解密AI黑盒子

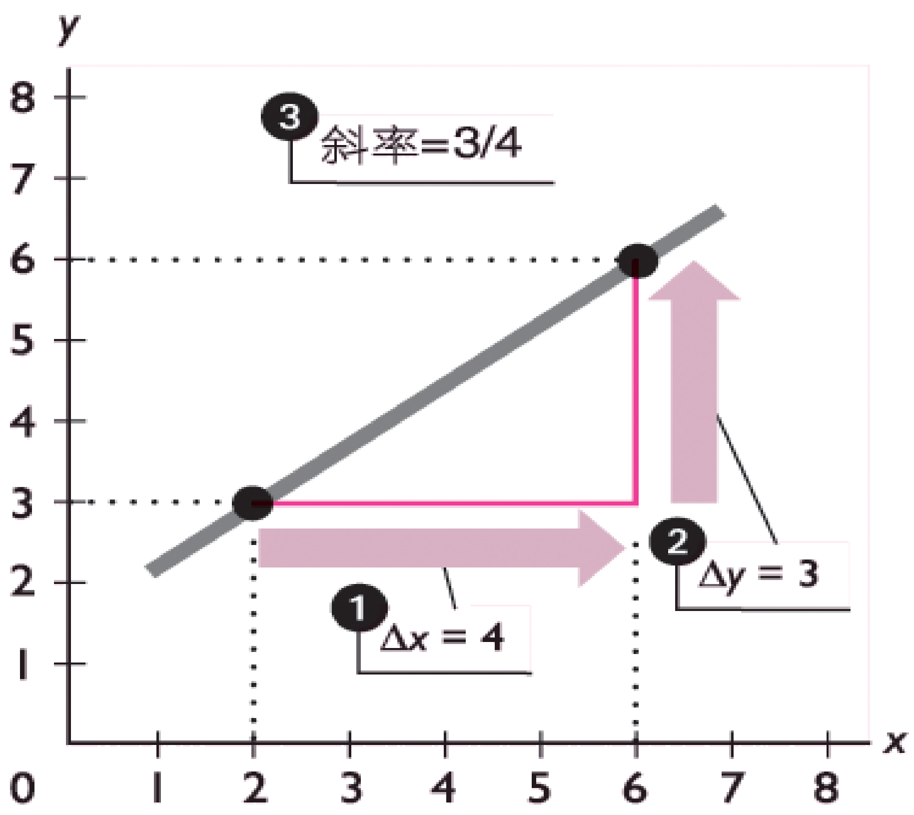
主題一：線性回歸與人工智慧理論

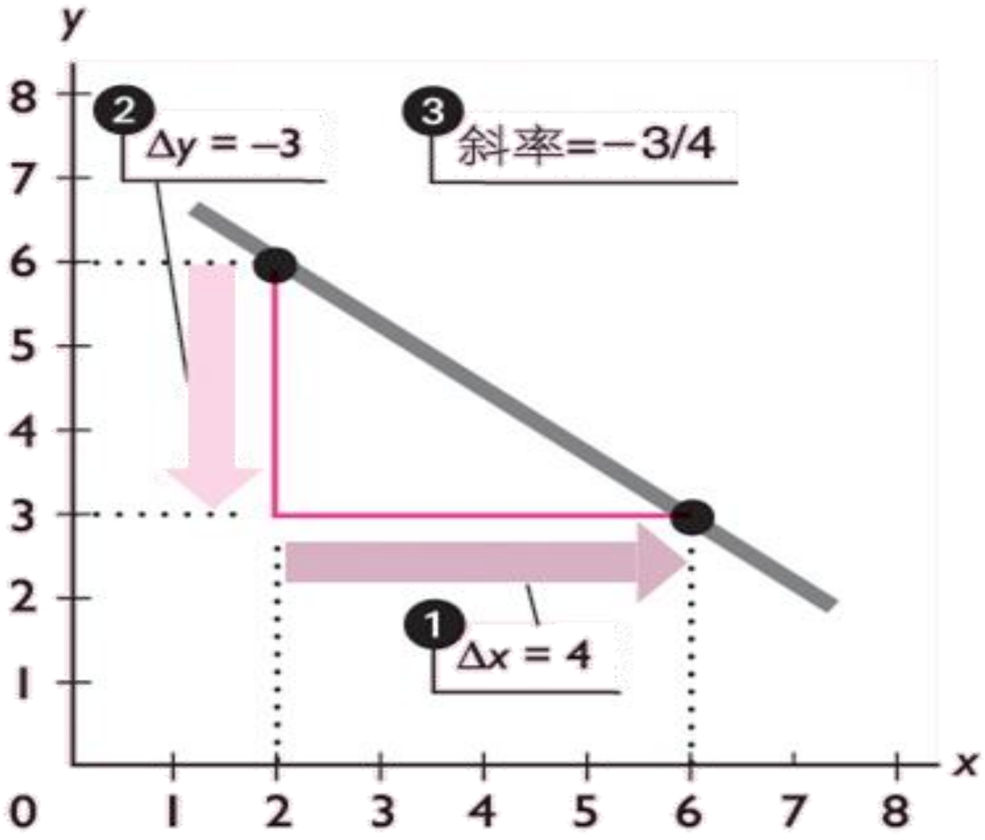
斜率和導函數

單元1

斜率和變化率

- 函數y=f(x)：用來表達依變量y和自變量x間的連續變化。
- 變化率=自變量每增加一單位，造成依變量(函數)的變化量。
- 斜率：直線傾斜程度的量度=$\dfrac{\text{縱軸變化量}\Delta y}{\text{橫軸變化量}\Delta x}$





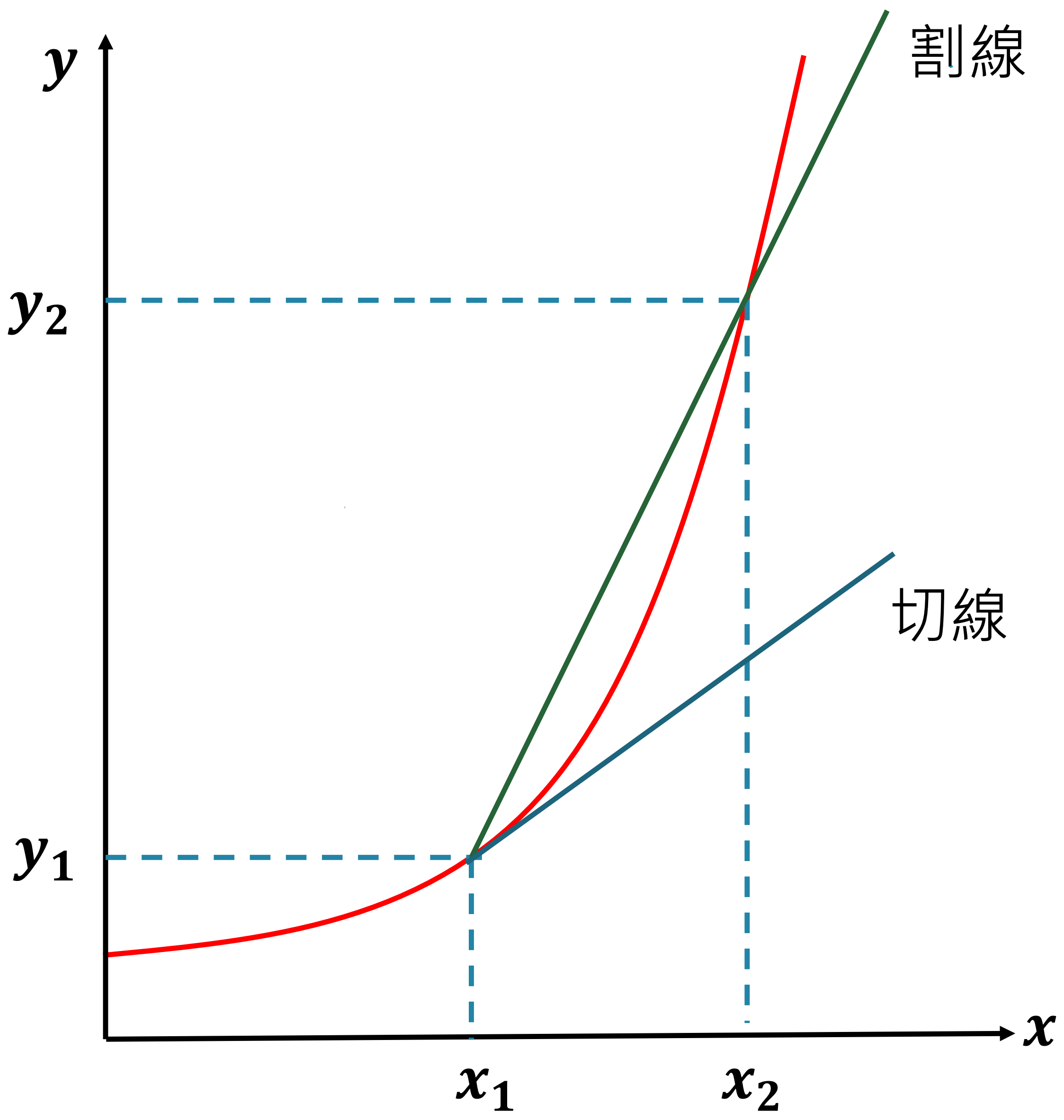
斜率和變化率

■ 平均變化率$=\dfrac{\Delta y}{\Delta x}=\dfrac{y_2-y_1}{x_2-x_1}$

=函數圖中通過兩點的割線斜率

■ 極限變化率$=\lim\limits_{\Delta x\to 0}\dfrac{\Delta y}{\Delta x}=\lim\limits_{\Delta x\to 0}\dfrac{y_2-y_1}{x_2-x_1}$

=函數圖中通過某點的切線斜率

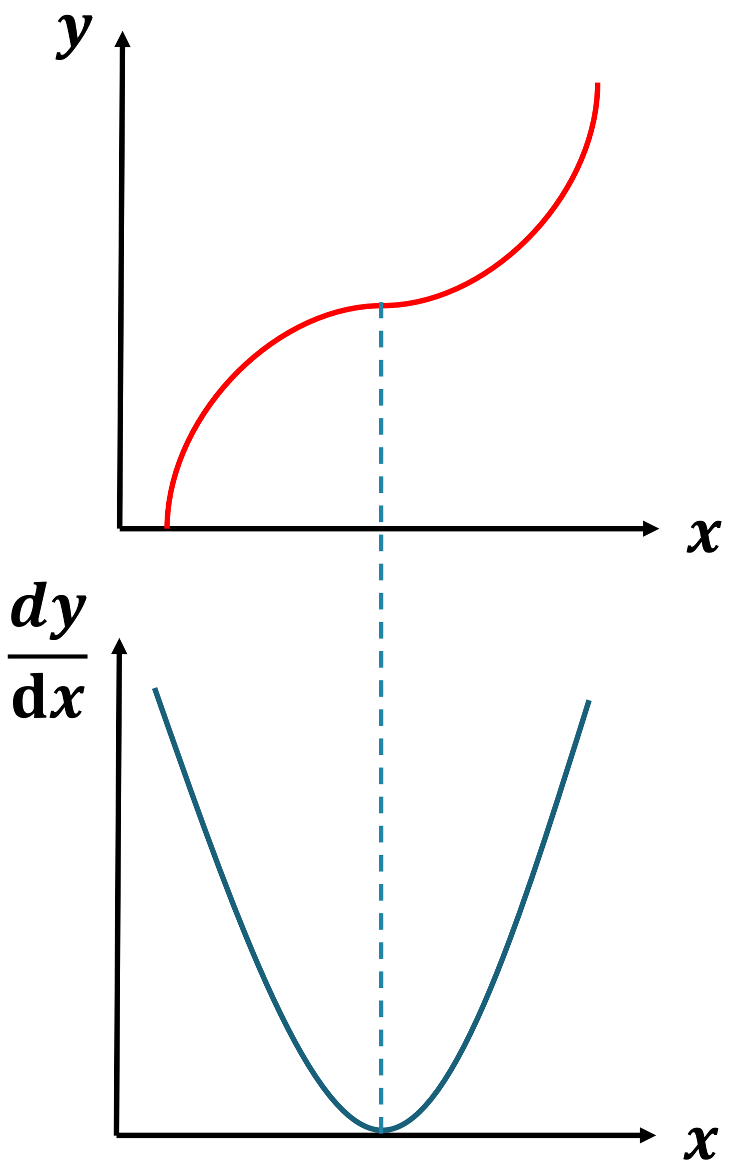


函數的微分(導函數)

■ 函數的微分(導函數)：求出原函數y=f(x)的切線斜率。

■ 微分的符號：$\lim\limits_{\Delta x \to 0} \frac{\Delta y}{\Delta x} = \frac{dy}{dx}$

■ 多項式微分規則：
❶指數乘到係數　❷指數減一次方　❸常數微分＝0
例：設多項式y=f(x)=$a_0 + a_1 x + a_2 x^2 + a_3 x^3 + ... + a_n x^n$
導函數：$\frac{dy}{dx} = a_1 + 2a_2 x + 3a_3 x^2 + ... + n a_n x^{n-1}$

