深度學習
Deep Learning

神經網路與梯度下降

Instructor: 林英嘉 (Ying-Jia Lin)
2025/09/10

Course GitHub

Slido # DL0910

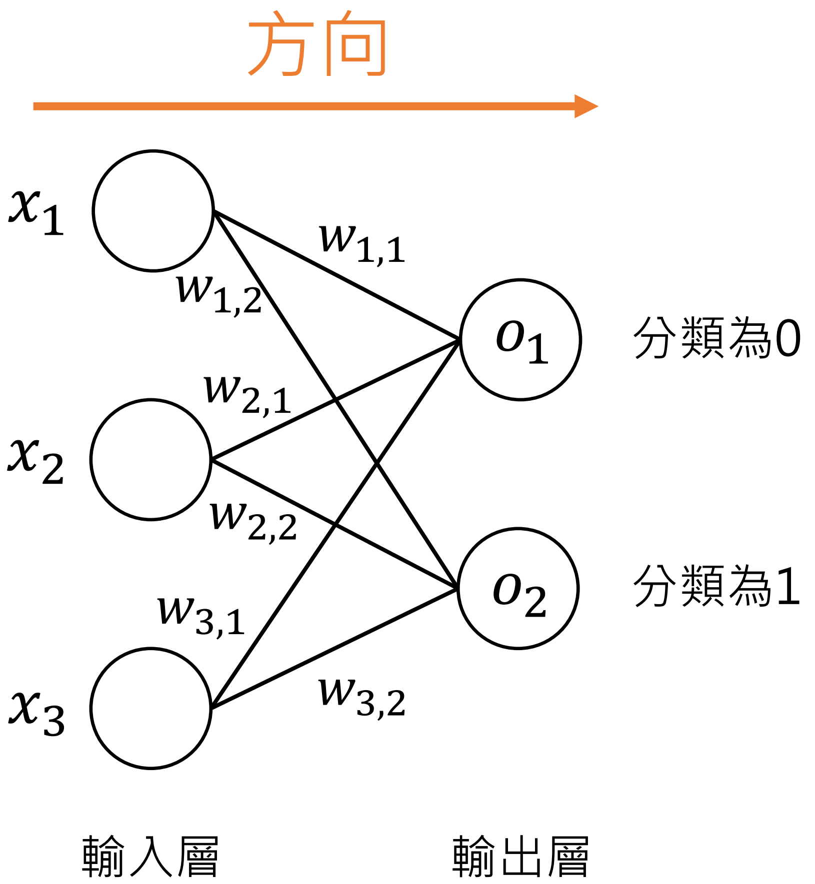
Outline

- Multi-layer perceptron (MLP)
  - 基礎神經網路
- 權重值如何被更新 (梯度下降)

[Recap] 什麼是深度學習?

- 深度學習是一種機器學習方法，著重於使用多層的類神經網路 (Neural Networks) 來完成任務，例如分類、生成等等。
- 在深度學習模型中，通常參數數量都非常龐大，這主要來自於隱藏層大小 (hidden size) 與模型層數 (number of hidden layers) 的增加。

[Recap] 簡單的類神經網路：Perceptron (感知機)



(1st 層) ⟶ 只有經過一層權重值計算

x 代表輸入

w 代表權重值

代表x有多少比例要被傳至下一層

今日重點

$$o_1 = x_1 w_{1,1} + x_2 w_{2,1} + x_3 w_{3,1}$$

Introduction to Matrix (矩陣)

- 矩陣：矩形**陣列**
- M 個 rows (列) x n 個 columns (行)
  - (程式中) 矩陣形狀為：(m, n)
  - 矩陣形狀又稱 shape, size

$$\begin{array}{cc} & \textbf{columns} \\ \textbf{rows} & \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{bmatrix} \end{array}$$

Matrix Multiplication

- 假設有兩矩陣A和B，形狀分別為 (m, n) 跟 (n,p)，

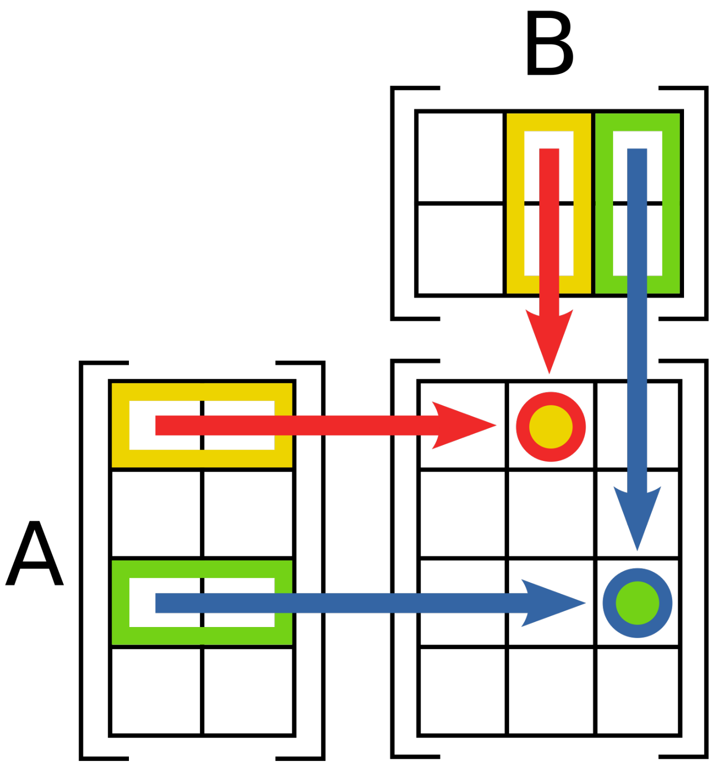
$$\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{bmatrix} \quad \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,p} \\ b_{2,1} & b_{2,2} & \dots & b_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{\mathrm{n},1} & b_{\mathrm{n},2} & \dots & b_{\mathrm{n},p} \end{bmatrix}$$

則 AB = $a_{1,1} \times b_{1,1} + a_{1,2} \times b_{2,1} + \cdots + a_{1,n} \times b_{\mathrm{n},1} +$

$a_{2,1} \times b_{1,2} + a_{2,2} \times b_{2,2} + \cdots + a_{2,n} \times b_{\mathrm{n},2} + \cdots$

$+ a_{\mathrm{m},1} \times b_{1,p} + a_{\mathrm{m},2} \times b_{2,p} + \cdots + a_{\mathrm{m},n} \times b_{\mathrm{n},p}$

Matrix Multiplication

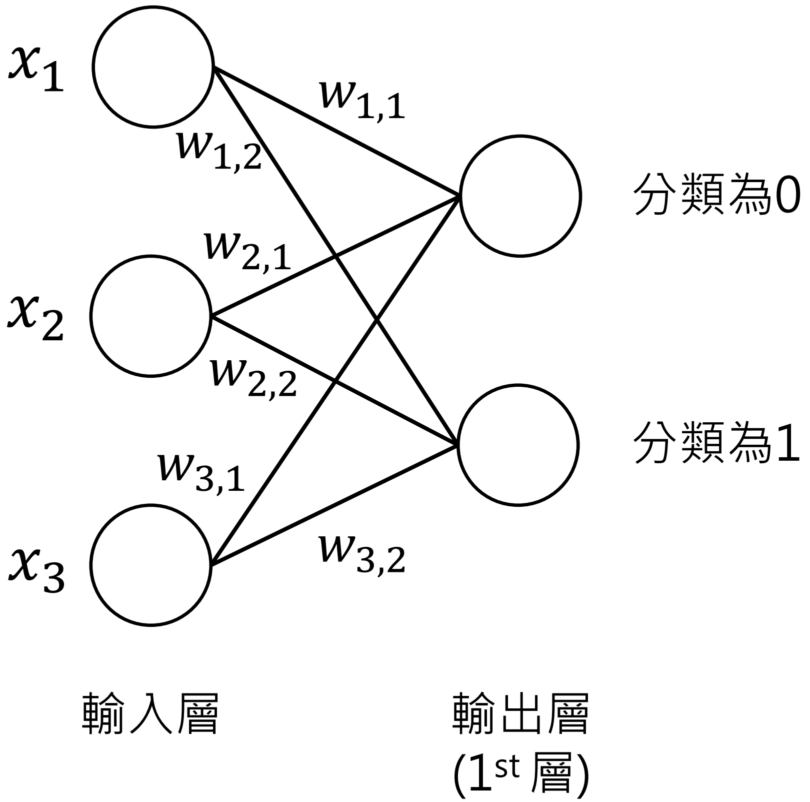


Example:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 2 & 1 \end{bmatrix} = \begin{bmatrix} 12 & 15 \end{bmatrix}$$

https://zh.wikipedia.org/zh-tw/矩陣乘法
#/media/File:Matrix_multiplication_diagram.svg

[Recap] 類神經網路：Perceptron (感知機)



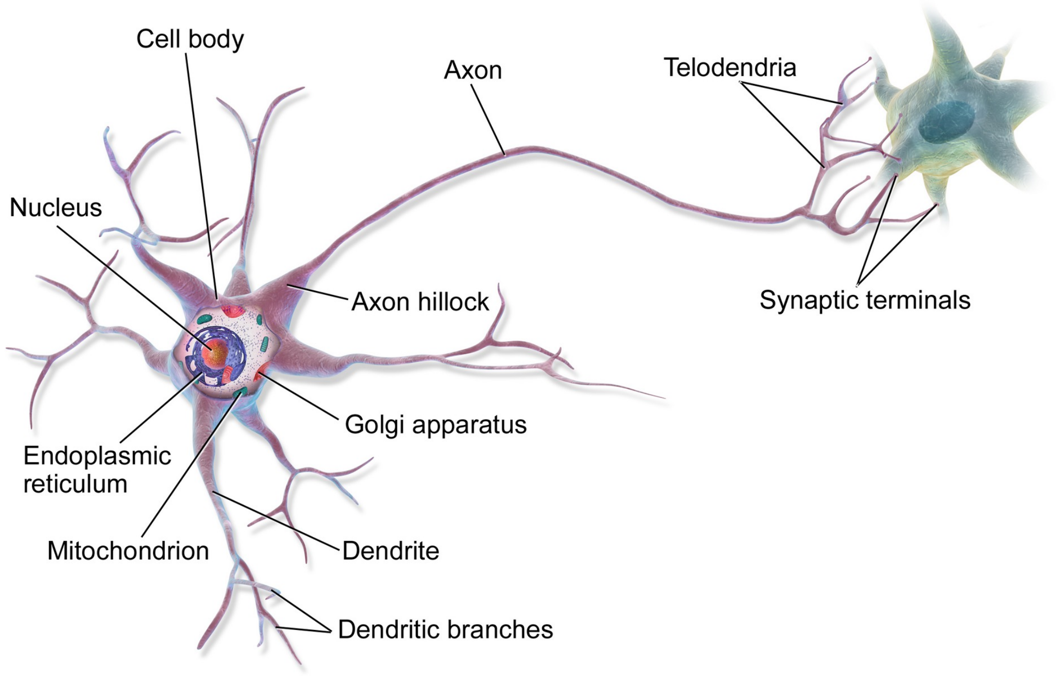


Figure source: https://zh-yue.wikipedia.org/wiki/神經細胞

模型 (網路) 的輸入是什麼？

數值型資料

$$[x_1 \quad x_2 \quad x_3] \longrightarrow \text{某一筆資料}$$

例如 血壓 心跳 BMI

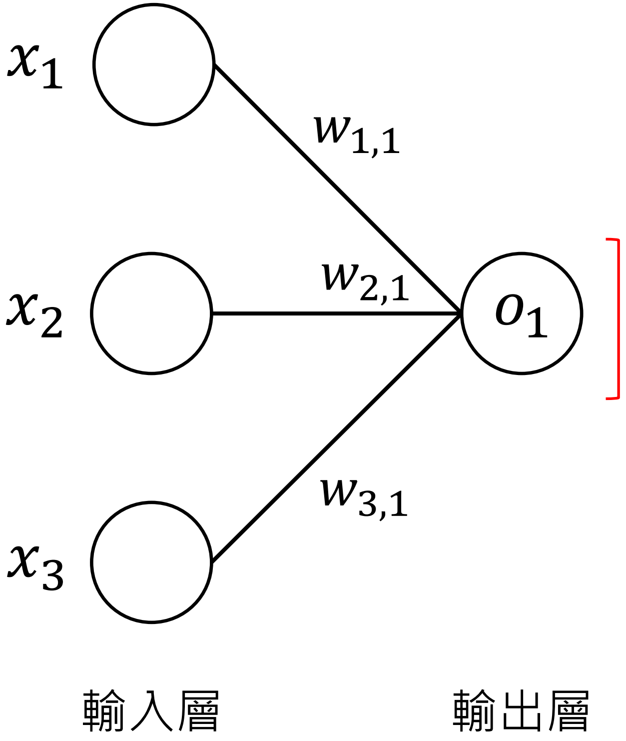
影像資料

影像像素值 (pixel)

