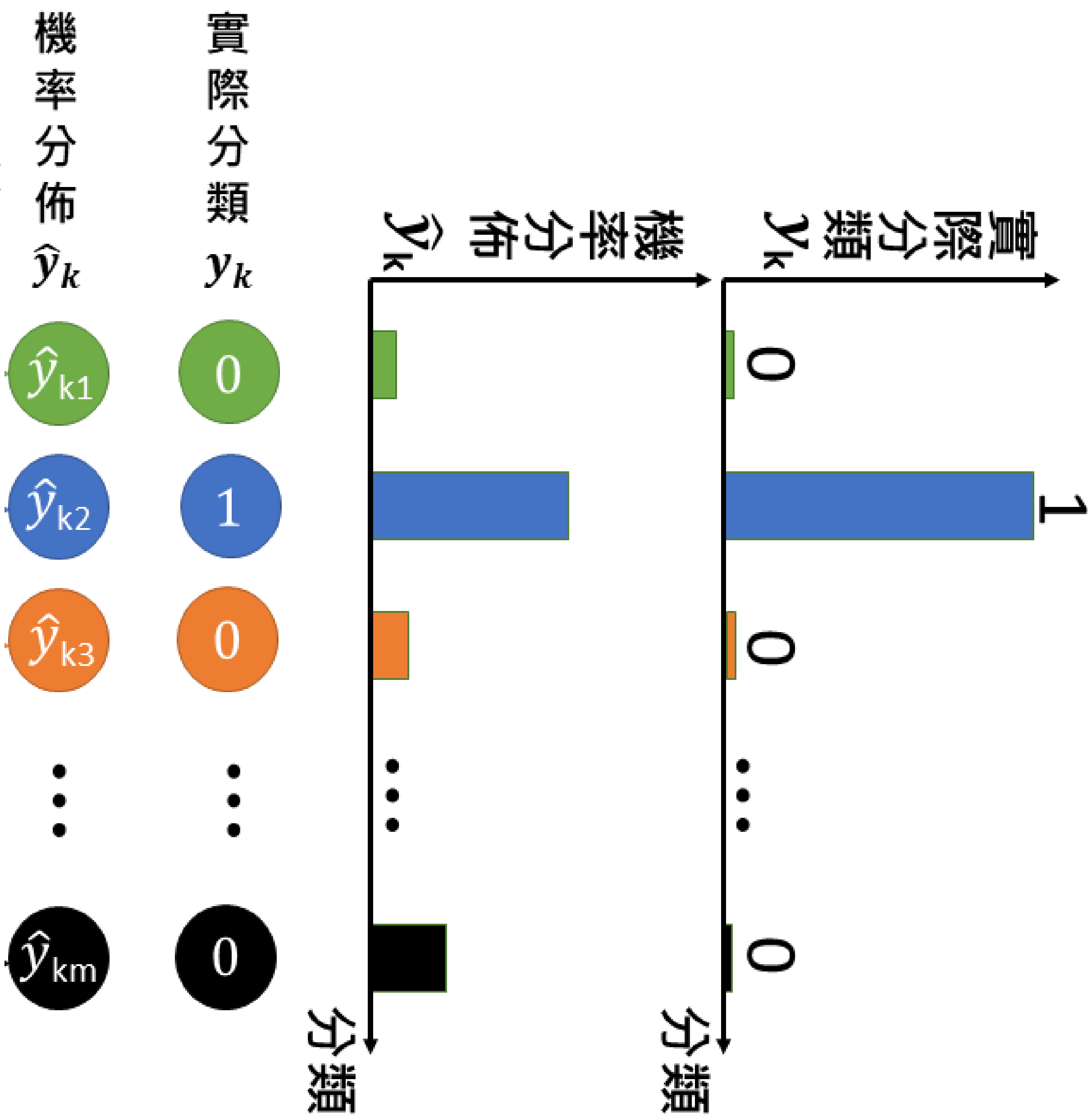
■ 多元分類的損失函數：仍使用cross entropy：

□ 單一第k個樣本的損失值

$$l_k = -\sum_{j=1}^{m} y_{kj} \cdot \ln(\hat{y}_{kj})$$

□ 所有樣本的損失值

$$\mathrm{L} = \sum_{k=1}^{K} l_k = -\sum_{k=1}^{K}\sum_{j=1}^{m} y_{kj} \cdot \ln(\hat{y}_{kj})$$



多元分類的梯度下降

- 損失函數對線性運算Z_{ki}的梯度：[證明見下頁，參考即可]

$$\frac{\partial L}{\partial Z_{ki}} = \sum_{k=1}^{K} (\hat{y}_{ki} - y_{ki})$$

- 函數L對線性運算的偏微分$\frac{\partial L}{\partial Z_{ki}}$仍為誤差總和，故後面$\frac{\partial L}{\partial \mathrm{w}}$、$\frac{\partial L}{\partial \mathrm{b}}$的梯度下降更新都相同。

實作2-5 利用SoftMax處理鳶尾花多元分類

- 建立一組多元分類的感知器：輸入資料有4個特徵，使用3個感知器為一組作為輸出。
- 3個感知器的輸出經過SoftMax函數處理後再輸出。
- 數據集分割為前100筆為訓練資料，後面剩下為測試資料。
- 利用訓練完成的多元分類感知器，以測試資料集驗證其準確度。