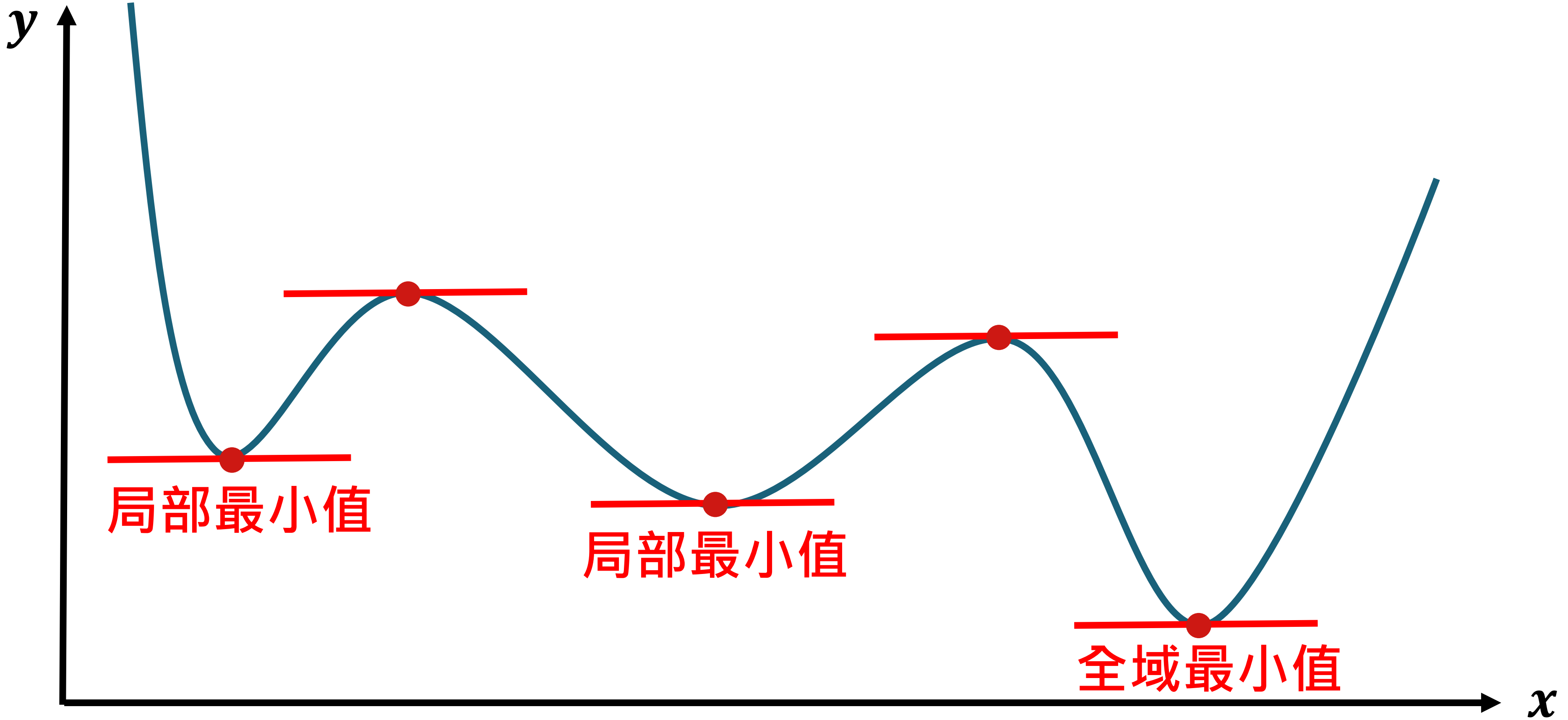


●動腦時間●

求出函數 y=f(x)=2x^4+x^3-3x^2+4x-8 的導函數。

函數的微分(導函數)

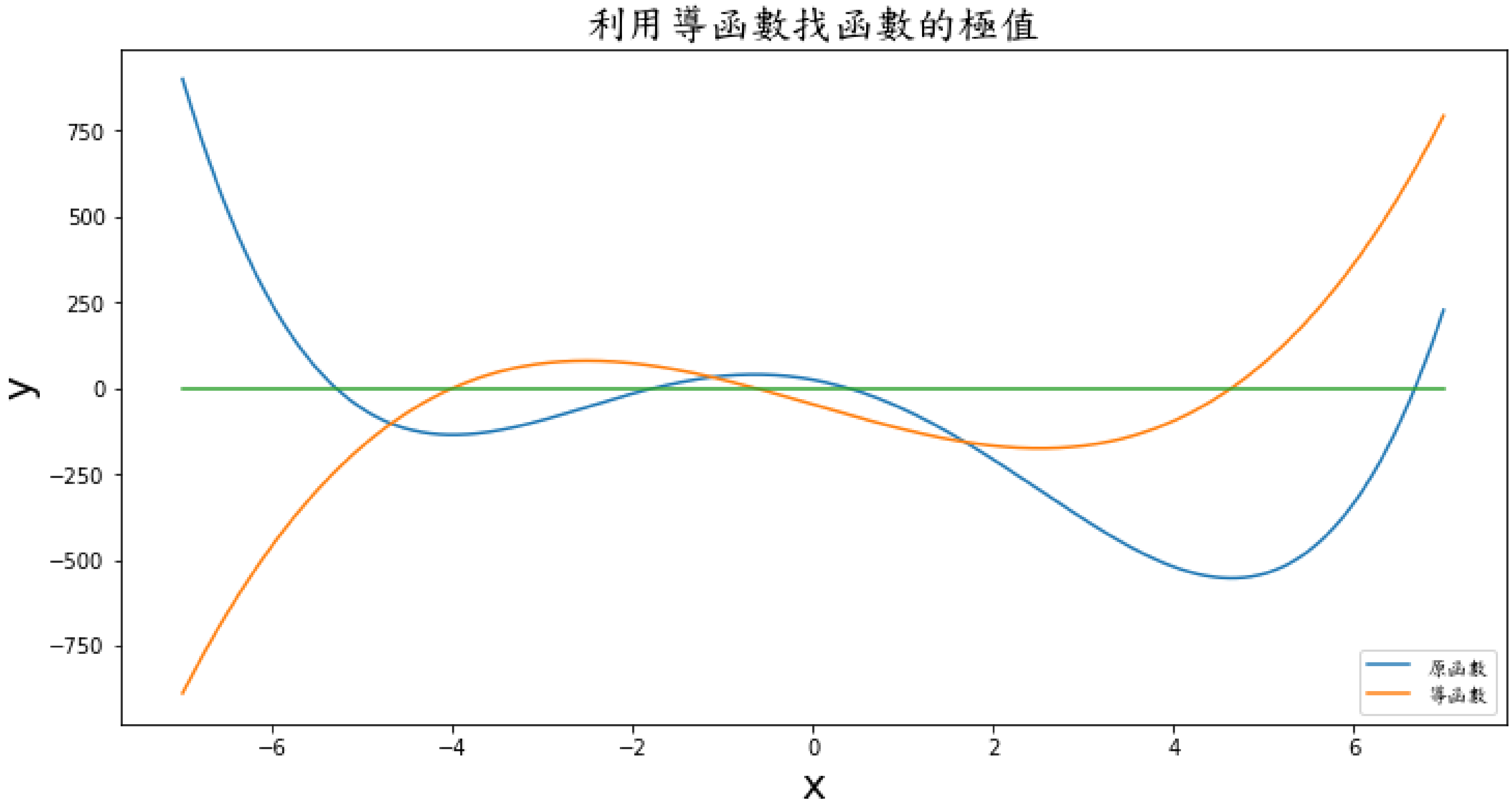
- 利用微分找函數的極值：當函數的導函數為零時(切線呈水平位置)，原函數會有極大值或極小值。
- 局部最小值可能會有好幾個位置，但全域最小值只有一個。



實作1-1 畫出多項式函數及其導函數的函數圖

- 實作目的：熟悉多項式函數的微分，透過導函數為零時尋找函數的極值，並能區分局部最小值和全域最小值。
- 實作要求：
  - 能利用多項式微分的規則，寫出多項式的導函數。
  - 能用 mathplotlib 套件畫出多項式函數及導函數的圖形。
  - 透過尋找導函數為零的位置，尋找函數的極值，並能區分局部最小值和全域最小值。

實作1-1 畫出多項式函數及其導函數的函數圖



微分的連鎖規則

加油錢 x $\xrightarrow{\frac{du}{dx}=0.03\text{公升/元}}$ 汽油容量 u $\xrightarrow{\frac{dy}{du}=12\text{公里/公升}}$ 行駛旅程 y

$\frac{dy}{dx}$=0.36公里/元

- 函數y=f(u) =f(g(x))⇨ $\frac{dy}{dx}=\frac{dy}{du}\times\frac{du}{dx}$

y對x的變化率$\frac{dy}{dx}$ $=$ y對u的變化率 $\frac{dy}{du}\times$ u對x的變化率 $\frac{du}{dx}$

●動腦時間●

求出下列函數的導函數

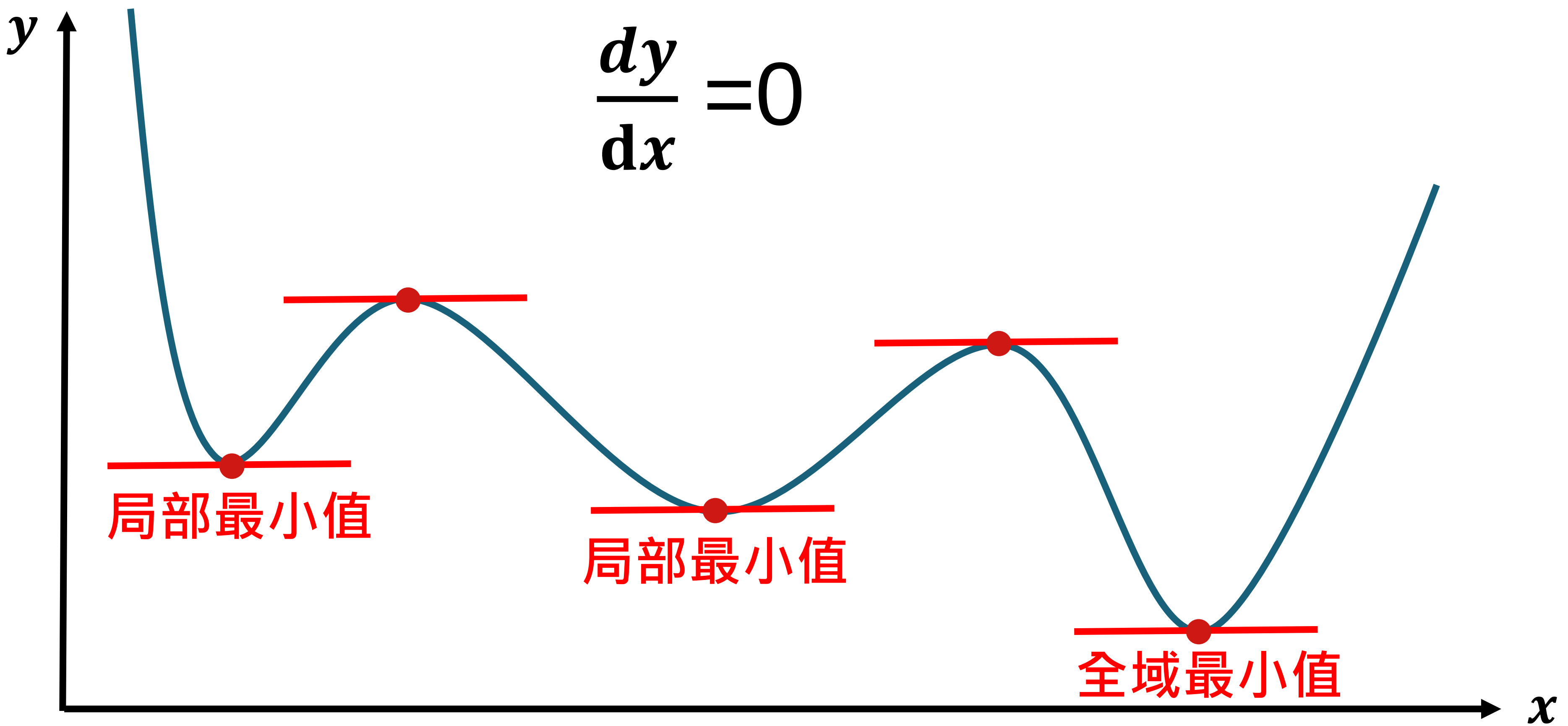
(1)$y=f(x)=(3-x+x^3)^2$ (2) $y=f(x)=(x^2+x-1)^{10}$

梯度下降算法
Gradient Descent

單元2

梯度下降算法

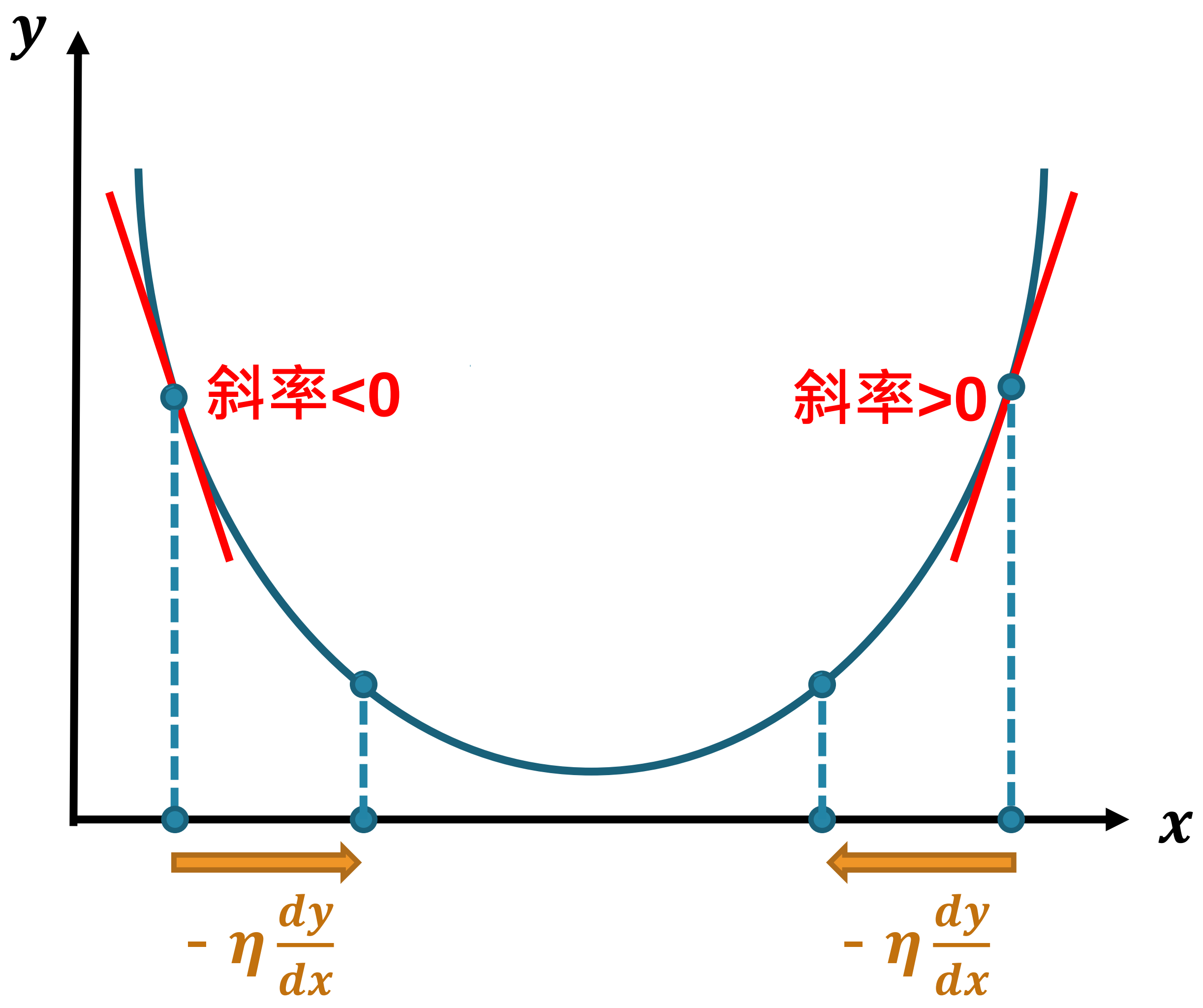
■ 利用導函數值為零來尋找函數的極值，必須求解方程式，不同函數要解的方程式各異，且不一定能解得出來，所以我們需要一個能找到函數最小值的統一作法。