$$\longrightarrow \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

Math for Perceptron using a toy example (1)

output_size = 1



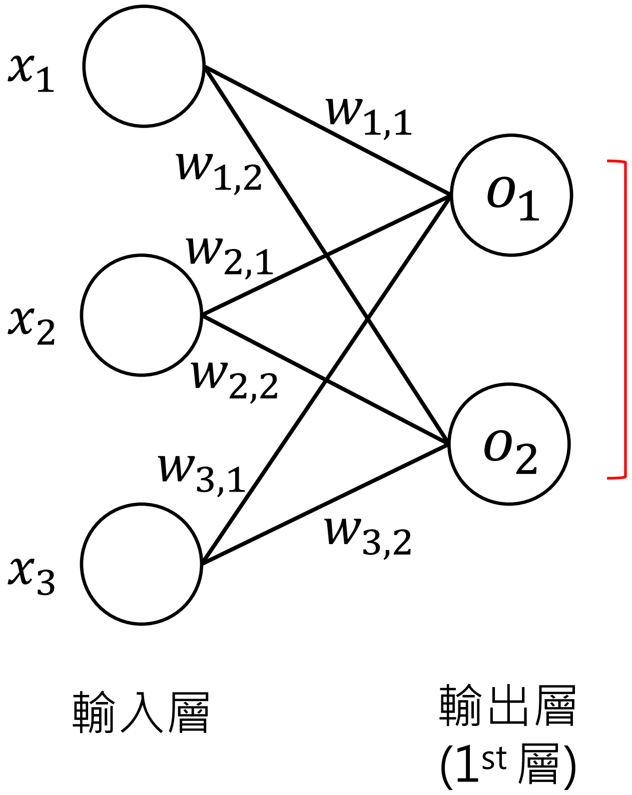
$$[x_1 \quad x_2 \quad x_3] \times \begin{bmatrix} w_{1,1} \\ w_{2,1} \\ w_{3,1} \end{bmatrix} + [b_1] = [o_1]$$

$\mathbb{R}^{1\times3}$ $\mathbb{R}^{3\times1}$ $\mathbb{R}^{1}$ $\mathbb{R}^{1}$

bias

Math for Perceptron using a toy example (2)

output_size = 2

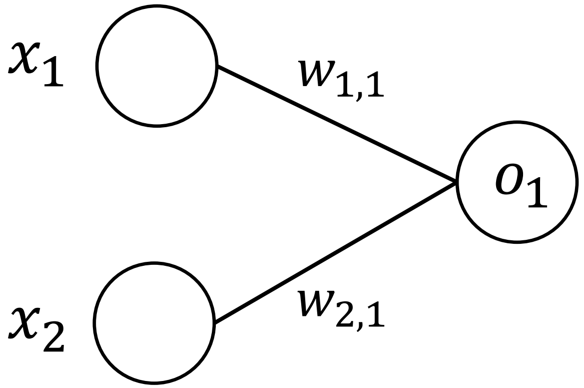


$$[x_1 \quad x_2 \quad x_3] \times \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \\ w_{3,1} & w_{3,2} \end{bmatrix} + [b_1 \quad b_2] = [o_1 \quad o_2]$$

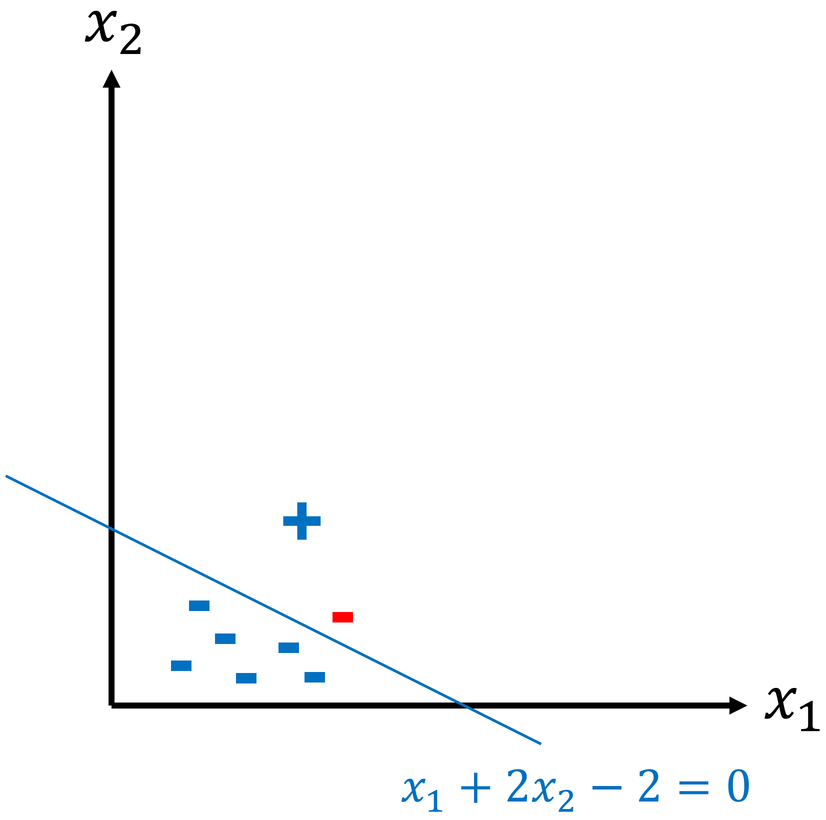
$\mathbb{R}^{1\times3}$ $\mathbb{R}^{3\times2}$ $\mathbb{R}^{2}$ $\mathbb{R}^{1\times2}$

bias

實驗結果以外的理論：畫線 (以2D為例)

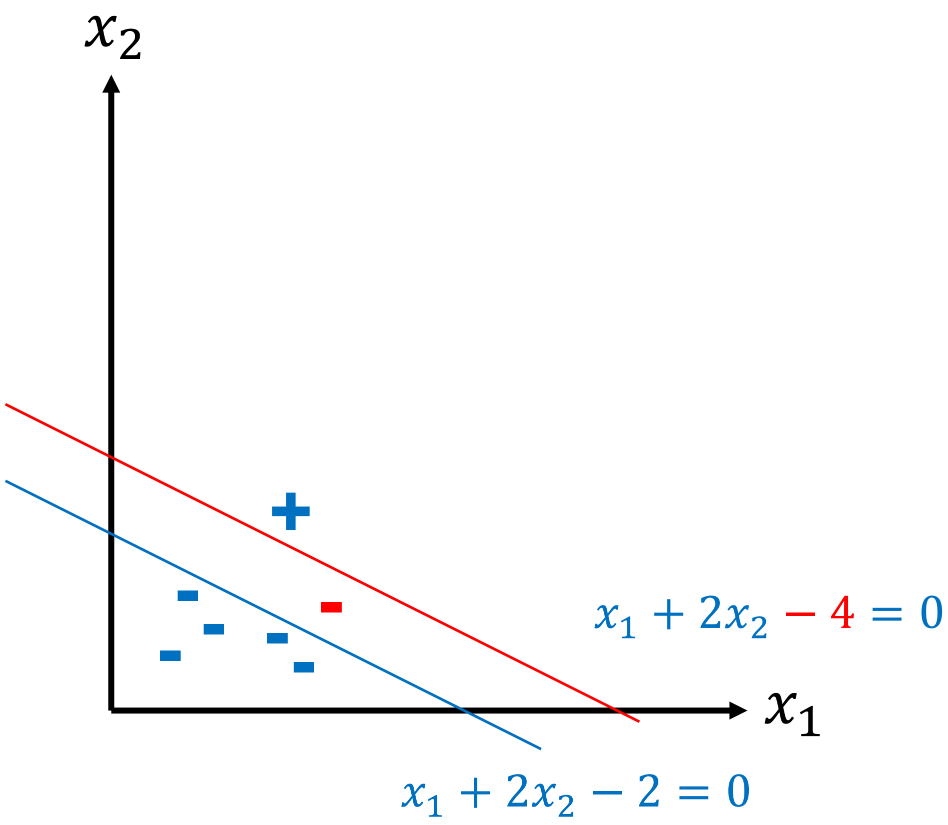


$$o_1 = x_1 w_{1,1} + x_2 w_{2,1}$$



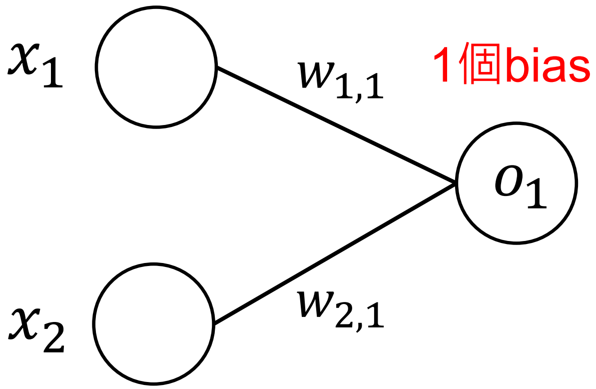
bias 的定義與說明

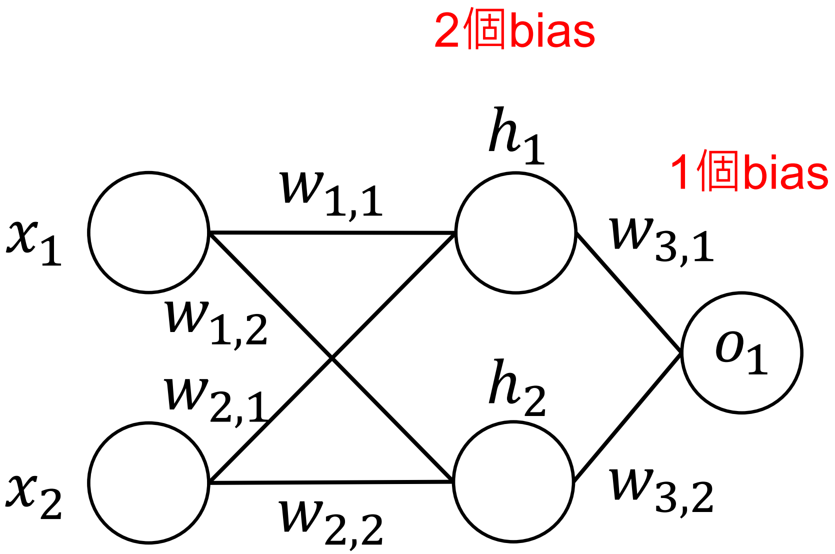
- bias (term): 偏置項
- 偏差值幫助將學到的函數平移
- bias 也是可調整的參數
  - 因此，bias 通常能使分類效果更好



bias 數量

- bias 數量等同於該層的輸出維度，因為每個神經元都需要一個偏移量，來調整自己的**輸出值**
  - hidden layer 就是 hidden size
  - output layer 就是 output size