梯度下降算法

- 梯度下降法：利用函數的斜率(梯度)，以逐步逼近的方法尋找函數的最小值。

- 梯度下降法的原理：
  - 函數的最小值在谷底：斜率=0的位置。
  - 若在函數圖形上切線斜率>0的位置：則谷底在向左移的方向。
  - 若在函數圖形上切線斜率<0的位置：則谷底在X向右移的方向。



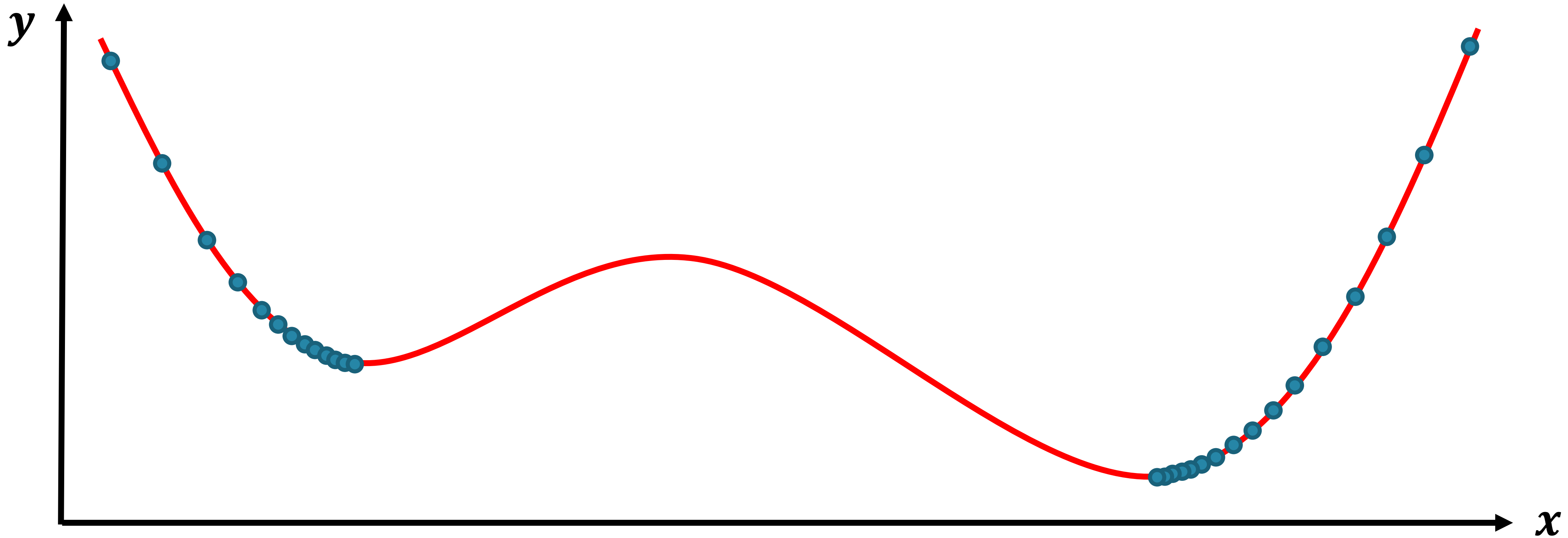
- 新的x的更新方程式為：

新的x=舊的x - $\eta\frac{dy}{dx}$

η為學習率：控制更新的幅度。

梯度下降算法

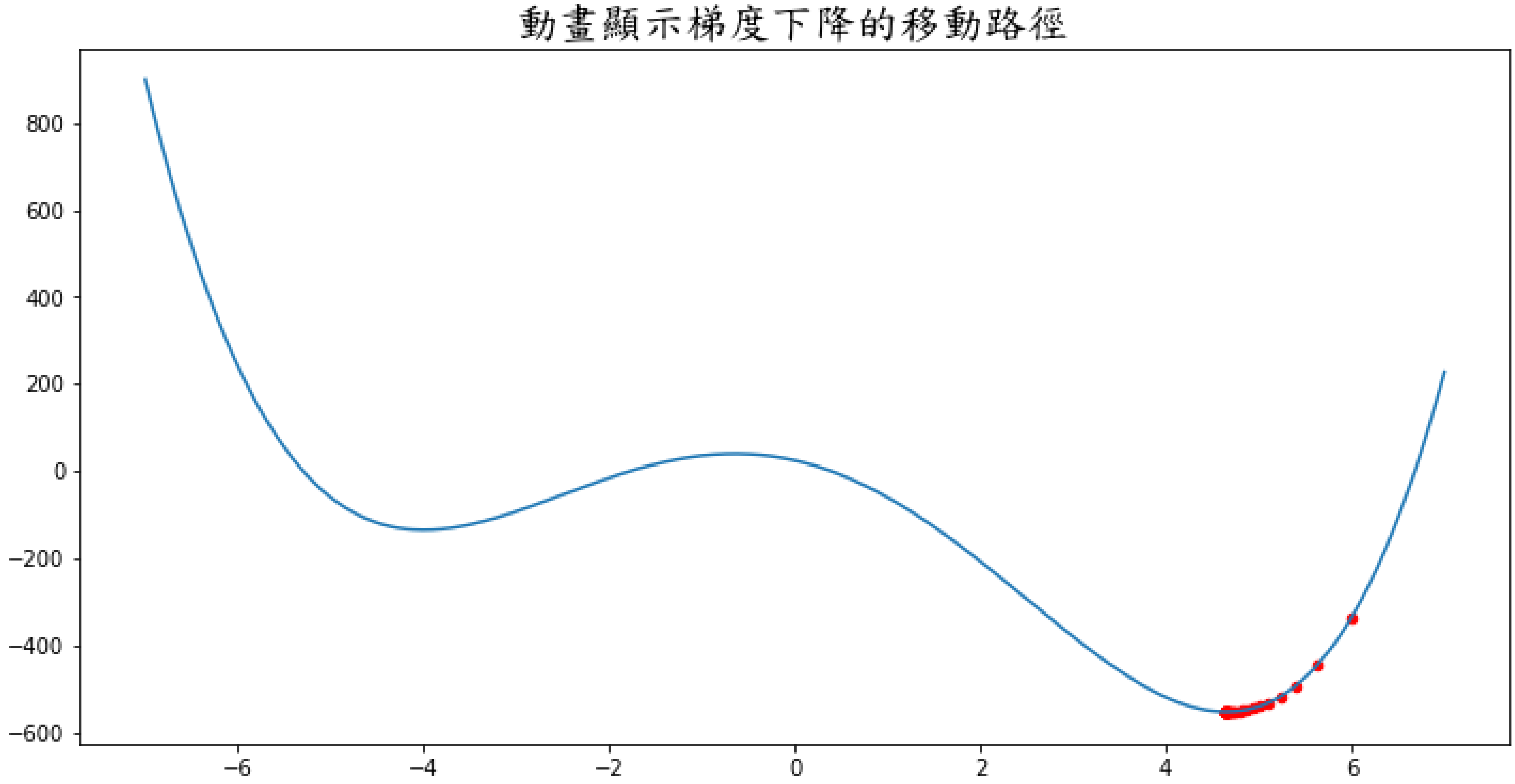
- 若函數有多個局部最小值，不同的起點可能會落到局部最小值，而不是全域最小值，梯度下降的起點隨機選定，以期能找到全域最小值。
- 梯度下降的終點可以用下列方式來判斷：
  - 方法一：設定梯度下降的更新學習次數，達到設定的更新次數即停止。
  - 方法二：檢查每次更新後的切線斜率，若接近 0 則非常靠近谷底。



實作1-2 梯度下降法逼近多項式函數最小值

- 實作目的：透過此實作能理解梯度下降法逼近函數最小值的方法，並用動畫展示如何一步步逼近函數最小值。
- 實作要求：
  - 能完成梯度下降的程式，並能動畫展示。
  - 能用修改梯度下降的起點，觀察最後求出梯度下降的終點，落在局部最小或是全域最小。
  - 能修改學習率的大小，觀察對梯度下降的速率和結果會造成什麼影響。

實作1-2 梯度下降法逼近多項式函數最小值

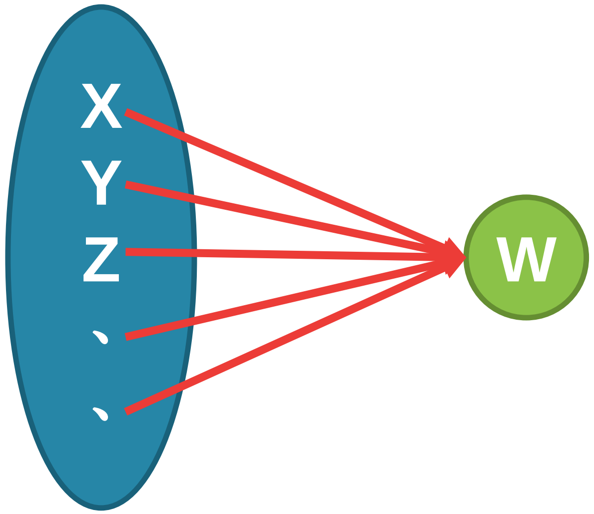


單元3

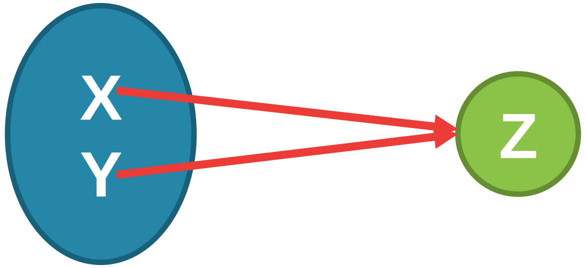
多變數函數
和
偏微分

多變數函數

■ 多變數函數：有序數對 (x,y,z, , ,) 有唯一的實數w與之對應，則稱w為點(x,y,z, , ,)之函數。
記作 w = f(x,y,z, , ,)



■ 雙變數函數：有序數對 (x,y) 有唯一的實數 z 與之對應，則稱 z 為點(x,y)之函數。
記作 z = f(x,y)



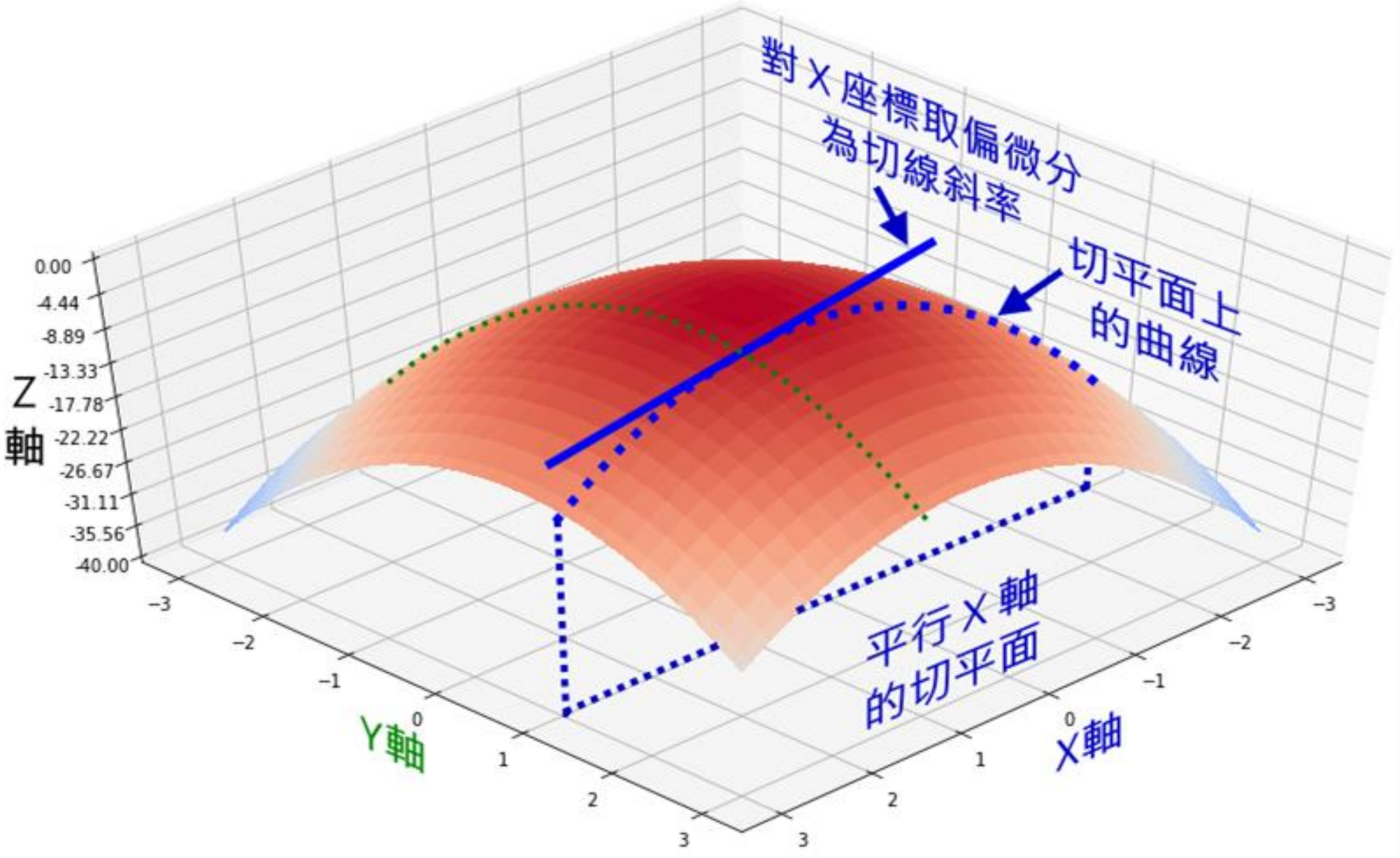
■ 超過兩個變數的函數，無法畫出其函數圖，所以本課程以討論雙變數函數為主。

雙變數函數的偏微分

- 偏微分：係指在沿某個軸(其它軸固定)的微分。

- 對x座標取偏微分，符號$\frac{\partial f}{\partial x}$：
將y座標視為定值，對x取微分。

- 幾何意義：為三維空間中的曲面
z = f(x, y) 與平行於x軸的鉛直面
相交曲線的切線斜率。

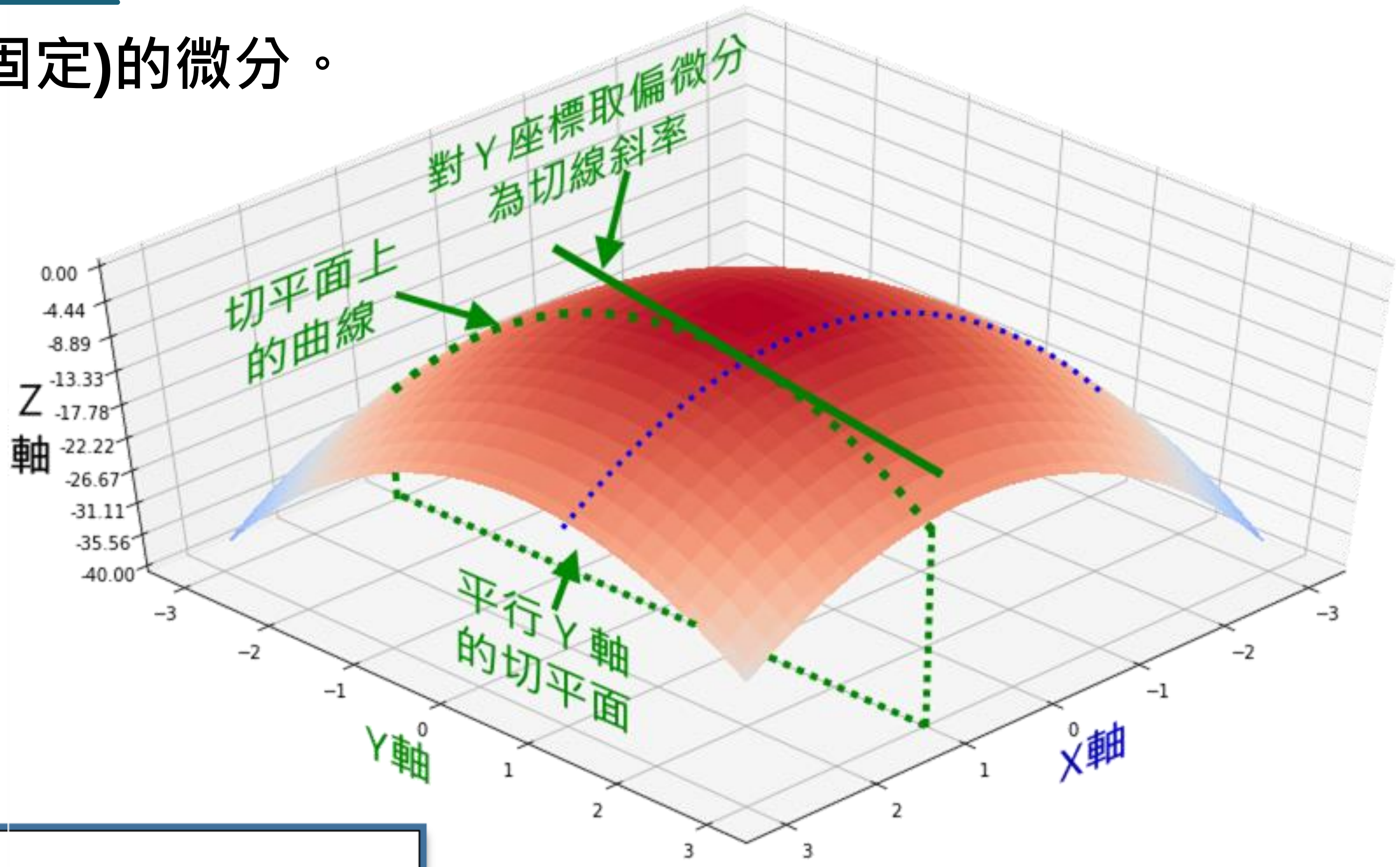


雙變數函數的偏微分

- 偏微分：係指在沿某個軸(其它軸固定)的微分。

- 對y座標取偏微分，符號$\frac{\partial f}{\partial y}$：
  將x座標視為定值，對y取微分。

- 幾何意義：為三維空間中的曲面 z = f(x, y) 與平行於y軸的鉛直面相交曲線的切線斜率。



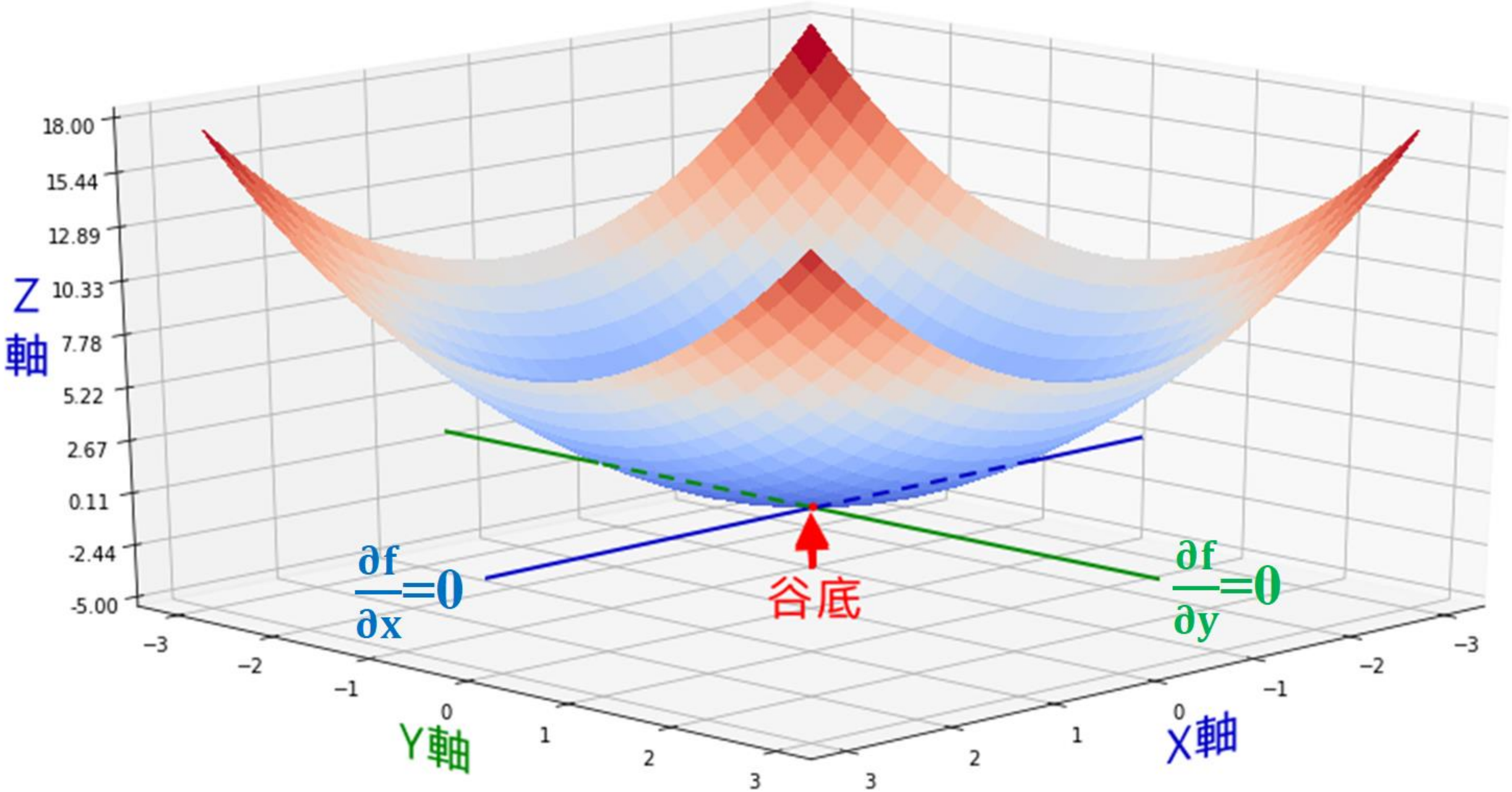
●動腦時間●

求出下列雙變數函數的偏微分$\frac{\partial f}{\partial x}$和$\frac{\partial f}{\partial y}$

(1) $z=f(x, y) =x^3+x^2y^3 - 2y^2$ (2) $z=f(x,y)=4-x^2-2y^2$

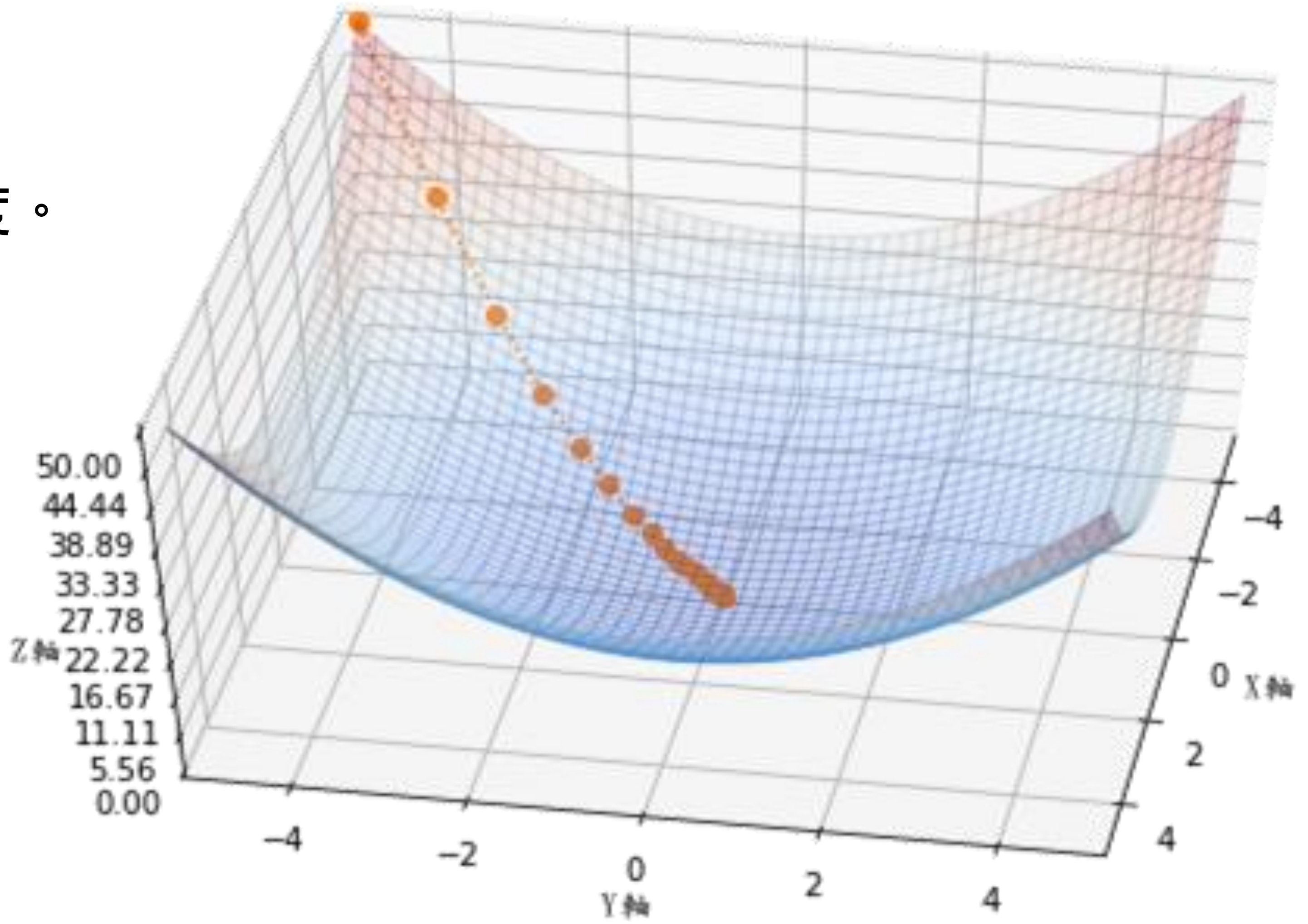
雙變數函數的極值

- 雙變數函數有極值的位置在對x偏微分為零和對y偏微分為零的位置。
- 可透過$\frac{\partial f}{\partial x}$ =0和$\frac{\partial f}{\partial y}$ =0，兩個方程式解出函數有極值的座標。



雙變數函數的梯度下降

- ■ 利用梯度下降求雙變數函數的極小值，其更新方程式為：
  - □ 新的x=舊的x$-\eta\frac{\partial f}{\partial x}$
  - □ 新的y=舊的y$-\eta\frac{\partial f}{\partial y}$
  - □ η為學習率：控制更新的幅度。

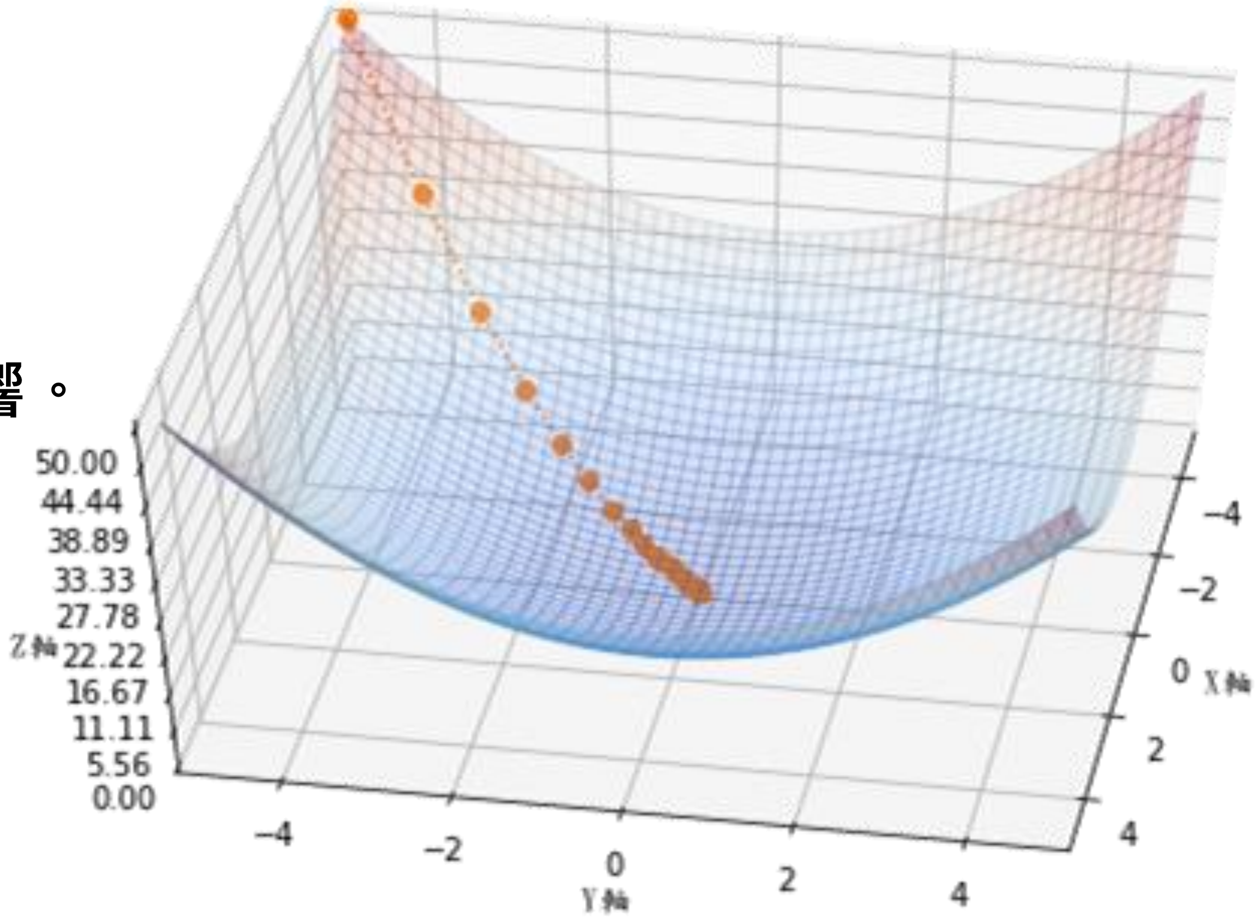


實作1-3 梯度下降法逼近雙變數函數最小值

■ 實作目的：透過此實作能理解梯度下降法逼近雙變數函數最小值的方法，並展示其更新路徑。

■ 實作要求：

- ☐ 能做雙變數函數的偏導函數。
- ☐ 能用修改梯度下降的起點，
  觀察最後求出梯度下降的終點。
- ☐ 能修改學習率的大小，觀察對
  梯度下降的速率和結果會造成什麼影響。