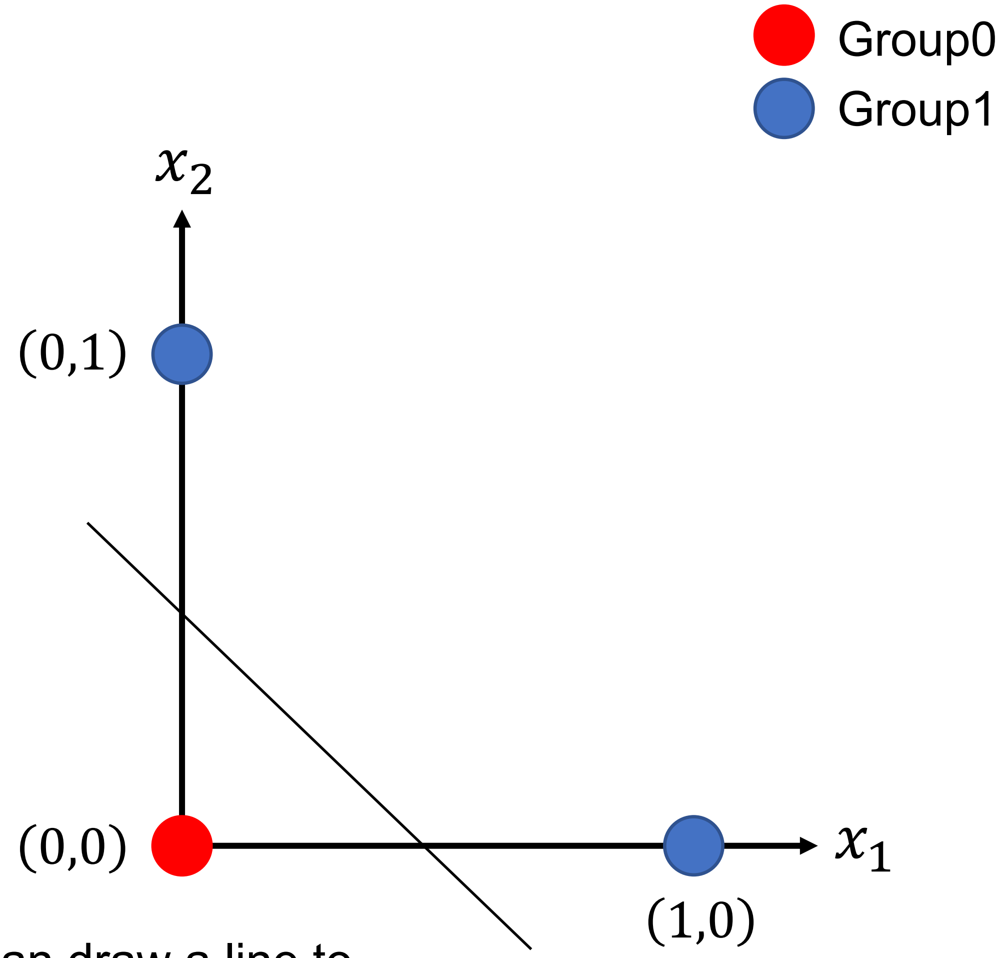


Perceptron 小結

- 權重對於輸入的矩陣乘法的過程相當於進行**線性**轉換
- 全連接 (fully-connected)
- bias 數量等同於該層的輸出維度

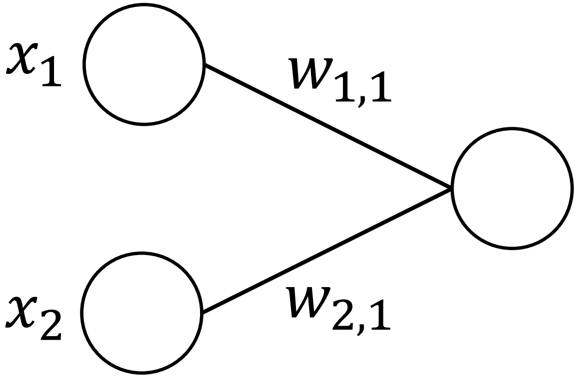
A toy example for classification

x1	x2	Output
0	0	0
0	1	1
1	0	1



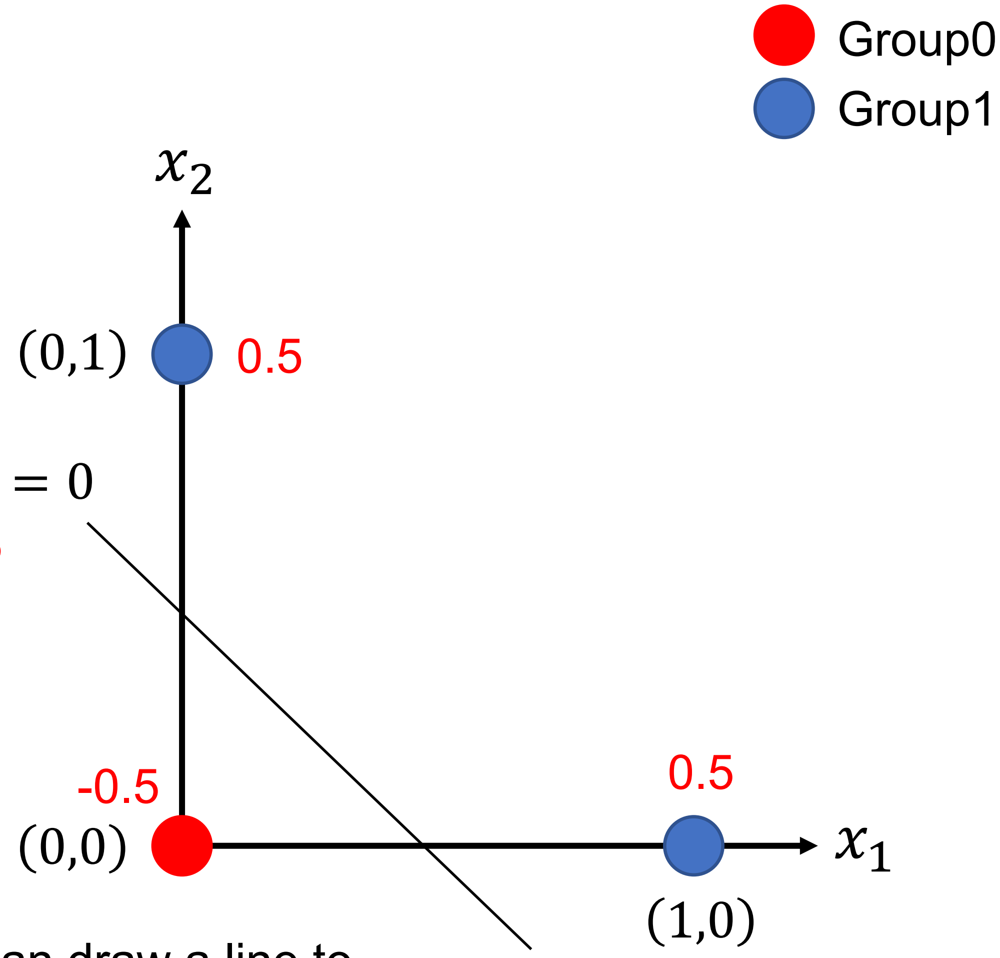
We can draw a line to classify two groups!

A toy example for classification with Perceptron



$$w_{1,1}x_1 + w_{2,1}x_2 + b_1 = 0$$

1 1 -0.5



$$[x_1 \quad x_2] \times \begin{bmatrix} w_{1,1} \\ w_{2,1} \end{bmatrix} + [b_1]$$

We can draw a line to
classify two groups!

XOR Problem

x1	x2	AND
0	0	0
0	1	0
1	0	0
1	1	1

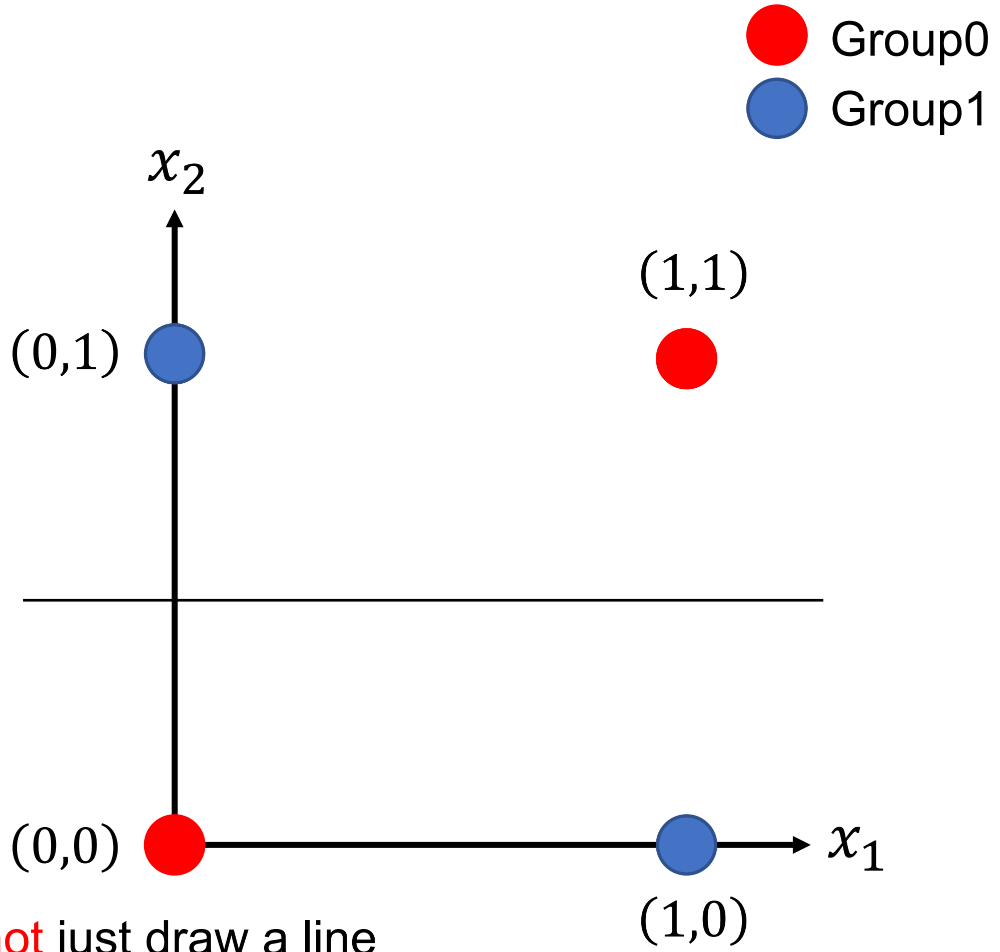
x1	x2	OR
0	0	0
0	1	1
1	0	1
1	1	1

x1	x2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

XOR Problem

XOR: Exclusive-or (邏輯算符互斥或)

x1	x2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

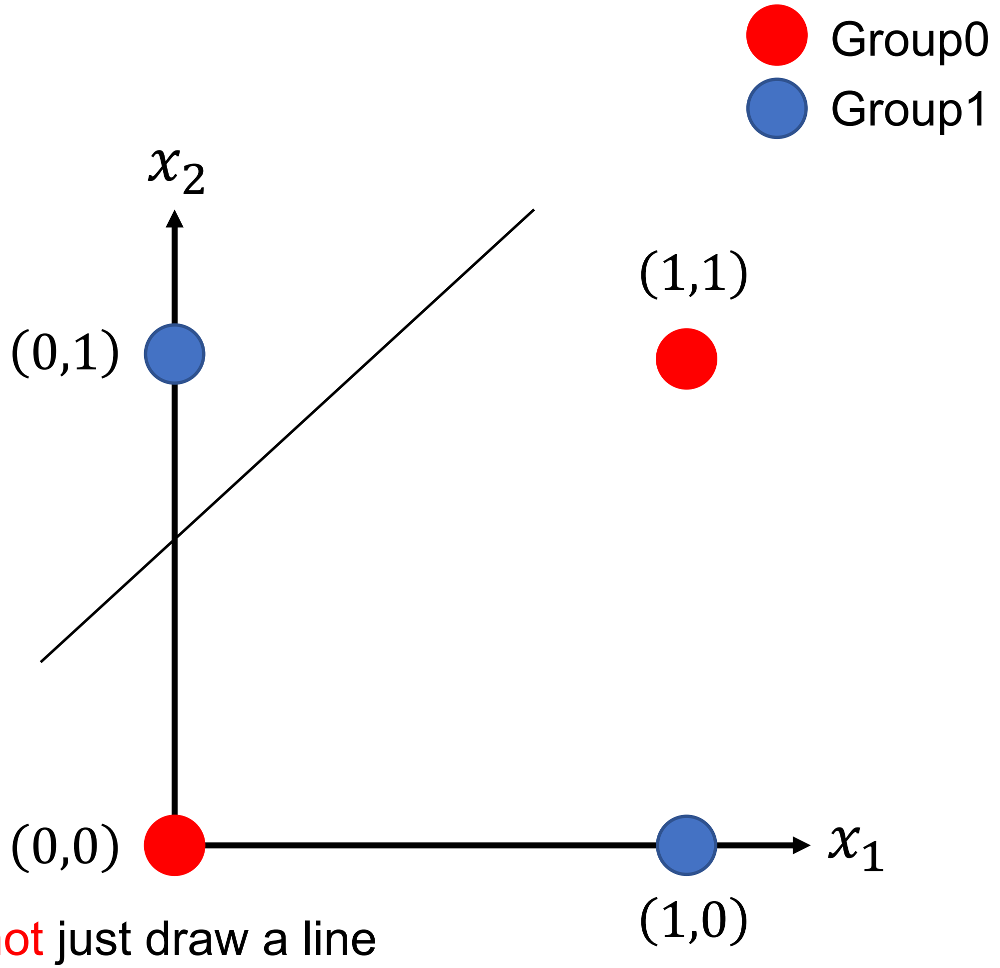


We cannot just draw a line to classify two groups!