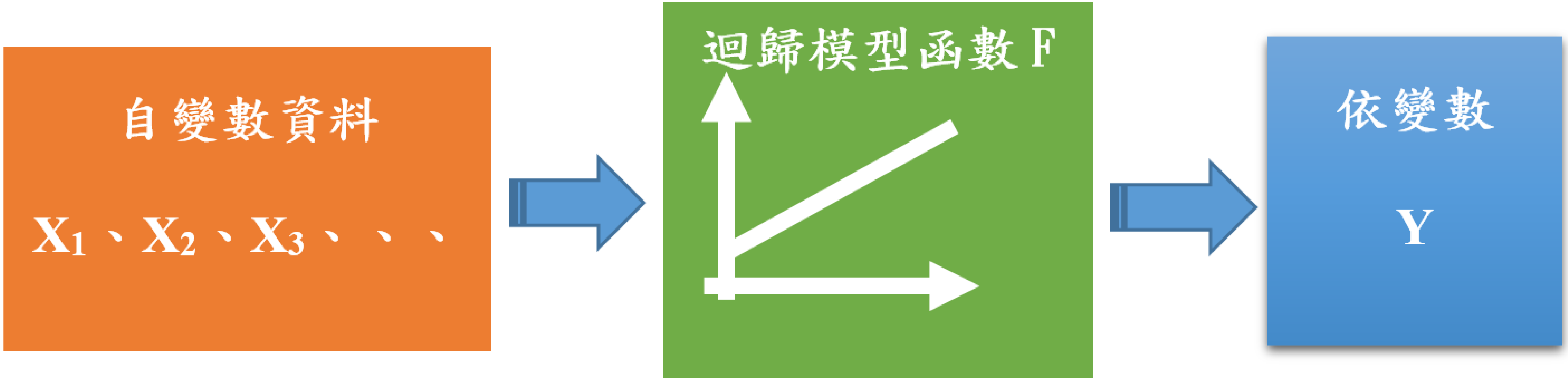
迴歸分析（Regression）

單元4

迴歸分析（Regression）

- 迴歸分析的目的在於透過過去資料找出自變數資料(x_1、x_2、x_3、、)和依變數資料y間存在的數學函數關係(稱為迴歸模型F)。

- 利用建立的迴歸模型來預測其他自變數造成可能的依變數結果。



迴歸的種類

依自變數多寡可分為：

- 簡單迴歸：求依變數與一個自變數的函數關係。

例：商品的電視廣告投放次數(自變數 x)和商品銷售數量(依變數 y)的關係。

廣告投放次數 x	24	22	15	4	9	20	5
商品銷售數量 y	591	543	410	310	319	520	338

- 複迴歸：求依變數與兩組以上自變數的函數關係。

例：母親的身高(自變數 x_1)、父親的身高(自變數 x_2)和成年子女身高(依變數 y)的關係。

母親的身高 x_1	153	160	170	163	148	173	150
父親的身高 x_2	171	183	177	170	165	189	163
子女身高 y	173	178	180	165	168	185	168

簡單線性迴歸(Simple Linear Regression)

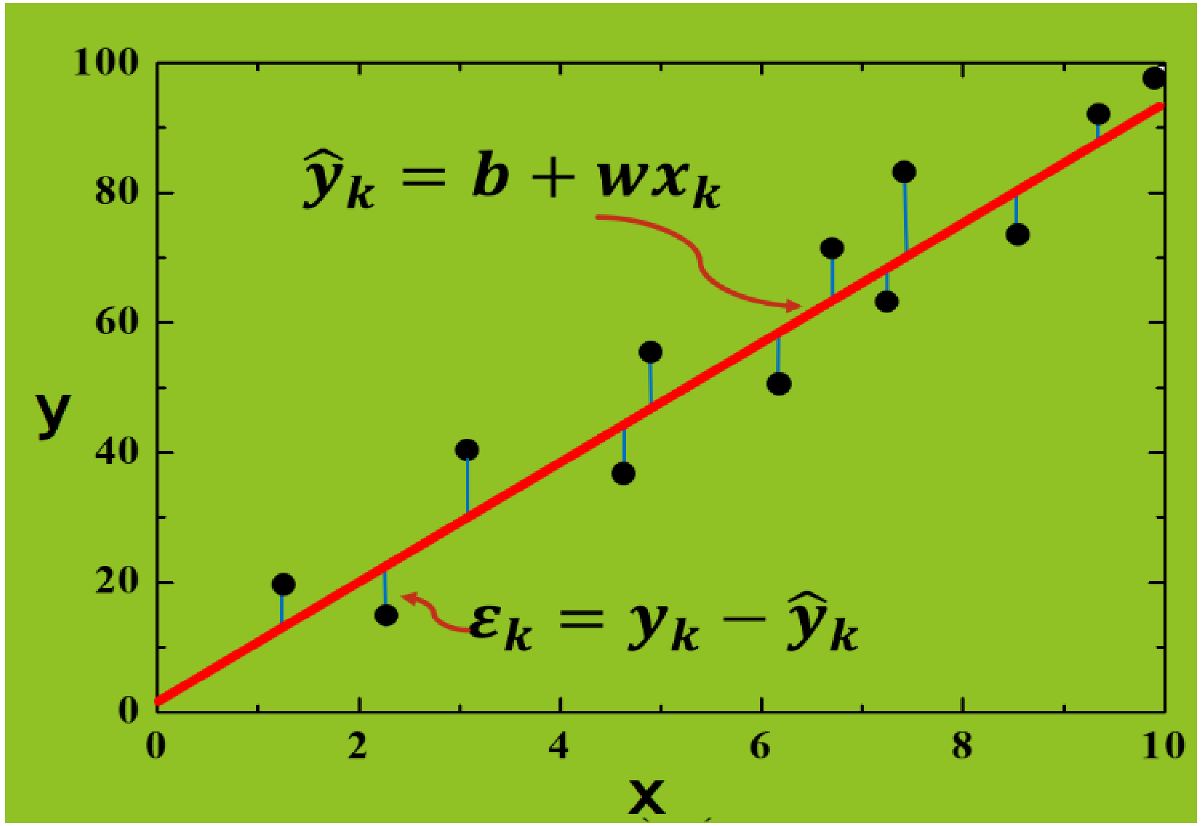
- 利用單一自變數(x)去預測一個依變數(y)，且自變數(x)為一次方。
  - ■ 迴歸模型函數可以設為：預測值$\hat{y}_k$=b+wx_k

  k為樣本編號，若有K個樣本，k = 1、2、3、、、K。

  - ■ 迴歸係數
    - ◆ b 為模型函數和縱軸的截距：

      控制迴歸直線的上下，稱為偏值。
    - ◆ w 為模型函數的直線斜率：

      控制迴歸直線的傾斜，稱為權重。
- ε_k為預測的誤差=(實際值y_k - 預測值$\hat{y}_k$)



損失函數(Loss Funcation)

- 需要對迴歸模型函數好壞建立評估機制，才能在機器學習演算法中，提供修正迴歸係數b、w的依據。

- 最小平方原理：找到最適合的迴歸係數b、w，使預測的誤差平方總和最小，讓模型函數能最符合數據的趨勢。
  註：預測誤差平方和是為了避免正負誤差之間互相抵消。

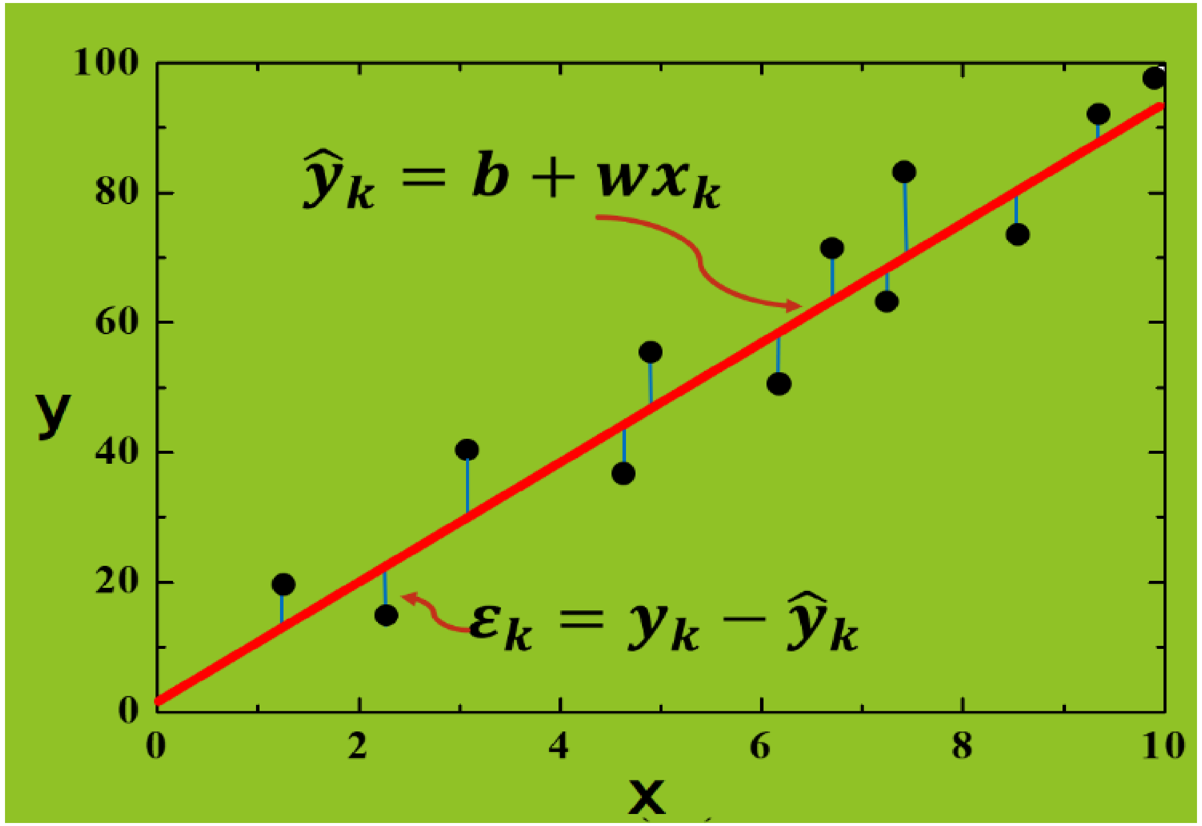
- 將預測的誤差平方總和定義為損失函數(Loss Funcation)：

$$L(b,\mathrm{w}) = \frac{1}{2}\sum_{k=1}^{K}(\varepsilon_k)^2 = \frac{1}{2}\sum_{k=1}^{K}(y_k - \hat{y}_k)^2 = \frac{1}{2}\sum_{k=1}^{K}[y_k - (\mathrm{b} + \mathrm{w}x_k)]^2$$

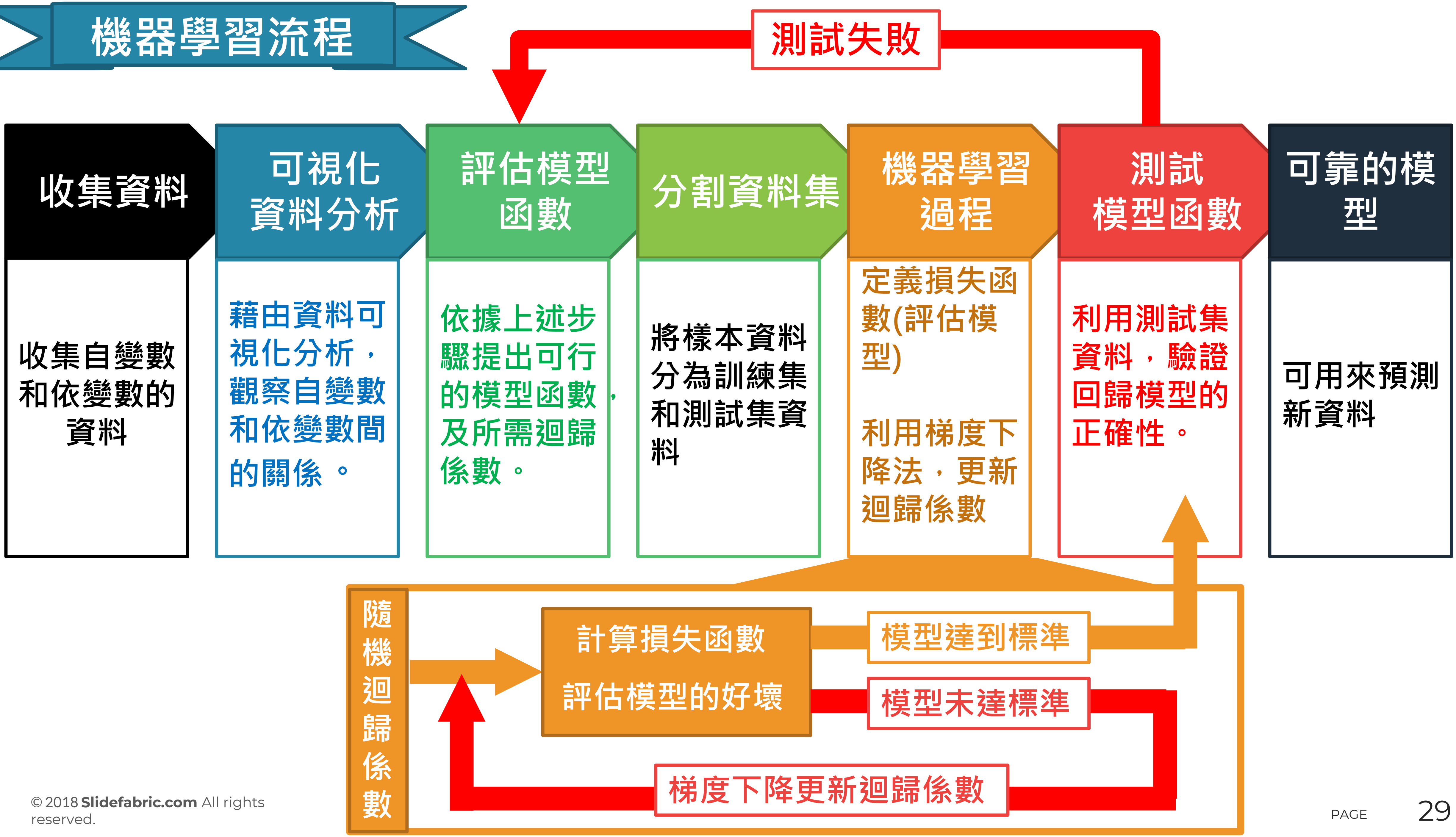
機器學習的處理流程

- 迴歸分析(Regression) 可以說是機器學習入門方法之一，在資料分析科學領域也是常見的統計方法。

- 接下來介紹如何透過機器學習的流程，求出簡單線性迴歸的模型函數，即找出適合的迴歸係數(權重和偏值)。



機器學習流程


測試失敗
收集資料
收集自變數和依變數的資料
可視化資料分析
藉由資料可視化分析，觀察自變數和依變數間的關係。
評估模型函數
依據上述步驟提出可行的模型函數，及所需迴歸係數。
分割資料集
將樣本資料分為訓練集和測試集資料
機器學習過程
定義損失函數(評估模型)
利用梯度下降法，更新迴歸係數
測試模型函數
利用測試集資料，驗證回歸模型的正確性。
可靠的模型
可用來預測新資料
隨機迴歸係數
計算損失函數
評估模型的好壞
模型達到標準
模型未達標準
梯度下降更新迴歸係數

實作1-4　載入資料集並作可視化分析

- 實作目的：將資料可視化，判斷迴歸模型趨勢。
- 實作資料：某廠商為了促銷新開發的產品購買了電視廣告，預算部門想要知道當每天廣告推播次數和當天產品銷售數量之間的關係，下表為統計數日的資料：

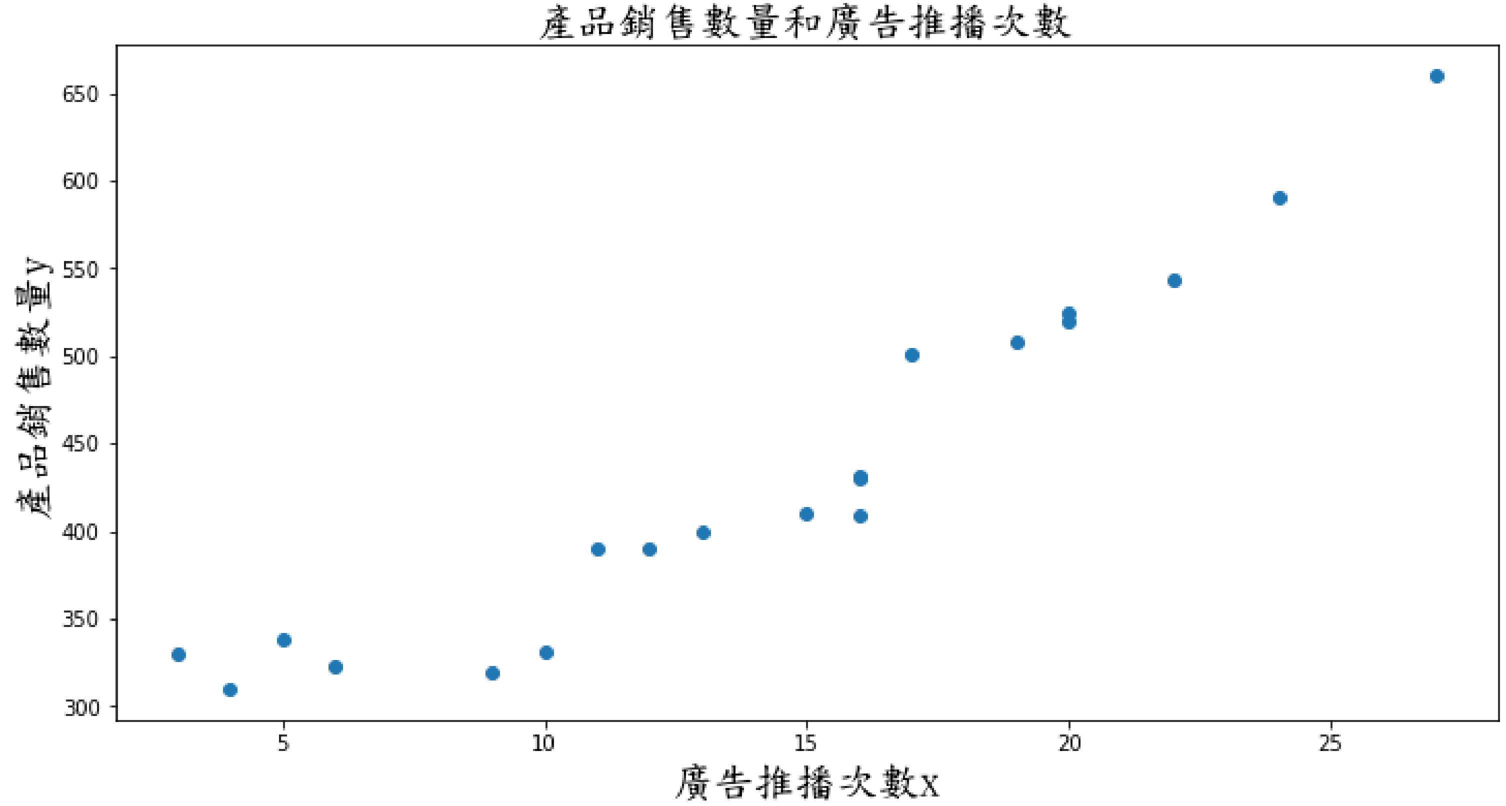
日期	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21
推播次數 x	24	22	15	4	9	20	5	3	17	19	13	10	12	11	16	27	16	16	6	20	24
銷售數量 y	591	543	410	310	319	520	338	330	501	508	399	331	390	390	431	660	409	430	323	524	591

實作1-4 載入資料集並作可視化分析

● 實作要求：

■ 能畫出產品銷售數量和廣告推播次數的關係圖。

■ 能根據畫出的關係圖，判斷可能的迴歸模型函數並寫出函數。



用矩陣來作模型預測

■ 特徵矩陣X：每一個樣本特徵(自變數)為一列所構成的矩陣。

■ 特徵矩陣X經過矩陣運算可以一步算出
所有樣本預測值矩陣$\widehat{Y}$(K列1行)表示：

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_k \end{bmatrix}[w] + [b] = \begin{bmatrix} wx_1 \\ wx_2 \\ wx_3 \\ \vdots \\ wx_k \end{bmatrix} + [b] = \begin{bmatrix} wx_1 + b \\ wx_2 + b \\ wx_3 + b \\ \vdots \\ wx_k + b \end{bmatrix} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_K \end{bmatrix} = \widehat{Y}$$

樣本數

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_k \end{bmatrix}$$

特徵矩陣X

■ [預估值矩陣$\widehat{Y}$] = [特徵矩陣X][權重矩陣W] + [偏值B]

梯度下降法更新迴歸係數

$$\hat{y}_k = \mathrm{b} + \mathrm{w}x_k \qquad \varepsilon_k = y_k - \hat{y}_k \qquad L = \frac{1}{2}\sum_{k=1}^{K}(\varepsilon_k)^2$$

$$\left.\begin{matrix} \mathrm{b} \\ \mathrm{w} \end{matrix}\right] \rightarrow \hat{y}_k \rightarrow \varepsilon_k \rightarrow \mathrm{L}$$

$$\frac{\partial \hat{y}_k}{\partial b} = 1 \text{ 、 } \frac{\partial \hat{y}_k}{\partial w} = x_k \qquad \frac{\partial \varepsilon_k}{\partial \hat{y}_k} = -1 \qquad \frac{\partial L}{\partial \varepsilon_k} = \sum_{k=1}^{K}\varepsilon_k$$

$$\frac{\partial L}{\partial \hat{y}_k} = \frac{\partial L}{\partial \varepsilon_k}\frac{\partial \varepsilon_k}{\partial \hat{y}_k} = \sum_{k=1}^{K}\varepsilon_k\,(-1) = \sum_{k=1}^{K}(\hat{y}_k - y_k) \begin{cases} \dfrac{\partial L}{\partial w} = \dfrac{\partial L}{\partial \hat{y}_k}\dfrac{\partial \hat{y}_k}{\partial w} = \displaystyle\sum_{k=1}^{K}(\hat{y}_k - y_k)x_k \\[2em] \dfrac{\partial L}{\partial b} = \dfrac{\partial L}{\partial \hat{y}_k}\dfrac{\partial \hat{y}_k}{\partial b} = \displaystyle\sum_{k=1}^{K}(\hat{y}_k - y_k) \end{cases}$$

梯度下降法更新迴歸係數

■ 迴歸係數b和w的梯度下降修正方程式為：

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

$$= b - \eta[(\hat{y}_1 - y_1) + (\hat{y}_2 - y_2) + (\hat{y}_3 - y_3) + 、、 + (\hat{y}_K - y_K)]$$

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

$$= w - \eta[(\hat{y}_1 - y_1)x_1 + (\hat{y}_2 - y_2)x_2 + (\hat{y}_3 - y_3)x_3 + 、、 + (\hat{y}_K - y_K)x_K]$$

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：

$[w]$=$[w]$ - $\eta[(\hat{y}_1-y_1)x_1+(\hat{y}_2-y_2)x_2+(\hat{y}_3-y_3)x_3+$ 、 、 $+(\hat{y}_3-y_3)x_k]$

$$= [w] \quad - \eta \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_k \end{bmatrix}^T \times \left(\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_K \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_K \end{bmatrix} \right)$$

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

實作1-5 梯度下降法求簡單線性迴歸的係數

- 承實作五的案例數據，建立產品銷售數量和廣告推播次數之間的簡單線性迴歸模型。
- 能畫出損失函數和學習次數間的關係圖，並指出學習飽和點的次數。
- 修改不同的學習率，比較達到學習滿足點所需的次數會有何變化?
- 畫出迴歸係數的更新路徑圖，想想是否有方法能使其更新次數變少，就能達到學習滿足點。

單元5

特徵縮放
feature scaling

特徵縮放(feature scaling)

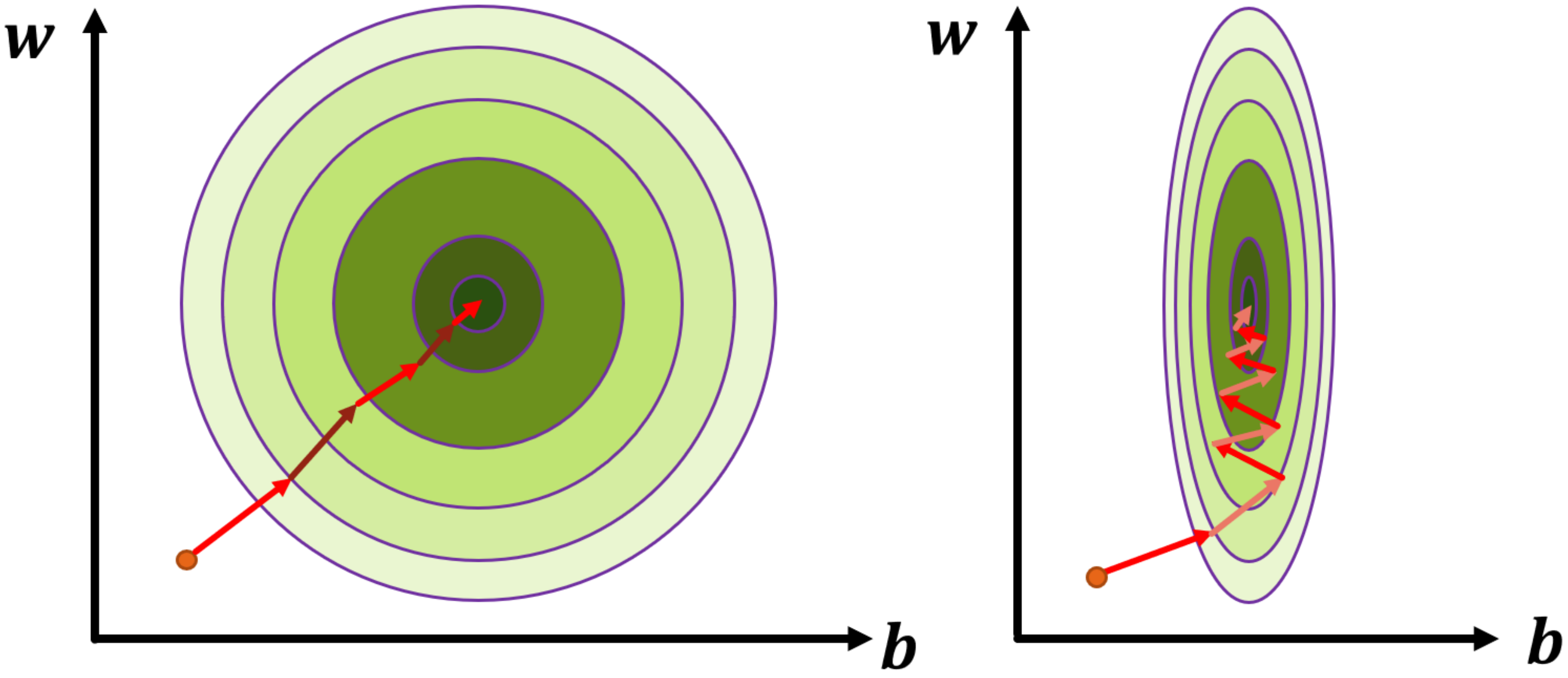
- 觀察梯度下降法b和w的修正方程式為

$$b = b - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k) \qquad\qquad w = w - \eta \sum_{k=1}^{K} (\hat{y}_k - y_k) x_k$$