XOR Problem

XOR: Exclusive-or (邏輯算符互斥或)

x1	x2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

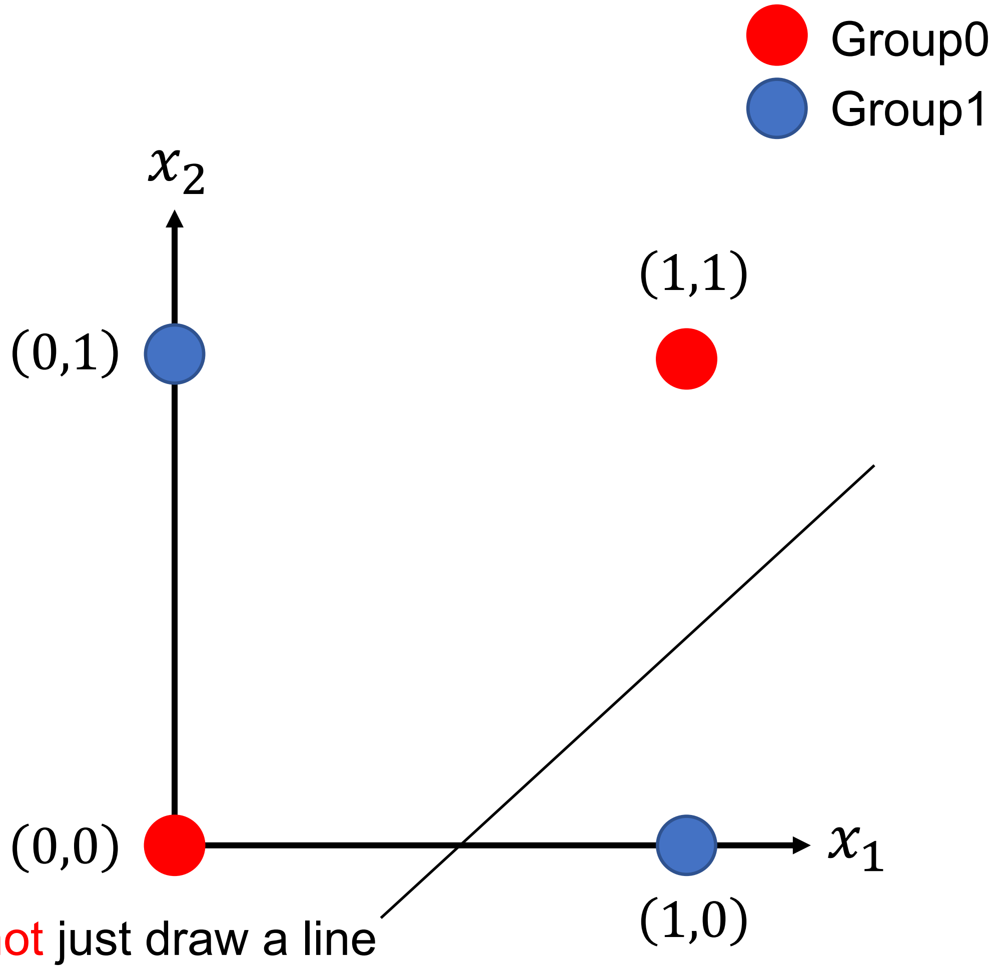


We cannot just draw a line to classify two groups!

XOR Problem

XOR: Exclusive-or (邏輯算符互斥或)

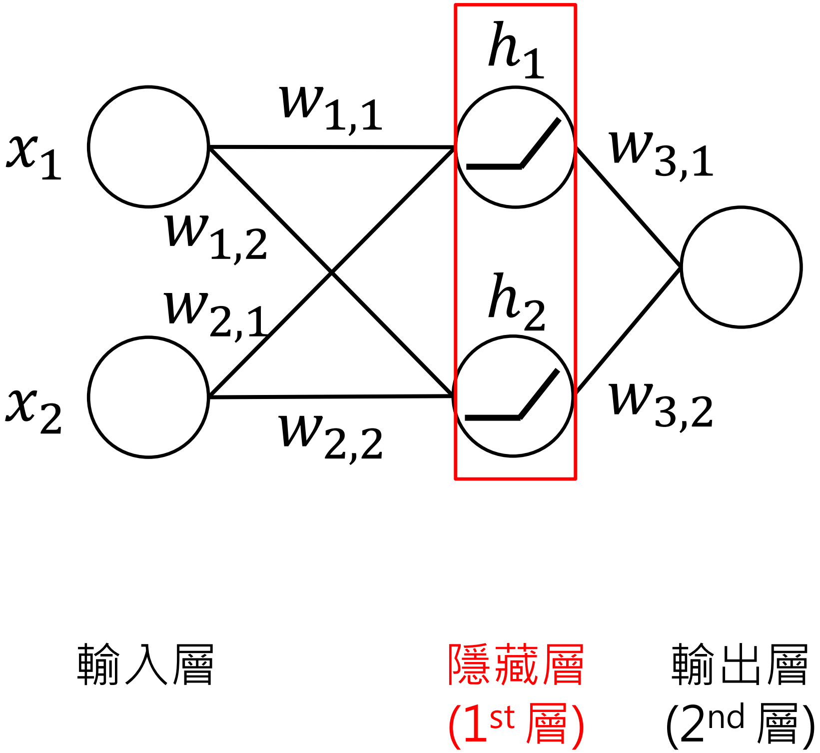
x1	x2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

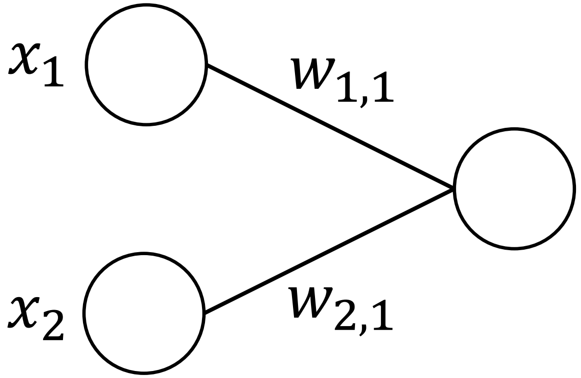


We cannot just draw a line to classify two groups!

MLP for the XOR Problem

ReLU (Rectified Linear Unit)





Perceptron通常沒有隱藏層

ReLU (Rectified Linear Unit)[1][2]

- Non-linear transformation
- Negative values will be transformed to zeros.
- Positive values remain their original ones.

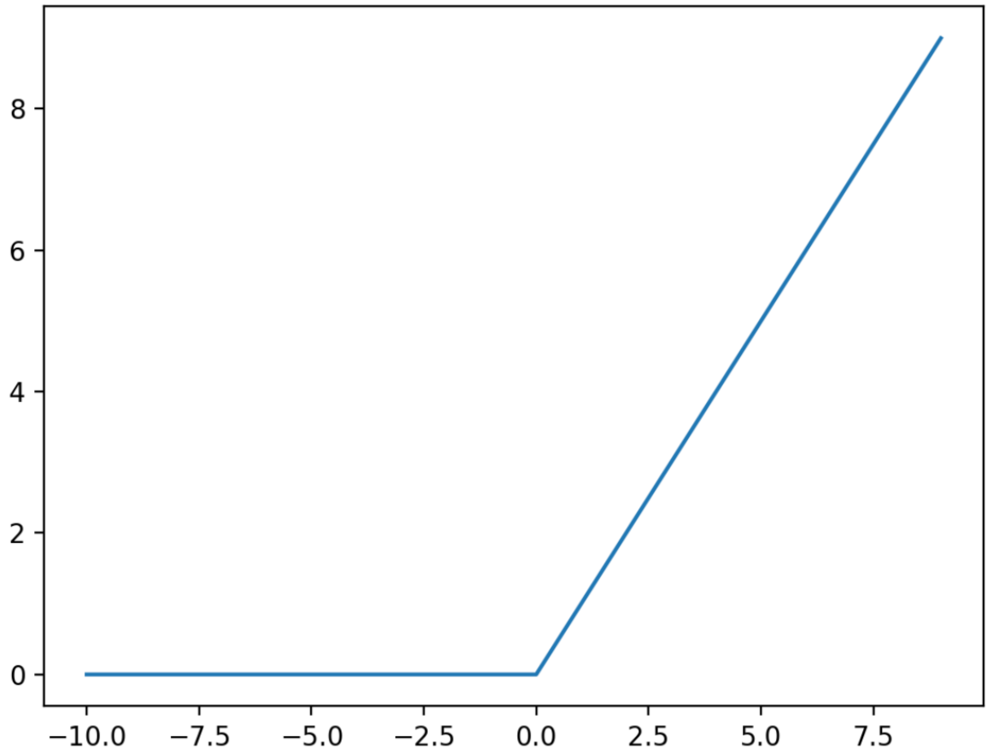


Figure source:
https://paperswithcode.com/method/relu

[1] Jarrett, Kevin, et al. "What is the best multi-stage architecture for object recognition?." ICCV 2009.
[2] Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve restricted boltzmann machines." ICML 2010.

為什麼 ReLU 是非線性函數？

線性函數需要有疊加性：f(x+y) = f(x) + f(y) 與齊次性：f(ax) = a*f(x)

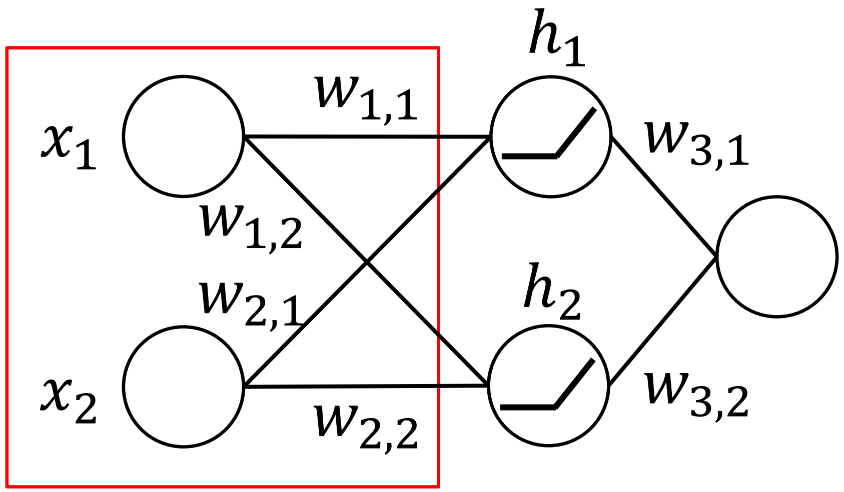
假設 ReLU 為 f :

左式：f(1+(-1)) = f(0) = 0

右式：f(1) + f(-1) = 1 + 0 = 1

左式與右式不符合，故 ReLU 為非線性。

Math of MLP for the XOR Problem (1)



x1	x2
0	0
0	1
1	0
1	1

任一筆資料

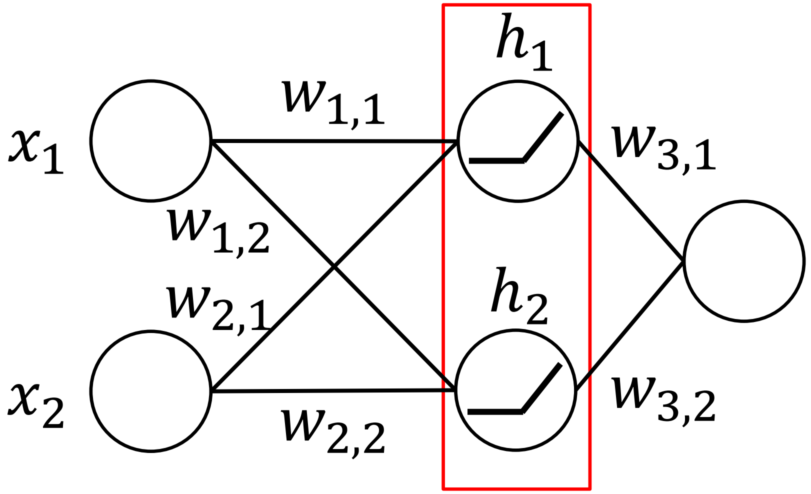
$$[x_1 \quad x_2] \times \begin{bmatrix} \overset{1}{w_{1,1}} & \overset{1}{w_{1,2}} \\ \underset{1}{w_{2,1}} & \underset{1}{w_{2,2}} \end{bmatrix} + [\overset{0}{b_1} \quad \overset{-1}{b_2}]$$

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ 0 & -1 \\ 0 & -1 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

Element-wise summation with bias

Math of MLP for the XOR Problem (2)



$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \xrightarrow{\text{ReLU}} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

ReLU (Rectified Linear Unit)