- 若特徵值(自變數)x_k遠大於1或遠小於1，會造成b和w在梯度下降時修正幅度相差很大。

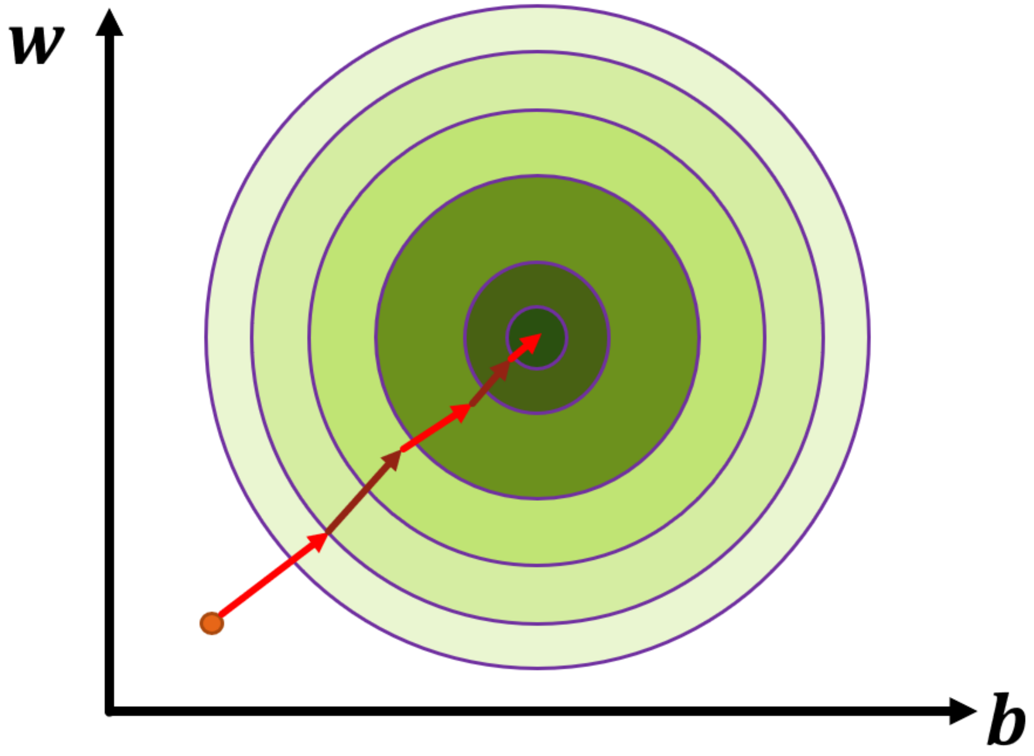
特徵縮放(feature scaling)

若某一個自變數範圍很大、另一個很小，當我們在做梯度下降時，整個等高線圖會呈現橢圓的形狀，因此收斂時沒辦法直接朝圓心(最低點)前進。



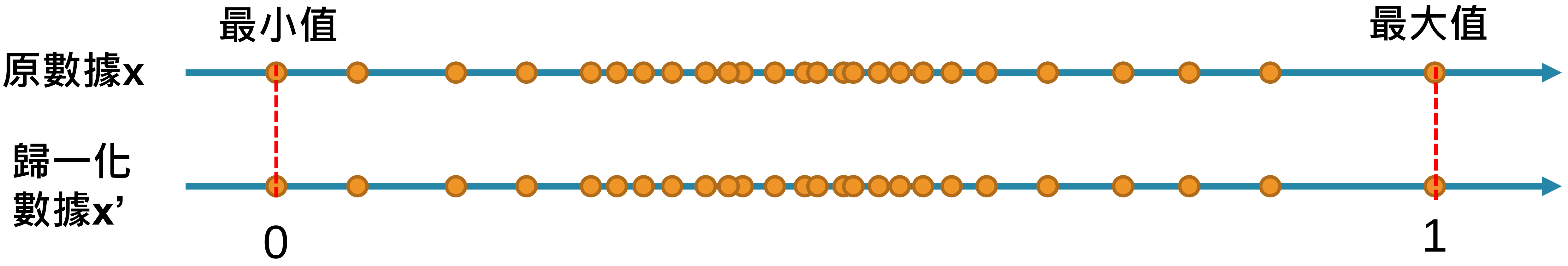
特徵縮放(feature scaling)

- 特徵縮放是將特徵資料按比例縮放，讓資料落在某一特定的區間。
- 用途：去除數據的單位限制，將其轉化為純數值，便於不同單位或量級的特徵能夠進行比較和加權。
- 優點：優化梯度下降法、提高精密度。



特徵縮放(feature scaling)

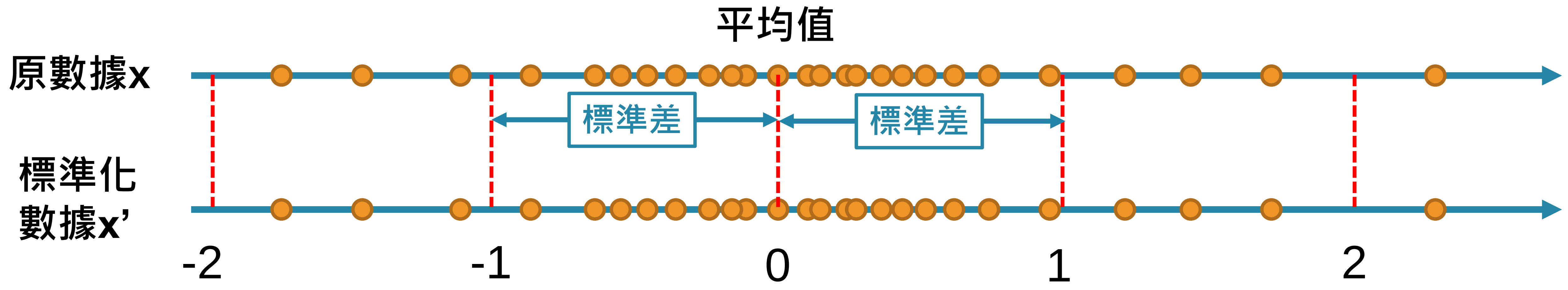
■ 特徵縮放的方法一：
歸一化：將原特徵數據按比例縮放到0到1的區間。



$$\text{縮放後的數據}\mathbf{X'}=\frac{\text{原數據}\mathbf{x}-\text{原數據最小值}}{\text{原數據最大值}-\text{原數據最小值}}$$

特徵縮放(feature scaling)

■ 特徵縮放的方法二：
標準化：會將所有特徵數據縮放成平均為0、標準差為單位。

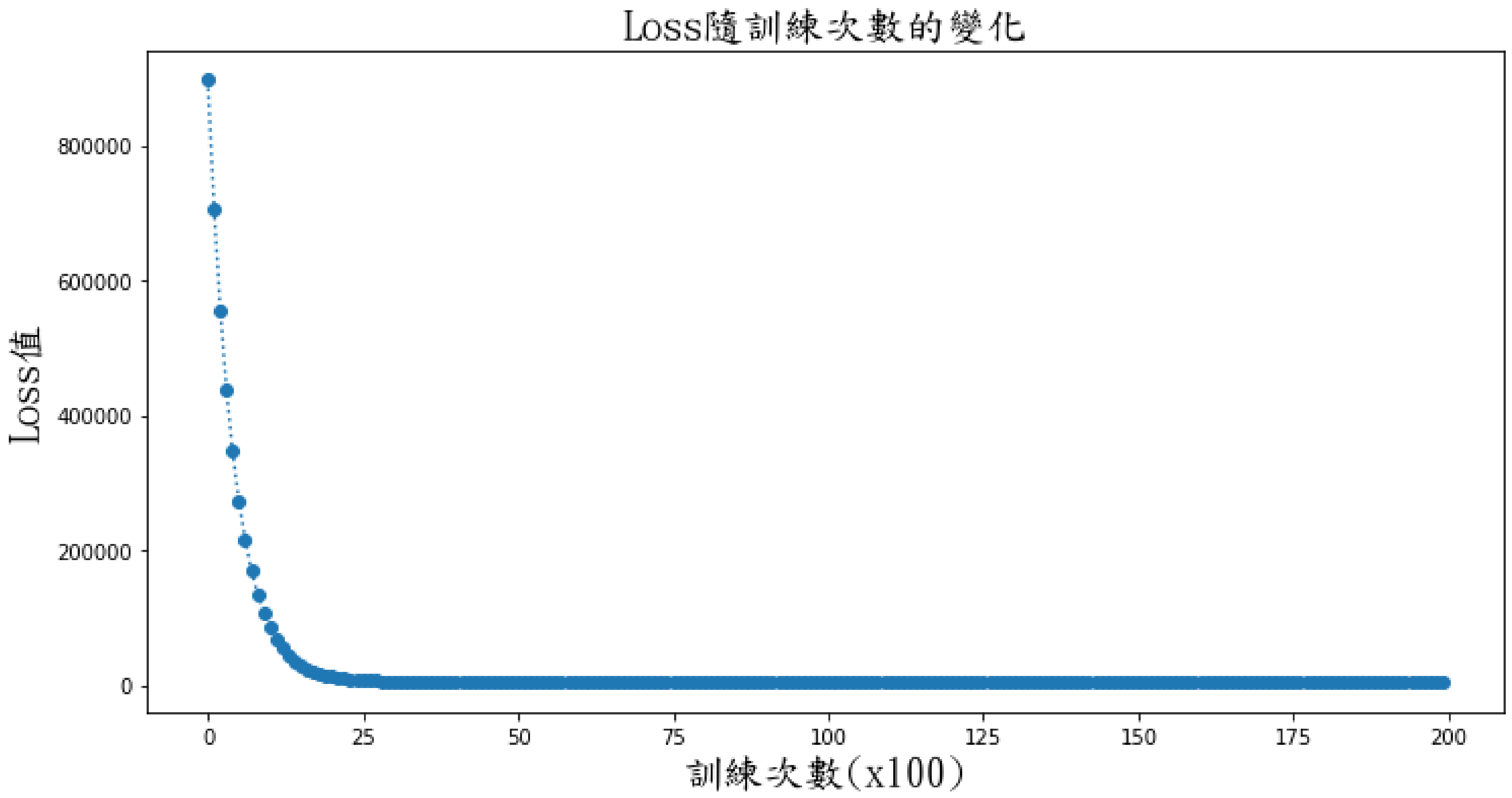


$$縮放後的數據X' = \frac{原數據x - 數據平均值}{原數據標準差}$$

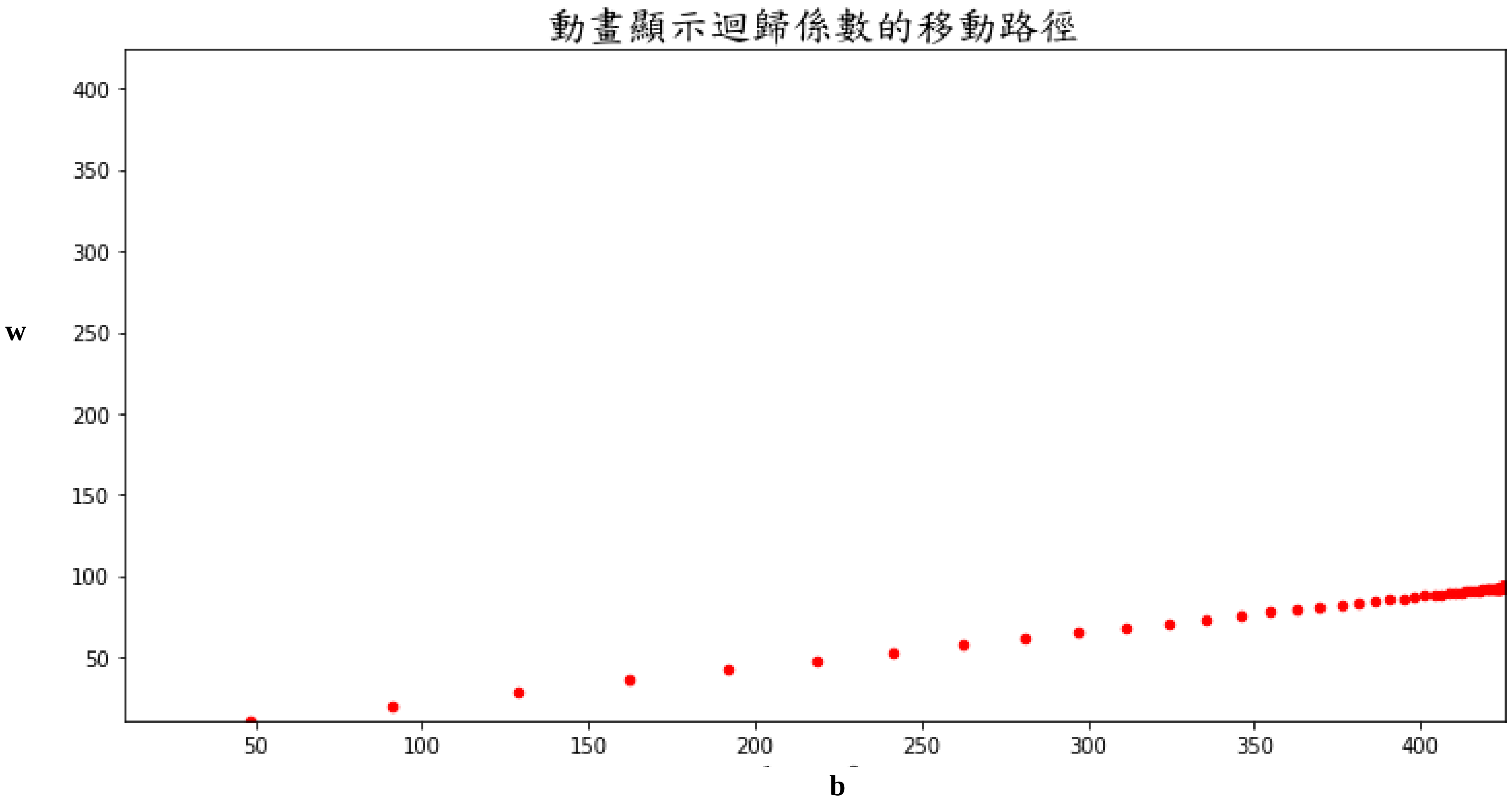
實作1-6 有特徵縮放的簡單線性迴歸模型

- 比較特徵縮放後，模型學習的效率和未作特徵縮放的差別。
- 學會兩種特徵縮放的方法。
- 比較有特徵縮放達到學習滿足點所需的次數會有何變化?
- 比較有特徵縮放迴歸係數的更新路徑圖有何變化?其如何影響達到學習滿足點所需的次數?