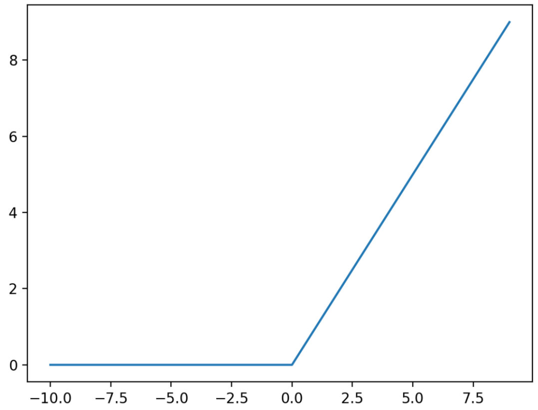


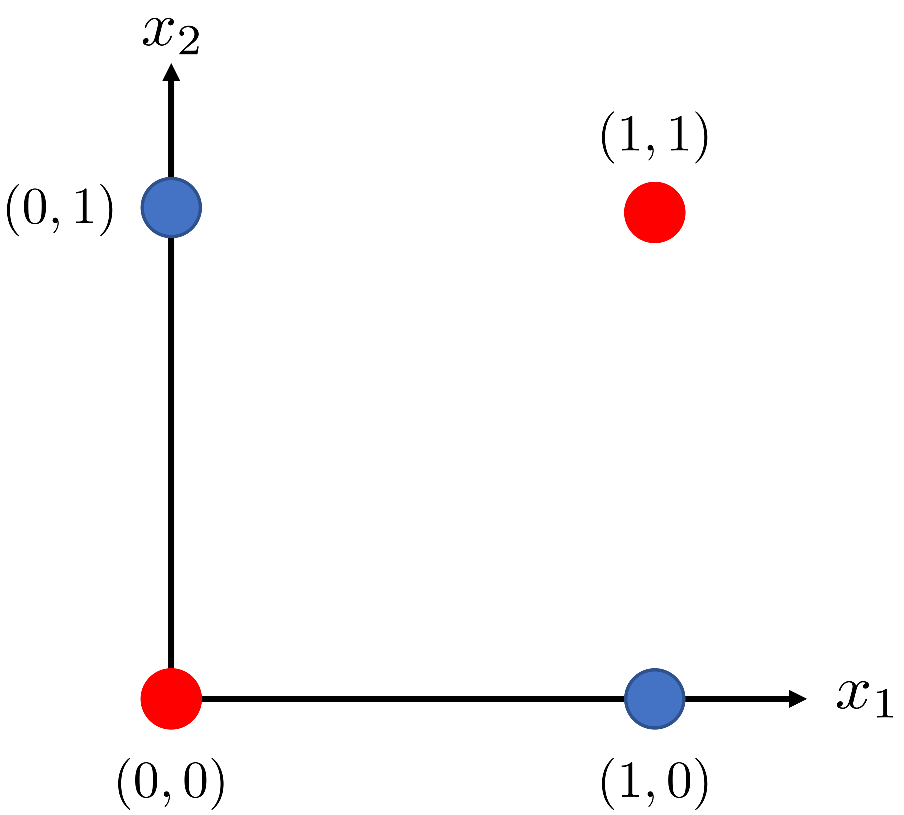
Figure source:
https://paperswithcode.com/method/relu

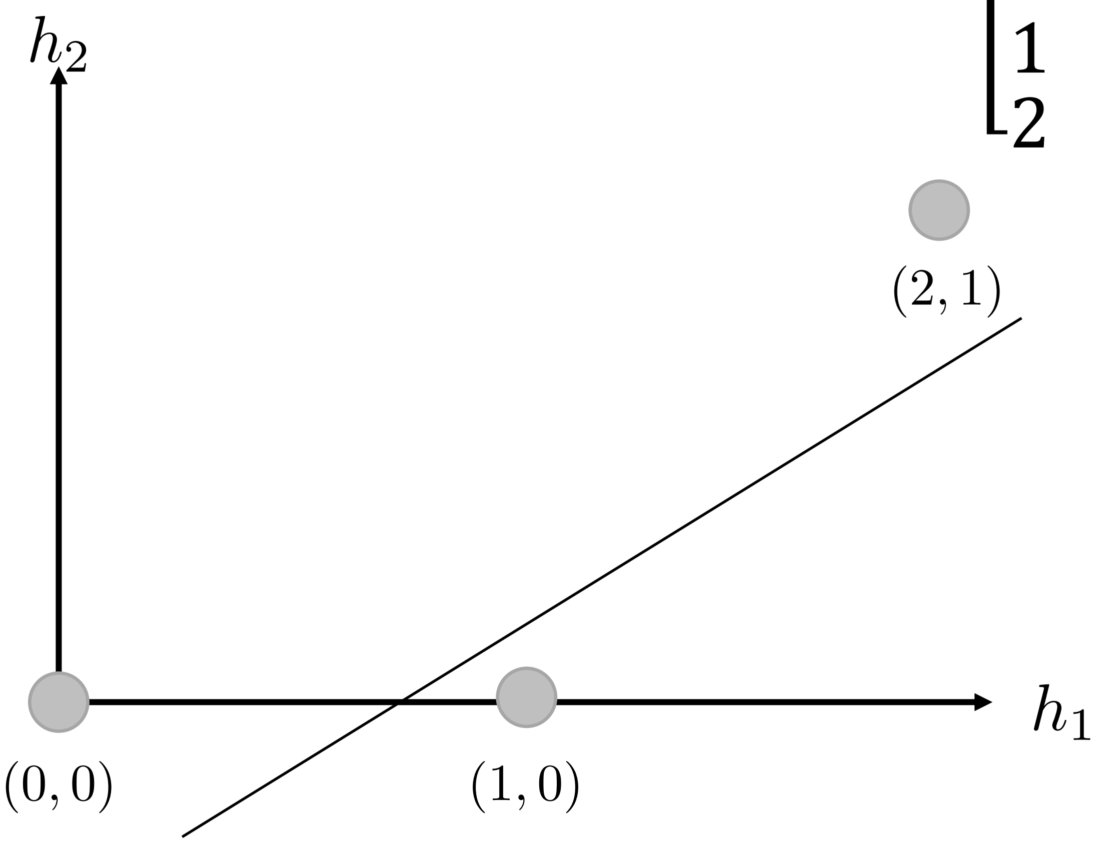
Math of MLP for the XOR Problem (2)

灰色點：隱藏狀態 (h)

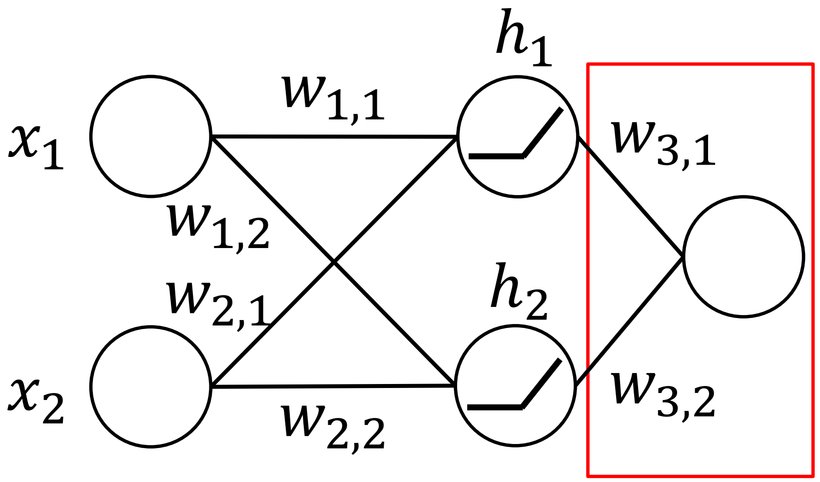
Current values

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$





Math of MLP for the XOR Problem (3)



任一筆資料

$$[x_1 \quad x_2] \times \begin{bmatrix} \overset{1}{w_{3,1}} \\ \underset{-2}{w_{3,2}} \end{bmatrix} + [\overset{0}{b_3}]$$

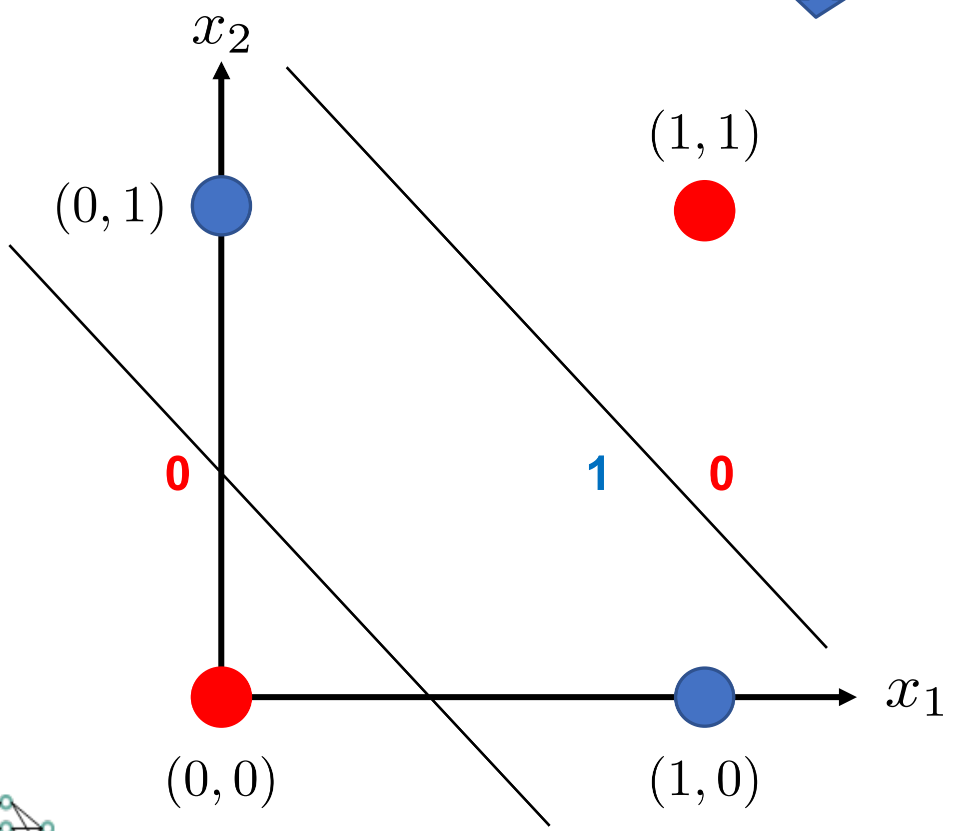
$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ -2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

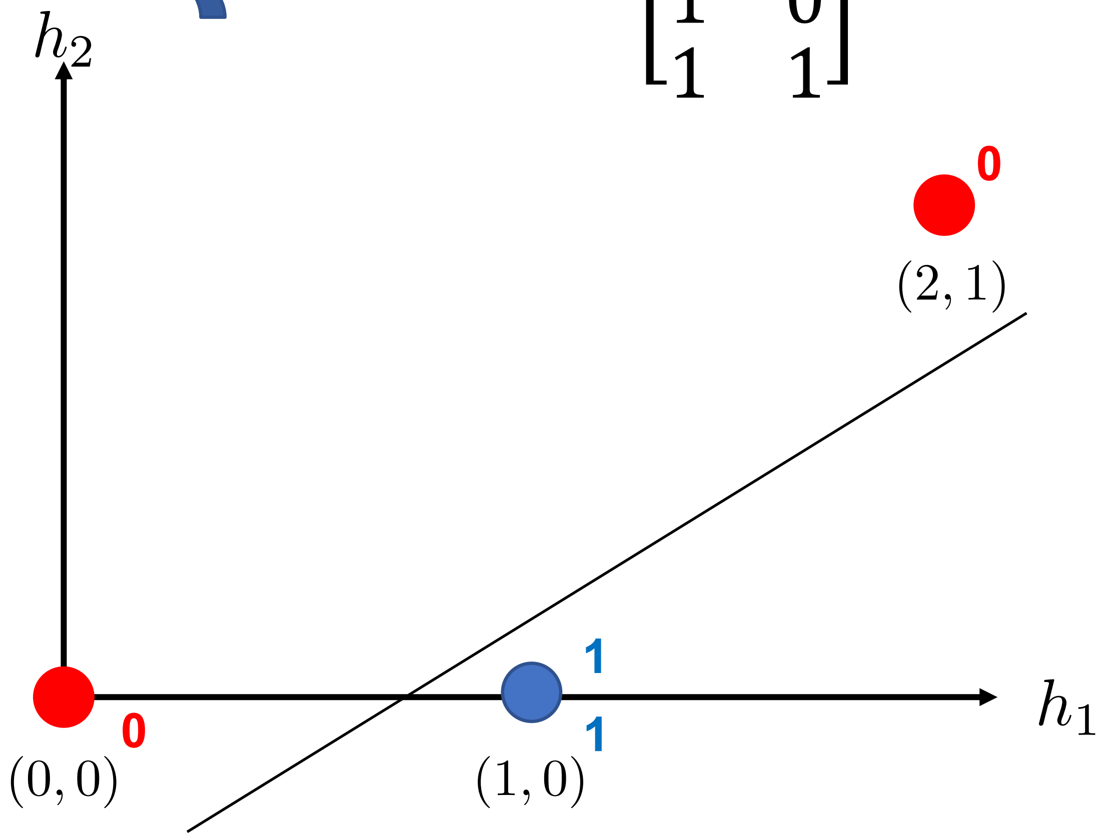
可視為分類層

Visualization of MLP for the XOR Problem

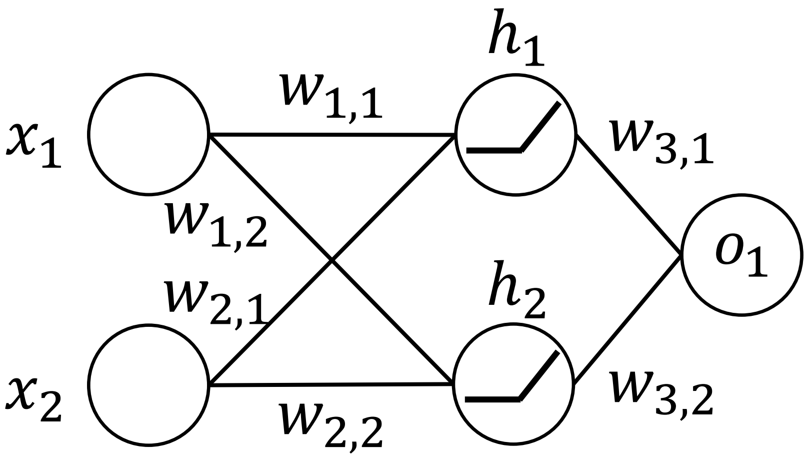
相當於隱藏層+非線性轉換
為原始資料點帶來兩條分割線

Initial values $\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$ Final values $\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$