實作1-6 有特徵縮放的簡單線性迴歸模型



實作1-6 有特徵縮放的簡單線性迴歸模型



單元6

曲線的線性迴歸

用曲線來擬合數據

- 我們以自變數 x 的 n 次方函數來擬合依變數 y 的關係。

假設共有 K 組的訓練集資料，如下表：

自變數x_k	x_1	x_2	x_3	x_4	x_5	x_6	x_7	、、	x_K
依變數y_k	y_1	y_2	y_3	y_4	y_5	y_6	y_7	、、	y_K

- 多項式迴歸的預測模型式：

$$\hat{y}_k = b + w_1 x_k + w_2 x_k^2 + w_3 x_k^3 + 、、、 + w_n x_k^n$$

- k 為第幾筆資料編號，k=1、2、3、、、K。
- $\hat{y}_k$可視為第 k 筆資料的預測期望值。
- 迴歸係數有 n+1 個(偏值和權重)：b、w_1、w_2、w_3、、、、w_n。

用曲線來擬合數據

■ 特徵矩陣X：每一個樣本特徵(自變數)為一列所構成的矩陣。

$$
\begin{array}{c|c}
 & \text{特徵數} \rightarrow \\
\text{樣本數} \downarrow &
\begin{bmatrix}
x_1^1 & x_1^2 & x_1^3 & \cdots & x_1^n \\
x_2^1 & x_2^2 & x_2^3 & \cdots & x_2^n \\
x_3^1 & x_3^2 & x_3^3 & \cdots & x_3^n \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
x_K^1 & x_K^2 & x_K^3 & \cdots & x_K^n
\end{bmatrix}
\end{array}
$$

特徵矩陣X

■ 特徵矩陣X經過矩陣運算可以一步算出所有樣本預測值矩陣$\widehat{Y}$表示：

$$
\begin{bmatrix}
x_1^1 & x_1^2 & x_1^3 & \cdots & x_1^n \\
x_2^1 & x_2^2 & x_2^3 & \cdots & x_2^n \\
x_3^1 & x_3^2 & x_3^3 & \cdots & x_3^n \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
x_K^1 & x_K^2 & x_K^3 & \cdots & x_K^n
\end{bmatrix}
\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix}
+ [b] =
\begin{bmatrix}
b + w_1 x_1 + w_2 x_1^2 \text{、、} + w_n x_1^n \\
b + w_1 x_2 + w_2 x_2^2 \text{、、} + w_n x_2^n \\
b + w_1 x_3 + w_2 x_3^2 \text{、、} + w_n x_3^n \\
\vdots \\
b + w_1 x_K + w_2 x_K^2 \text{、、} + w_n x_K^n
\end{bmatrix}
=
\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_K \end{bmatrix}
= \widehat{Y}
$$

■ 矩陣運算表示法$\widehat{Y} = XW + B$

損失函數(Loss Funcation)

$$L(b, w_1, w_2、、w_n) = \frac{1}{2}\sum_{k=1}^{K}(\varepsilon_k)^2 = \frac{1}{2}\sum_{k=1}^{K}(y_k - \hat{y}_k)^2$$

b

w_1

w_2

w_n

$$\hat{y}_k = \mathrm{b} + w_1 x_k + w_2 x_k^2、、 + w_n x_k^n$$

$$\varepsilon_k = y_k - \hat{y}_k$$

$$L = \frac{1}{2}\sum_{k=1}^{K}(\varepsilon_k)^2$$

$\hat{y}_k$ → ε_k → L

$$\frac{\partial \hat{y}_k}{\partial b} = 1、\frac{\partial \hat{y}_k}{\partial w_1} = x_k、\frac{\partial \hat{y}_k}{\partial w_2} = x_k^2$$

$$、、、\frac{\partial \hat{y}_k}{\partial w_n} = x_k^n$$

$$\frac{\partial \varepsilon_k}{\partial \hat{y}_k} = -1$$

$$\frac{\partial L}{\partial \varepsilon_k} = \sum_{k=1}^{K}\varepsilon_k$$

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

梯度下降法

■ 迴歸係數b、w_1、w_2、、、w_m的梯度下降修正方程式為：

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

$$\begin{cases} b=\mathrm{b}-\eta\dfrac{\partial L}{\partial b}=\mathrm{b}-\eta\displaystyle\sum_{k=1}^{K}(\hat{y}_k-y_k) \\ w_1=w_1-\eta\dfrac{\partial L}{\partial w_1}=w_1-\eta\displaystyle\sum_{k=1}^{K}(\hat{y}_k-y_k)x_k \\ w_2=w_2-\eta\dfrac{\partial L}{\partial w_2}=w_2-\eta\displaystyle\sum_{k=1}^{K}(\hat{y}_k-y_k)x_k^2 \\ 、、、、、 \\ w_n=w_n-\eta\dfrac{\partial L}{\partial w_n}=w_n-\eta\displaystyle\sum_{k=1}^{K}(\hat{y}_k-y_k)x_k^n \end{cases}$$

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：

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} - \eta \begin{bmatrix} (\hat{y}_1 - y_1)x_1^1 + (\hat{y}_2 - y_2)x_2^1 + (\hat{y}_3 - y_3)x_3^1 + 、、 + (\hat{y}_K - y_K)x_K^1 \\ (\hat{y}_1 - y_1)x_1^2 + (\hat{y}_2 - y_2)x_2^2 + (\hat{y}_3 - y_3)x_3^2 + 、、 + (\hat{y}_K - y_K)x_K^2 \\ \vdots \\ (\hat{y}_1 - y_1)x_1^n + (\hat{y}_2 - y_2)x_2^n + (\hat{y}_3 - y_3)x_3^n + 、、 + (\hat{y}_K - y_K)x_K^n \end{bmatrix}$$

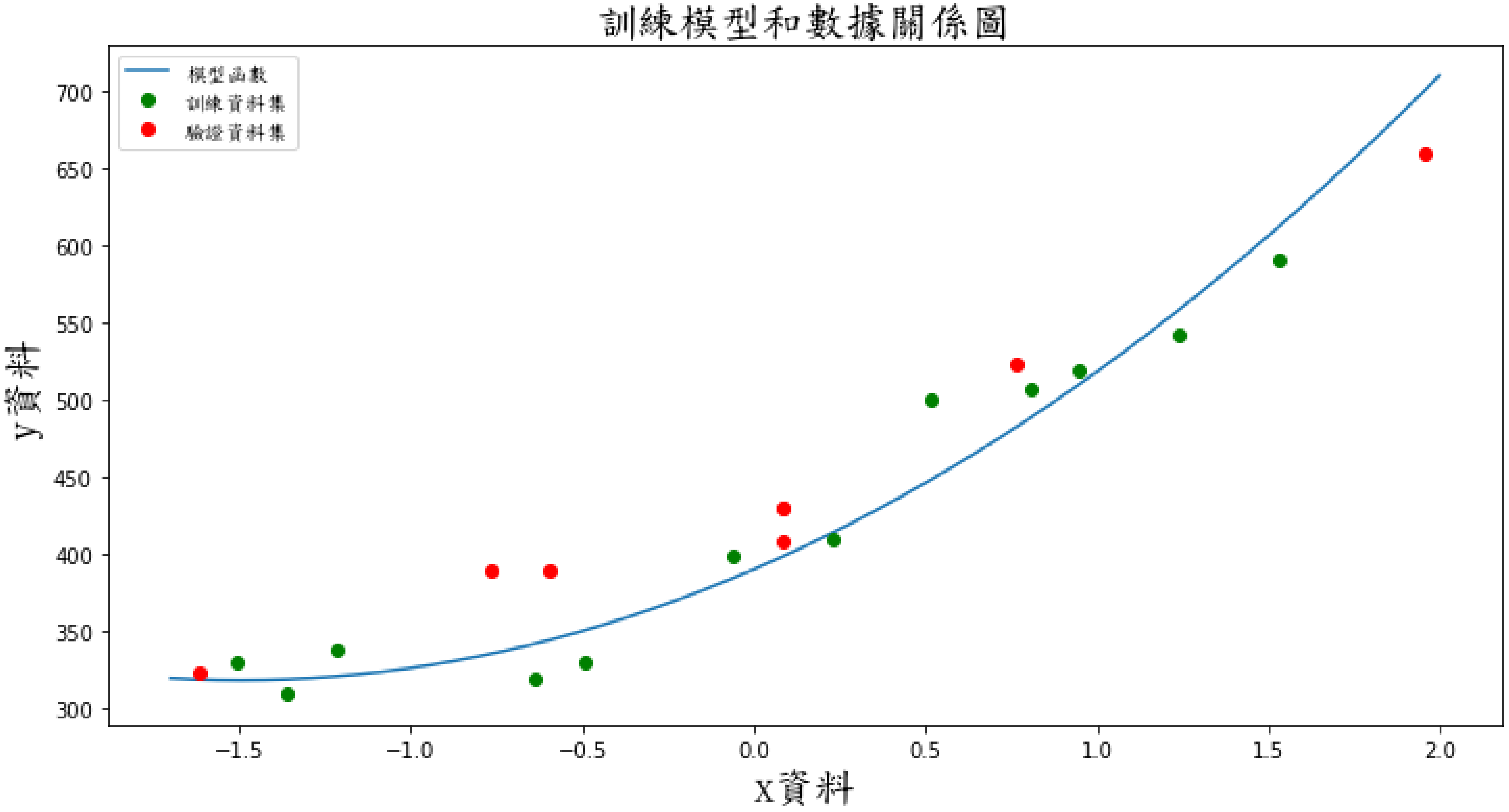
$$= \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} - \eta \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 & \cdots & x_1^n \\ x_2^1 & x_2^2 & x_2^3 & \cdots & x_2^n \\ x_3^1 & x_3^2 & x_3^3 & \cdots & x_3^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_K^1 & x_K^2 & x_K^3 & \cdots & x_K^n \end{bmatrix}^T \times \left(\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_K \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_K \end{bmatrix} \right)$$

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

實作1-7 曲線的線性迴歸

- 實驗目的：將迴歸模型假設為自變數的 2、3、4、、、次方多項式。
- 觀察重點：在相同訓練次數下
  - ■比較簡單線性迴歸和非線性迴歸的損失函數大小。
  - ■比較簡單線性迴歸和非線性迴歸的損失函數的下降趨勢。

實作1-7 曲線的線性迴歸



過度擬合
Over fitting

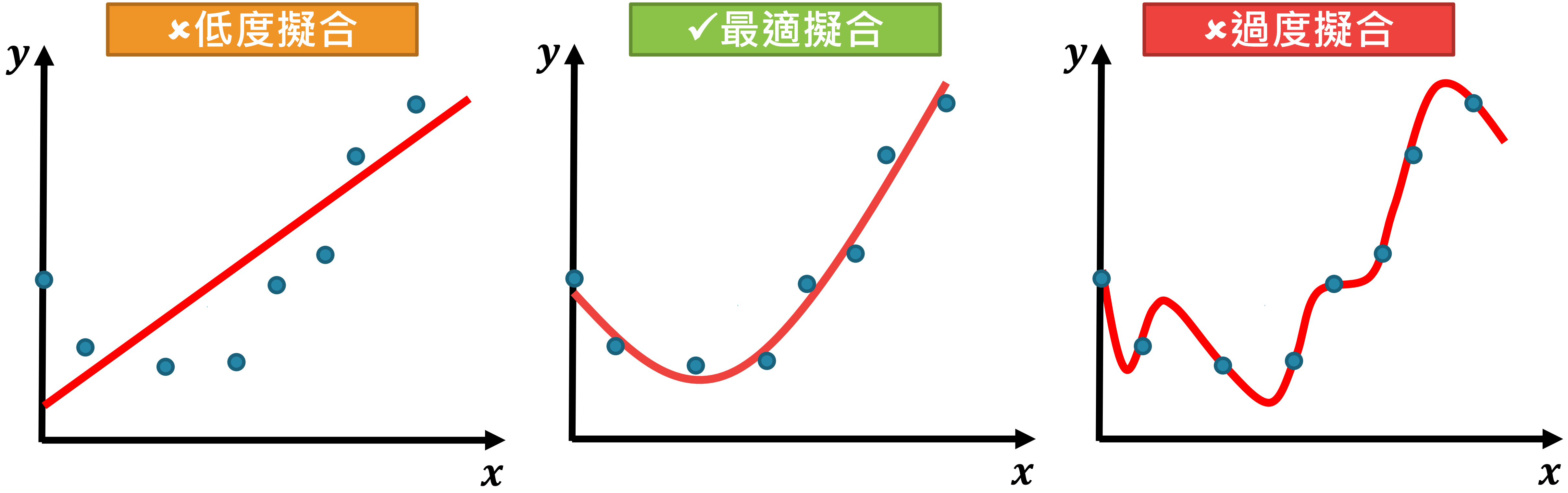
單元7

過度擬合Over fitting

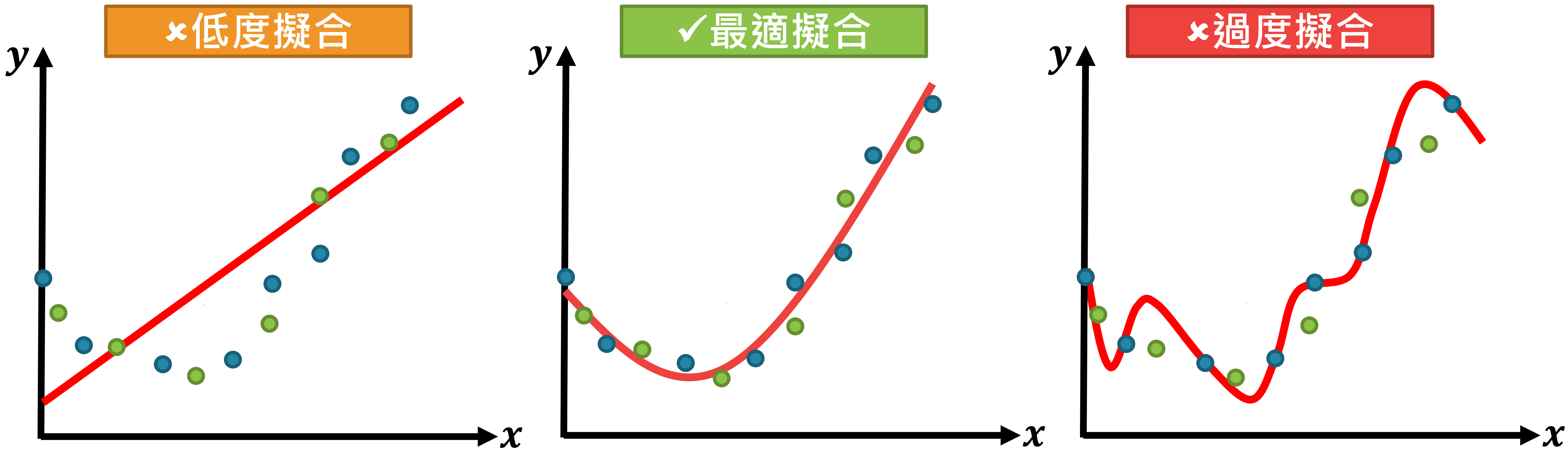
多項式函數的特性是次方越高，轉折點(微分為零的位置)越多，越容易擬合訓練數據的分佈，損失函數會變很小。

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



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。

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。

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