上一頁的計算過程



x1	x2	XOR	o_1
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

$$o_1 = h_1 w_{3,1} + h_2 w_{3,2} + b_3$$

$$= \mathrm{ReLU}(x_1 \underset{1}{w_{1,1}} + x_2 \underset{1}{w_{2,1}} + \underset{0}{b_1}) \underset{1}{w_{3,1}} + \mathrm{ReLU}(x_1 \underset{1}{w_{1,2}} + x_2 \underset{1}{w_{2,2}} + \underset{-1}{b_2}) \underset{-2}{w_{3,2}} + \underset{0}{b_3}$$

Summary (for XOR with MLP)

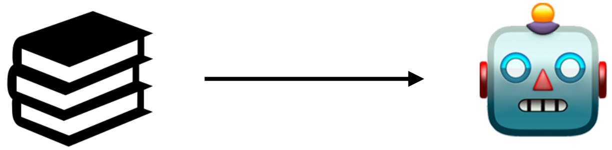
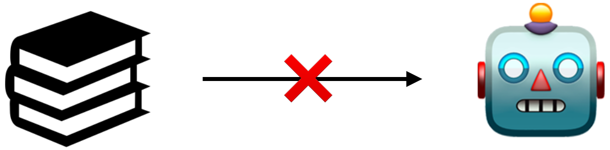
- XOR 在輸入空間中 線性不可分，一般線性分類器無法解決
- 非線性轉換 (ReLU) 創造隱藏空間，讓資料從不可分 → 可分
  - 在隱藏空間中，XOR 點已經 線性可分
  - 輸出層只需一條線 (線性分類器) 就能完成分類
- MLP 的力量：線性 + 非線性堆疊 → 能解決更複雜的決策邊界

More resources for XOR

- Community blog post
  - https://dev.to/jbahire/demystifying-the-xor-problem-1blk
  - https://chih-sheng-huang821.medium.com/機器學習-神經網路-多層感知機-multilayer-perceptron-mlp-運作方式-f0e108e8b9af
- Stanford course:
  - https://youtu.be/s7nRWh_3BtA?si=ZEGRN5pmPchM0NxM
- Also, in Deep Learning book (Chapter 6.1)

Gradient Descent

Machine Learning Approaches

Approaches	Learning Process	Common Tasks
Supervised Learning (監督式學習)	 準備許多有答案 (label) 的資料來訓練模型	Classification Regression
Unsupervised Learning (無監督式學習)	 模型在沒有標註的資料下學習資料本身的內在結構或模式	Clustering Similarity Matching Dimensionality Reduction

Classification and Regression

Classification

→ Model →

Cat ☑

Dog

Apple

Regression

→ Model → 1100

[Example] Clustering

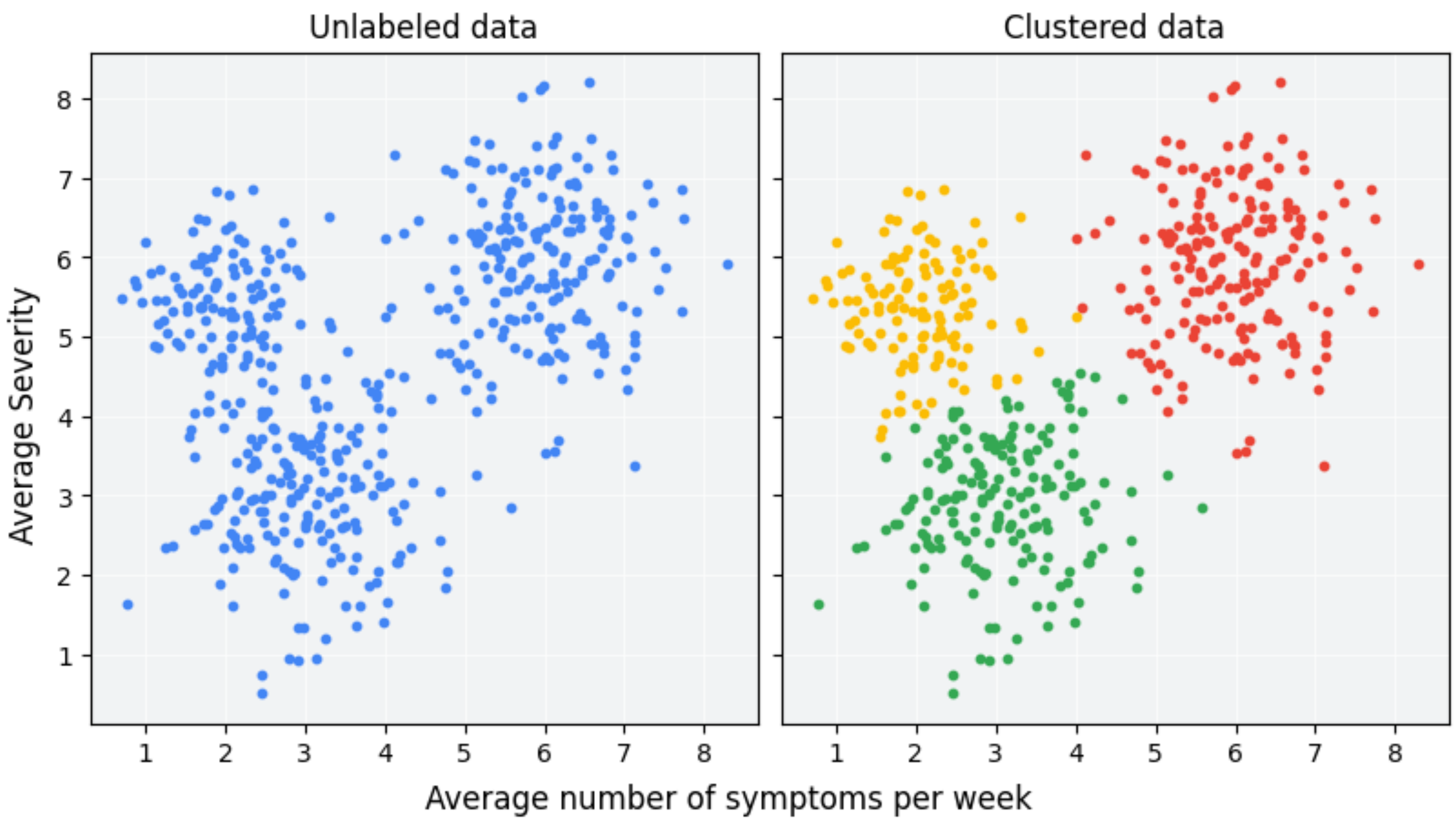
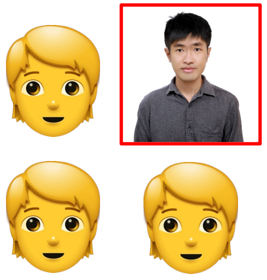


Figure source: https://developers.google.com/machine-learning/clustering/overview?hl=zh-tw

[Example] Similarity Matching

以人臉辨識為例

資料庫



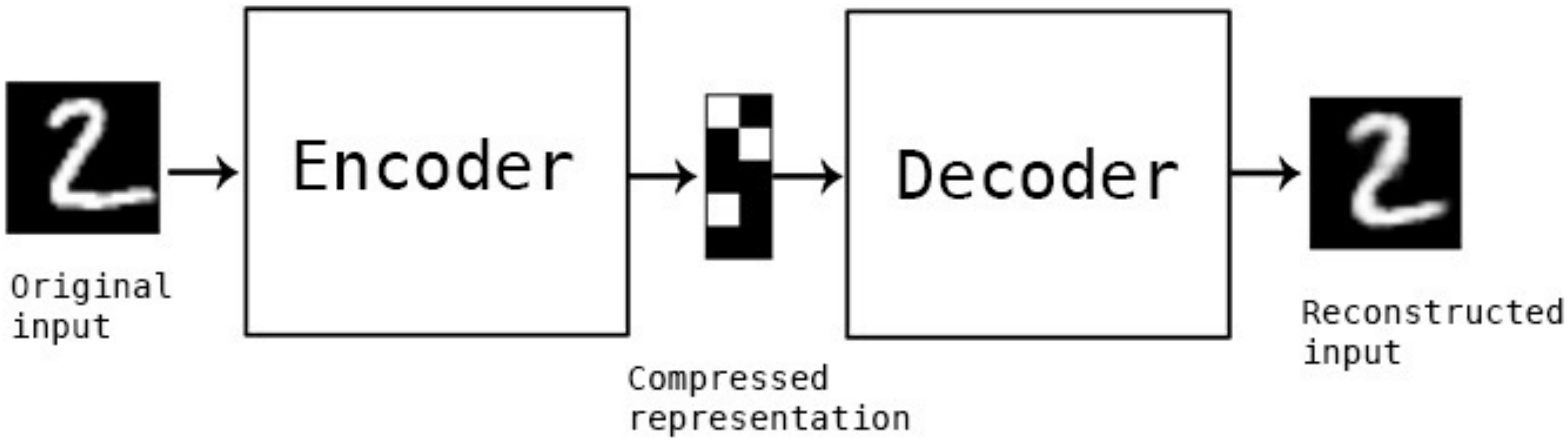
我被掃描的照片

Similarity Matching

Valid or Invalid

[Example] Dimensionality Reduction

以 AutoEncoder 為例



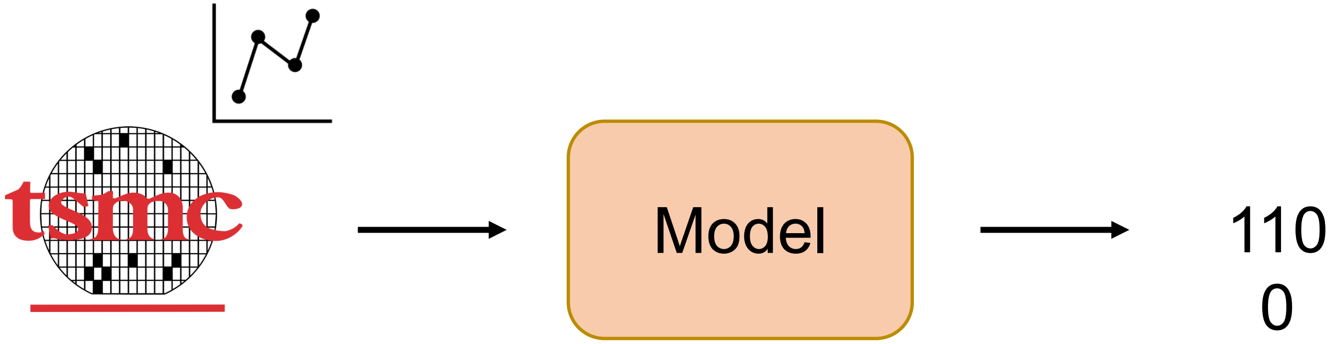
類似於近年的 self-supervised learning

Figure source: https://blog.keras.io/img/ae/autoencoder_schema.jpg

Regression

Regression Tasks (1)

Regression

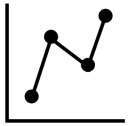


2/17	2/18	2/19	2/20	2/21
1080	1085	1095	1080	?

Prediction based on the past data (Time-series)

Regression Tasks (2)

<u>Regression</u>



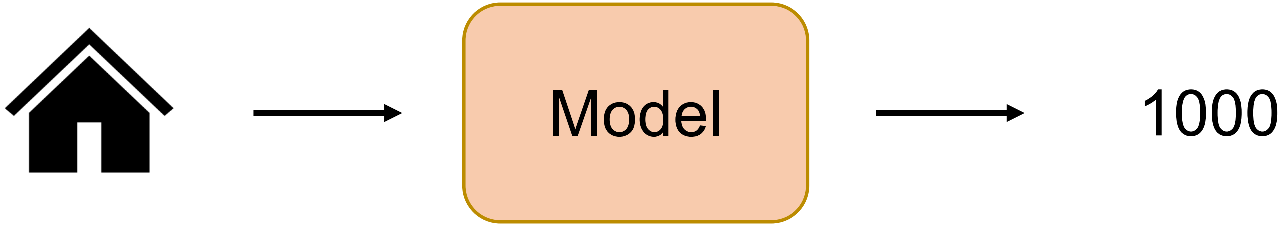
 → Model → 15 ° C

2/17	2/18	2/19	2/20	2/21
15 ° C	14 ° C	13 ° C	15 ° C	?

Prediction based on the past data (Time-series)

Regression Tasks (3)

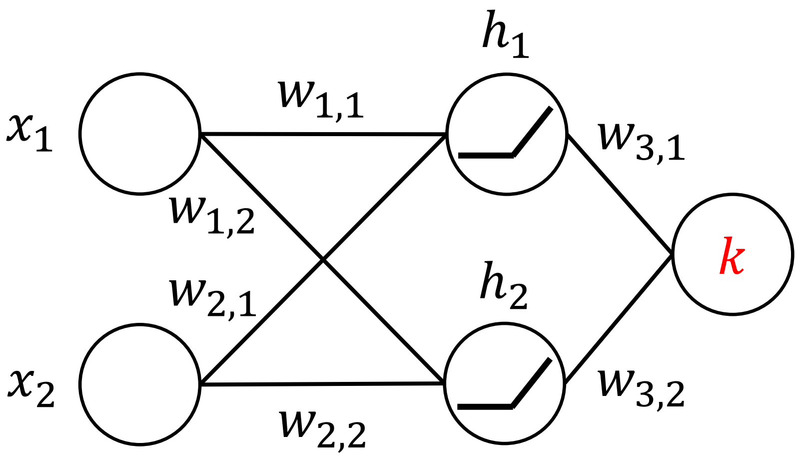
<u>Regression</u>



Area	City	Num of Rooms	House Age	Price
15	Taoyuan	3	10	?

Prediction based on the different features (Tabular data)

Machine Learning is to train a function



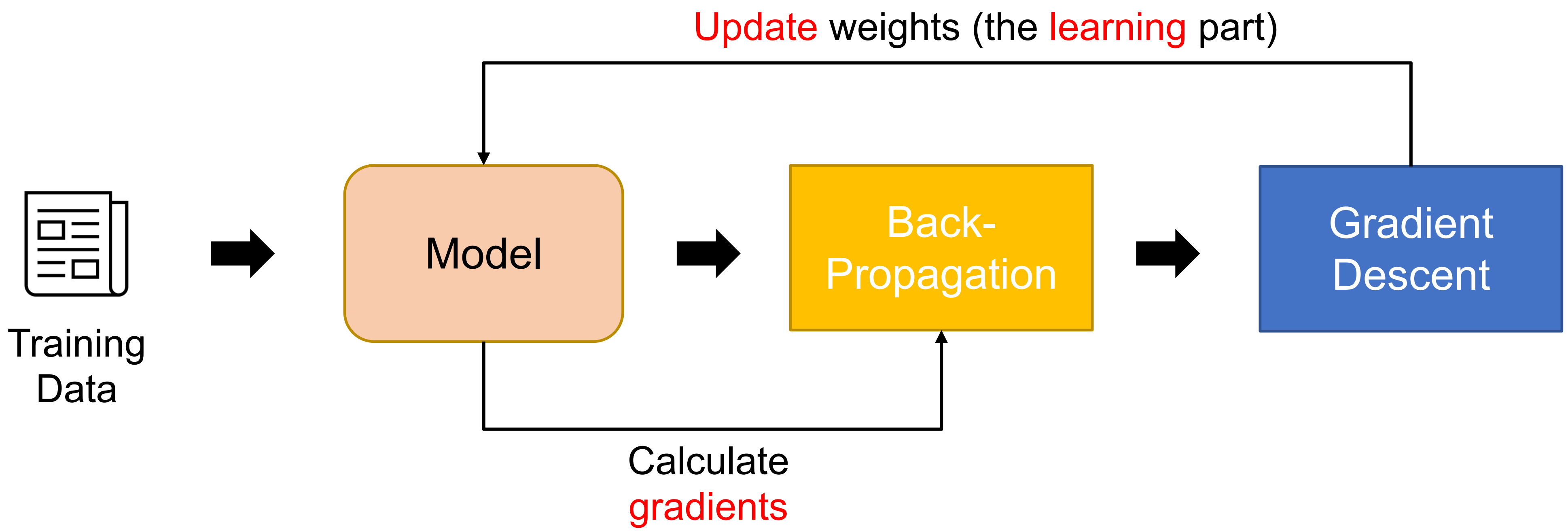
模型預測 (Predictions) ⟷ 正確答案 (Ground-truths)

$f(x_1, x_2) = k$

- 透過 Gradient Descent (梯度下降) 尋找最佳解
- 透過 Back-Propagation 的結果更新權重值 (w 跟 b)

Training Process of a Deep Learning Model

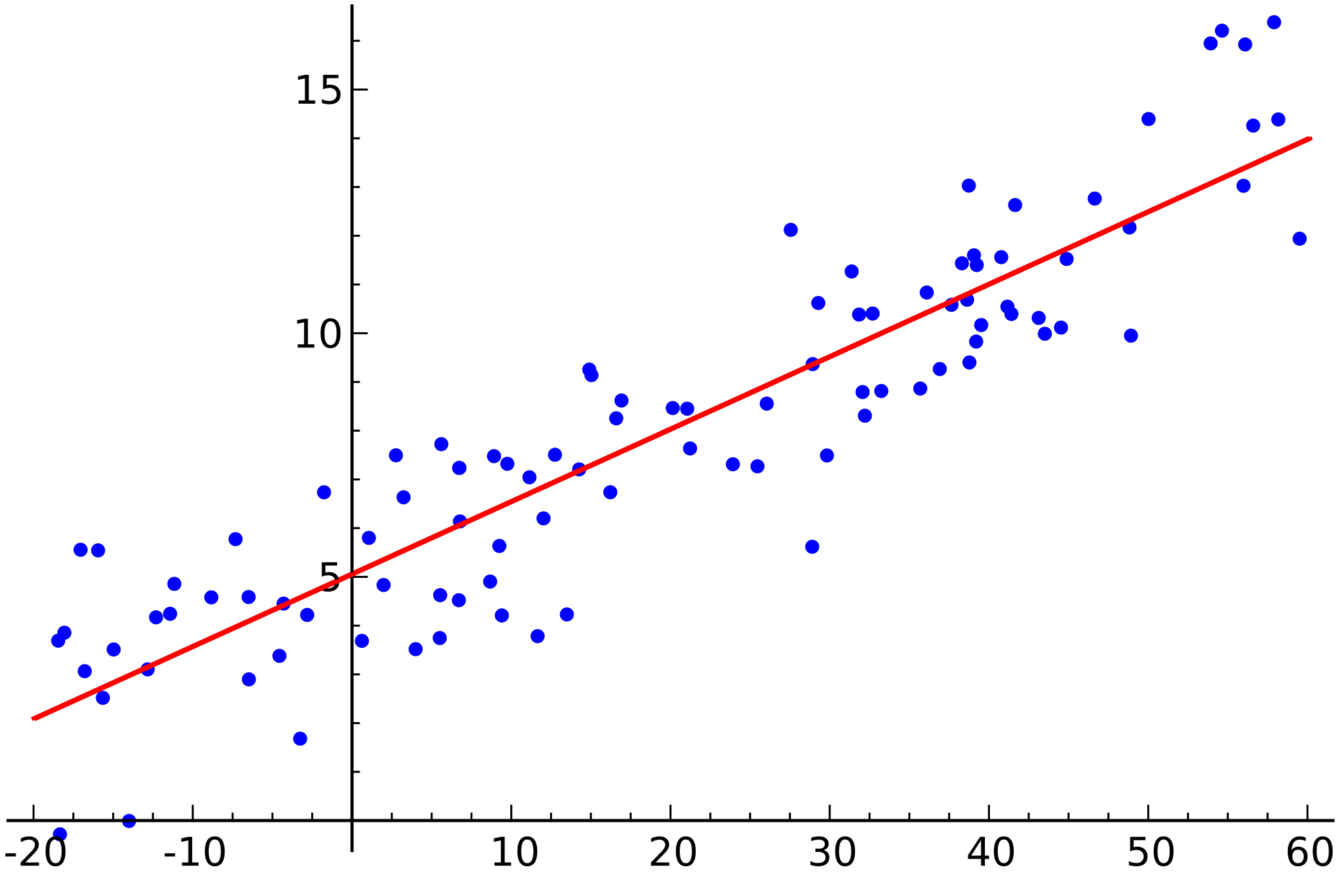
- 深度學習模型被訓練的流程



Gradient Descent vs. Back-propagation

	Gradient Descent	Back-propagation
功能	根據梯度更新權重值 (weights)	計算神經網路中的梯度，以供梯度下降使用
最終目標	透過梯度找出 function 最佳解 (最小化目標函數)	計算梯度，以便使用梯度下降找到最佳解

Linear Regression (線性迴歸)



紅線為直線方程式 $y = wx + b$

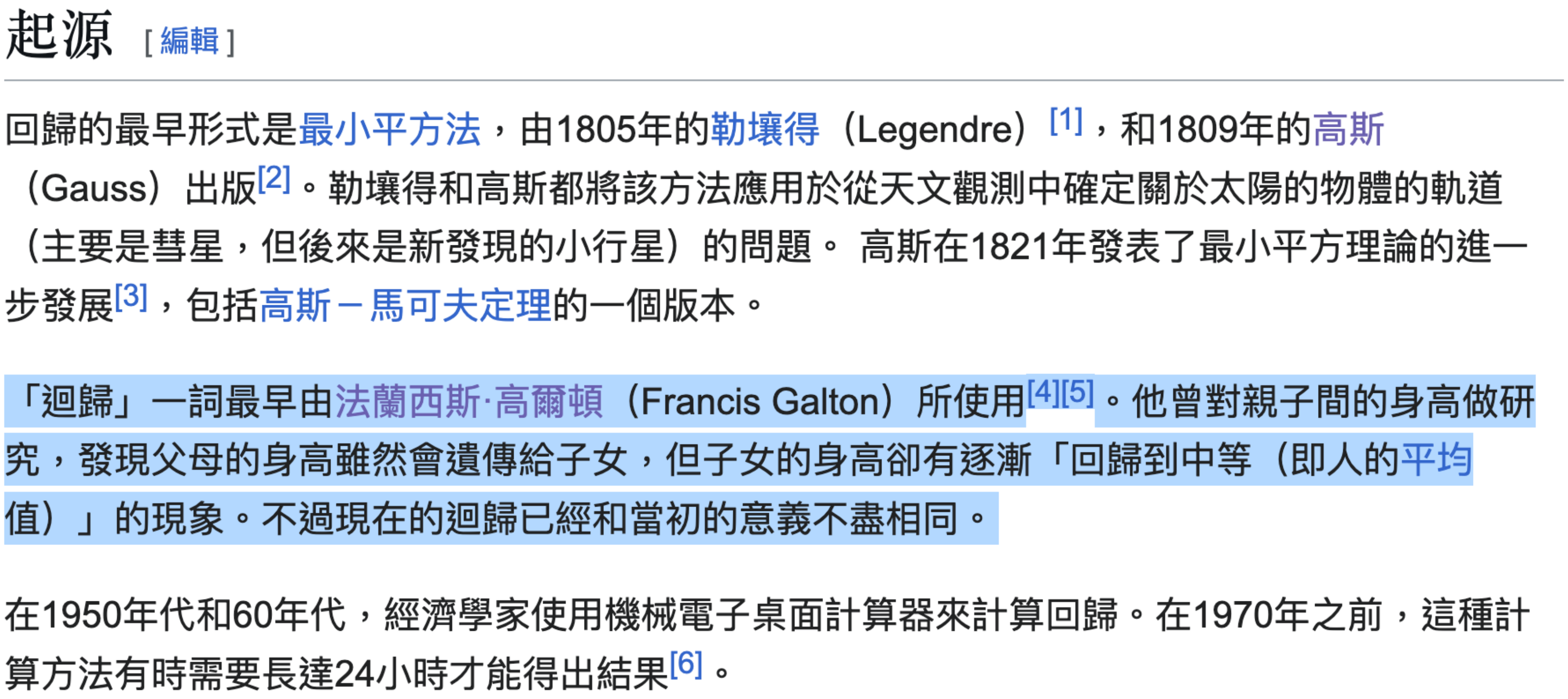
Linear regression 通常透過均方誤差來擬合資料

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

紅線代表可以使所有藍色點的MSE最小的w跟b代入直線方程式後得到的線

https://zh.wikipedia.org/zh-tw/迴歸分析

為什麼Regression叫做Regression？



https://zh.wikipedia.org/zh-tw/迴歸分析
https://stats.stackexchange.com/questions/11087/why-are-regression-problems-called-regression-problems

Math Warning!!

What are gradients? (數學定義)

- First-order derivative (s)
  - Univariate function: a scaler
  - Multivariate function: a vector
- Univariate Example:

Original function: $f(x) = x^2$

First-order derivative: $f'(x) = 2x = \nabla_x f$

What are gradients? (數學定義)

- First-order derivative (s)
  - Univariate function: a scaler
  - Multivariate function: a vector
- Multivariate Example:

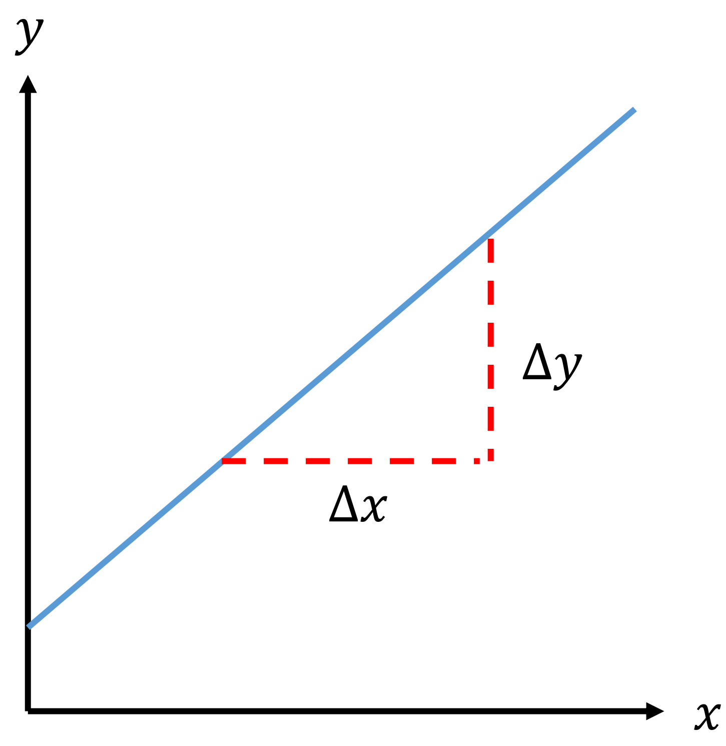
Original function: $f(x, y) = x^2 + y^2$

First-order derivatives: $f'(x, y) = \begin{bmatrix} 2x \\ 2y \end{bmatrix} = \nabla_{x,y} f$

What are gradients? (數學意義)

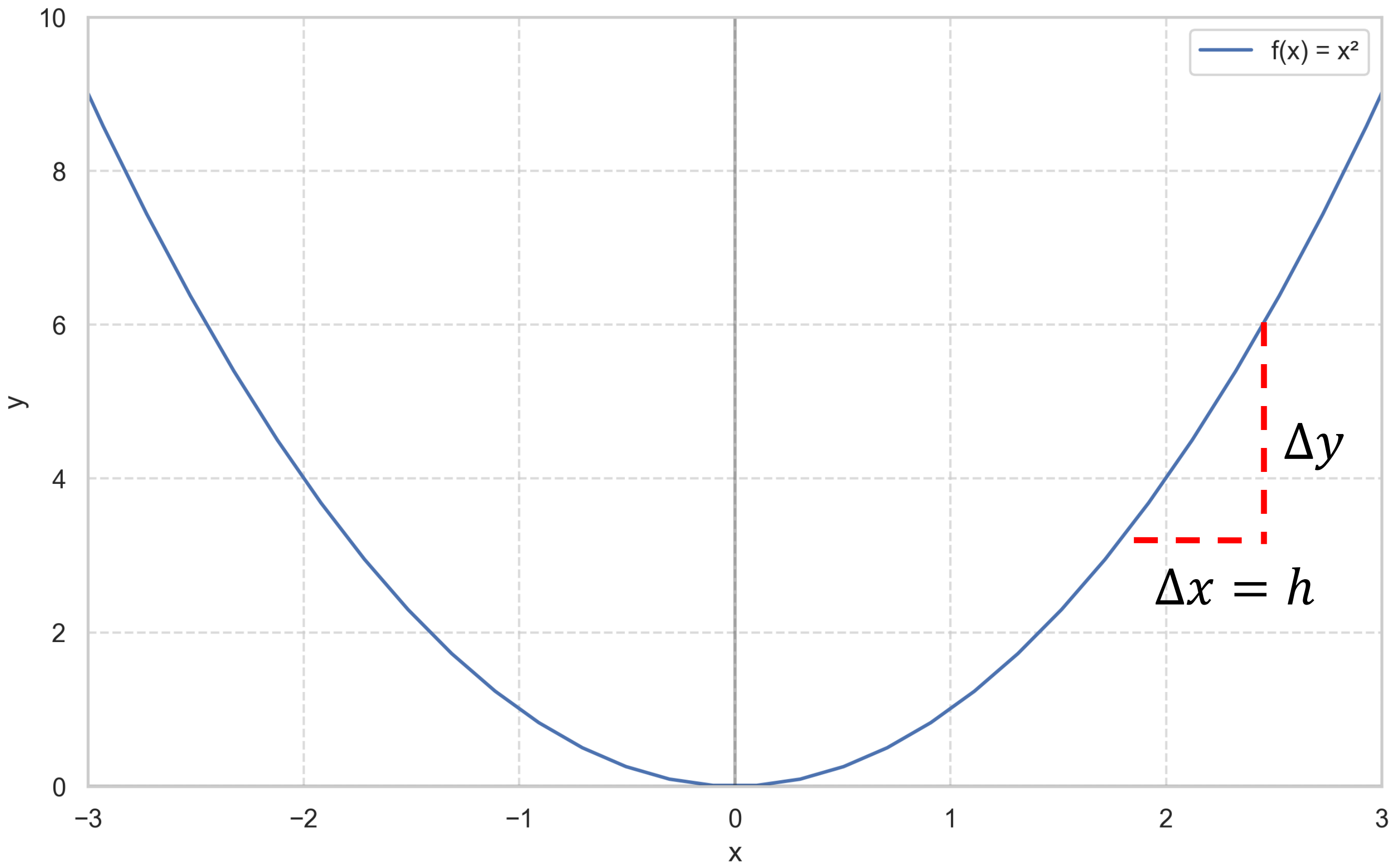
$y = mx + b$

Slope (斜率) = m



$$
\begin{aligned}
\text{斜率} &= \frac{\Delta y}{\Delta x} \\
&= \frac{y_2 - y_1}{x_2 - x_1} \\
&= \frac{(mx_2 + b) - (mx_1 + b)}{x_2 - x_1} \\
&= \frac{m(x_2 - x_1)}{x_2 - x_1} \\
&= m
\end{aligned}
$$

What are gradients? (數學意義)



斜率$= \frac{\Delta y}{\Delta x}$ $\qquad f(x) = x^2$

一次導函數：

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

$$= 2x$$

梯度等於一次導函數的值，也就是斜率

計算導數

$f(x) = x^2$

$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$

$= \lim_{h \to 0} \frac{(x+h)^2 - x^2}{h}$

$$= \lim_{h \to 0} \frac{x^2 + 2xh + h^2 - x^2}{h}$$

$= \lim_{h \to 0} 2x + h \quad = 2x$ (h 趨近於 0)

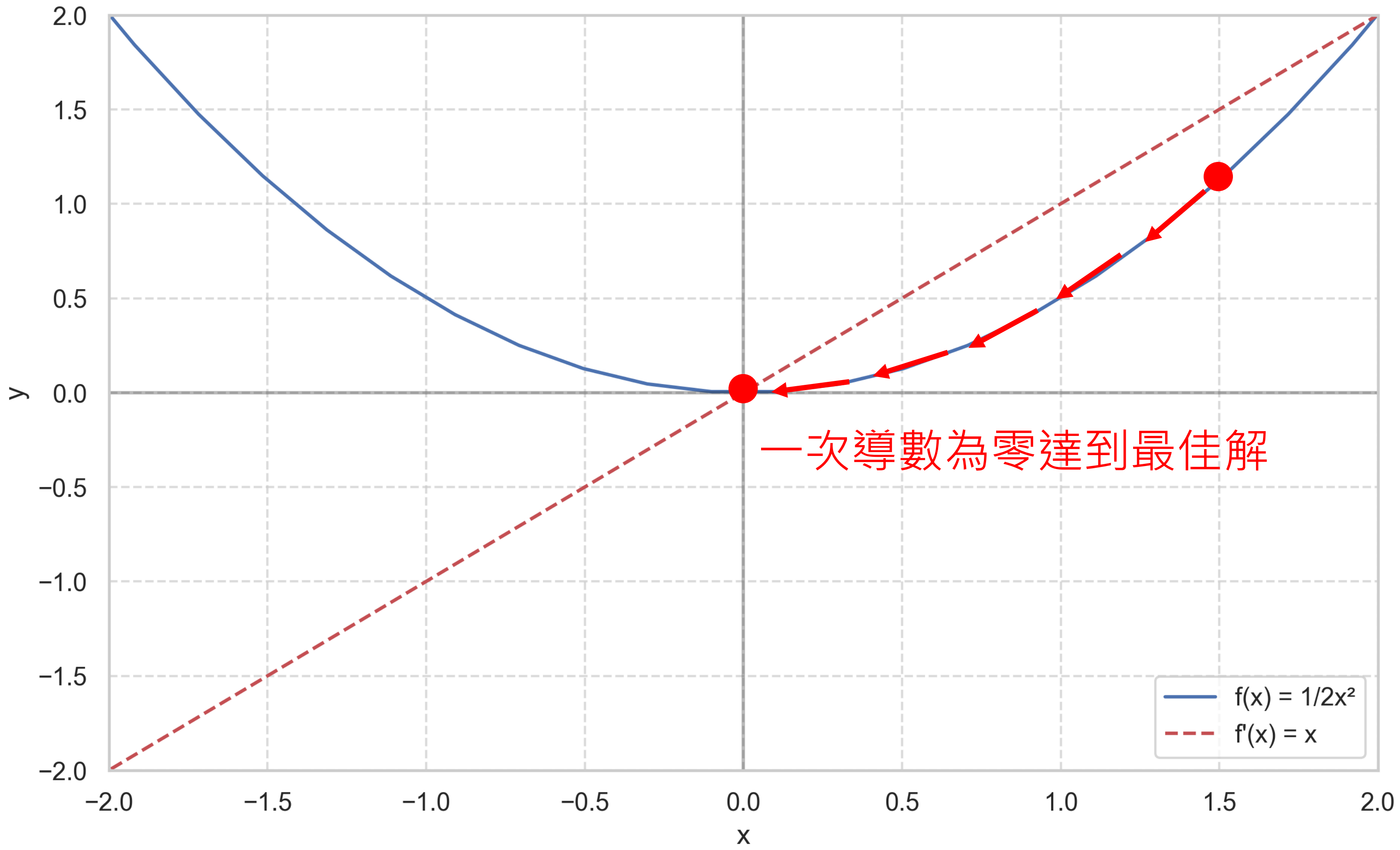
謝謝陳冠甫老師找出公式錯誤

What are gradients? (物理意義)

$$f'(x) = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h}$$

- 有一跑者10分鐘內跑了2km，請問這個人一秒鐘跑多少公里？
- 速度 = 距離 / 時間
- 一秒鐘可以被想成h趨近於0
- 因此，梯度代表在極短情況 (這裡用時間舉例) 的變化量

Gradient Descent



Gradient Descent (梯度下降法)

Assume x is a trainable parameter (weight), f is a differentiable function:

Gradient descent: $$x' = x - \eta \nabla_x f(x)$$

Objective function (example): $$f(x) = \frac{1}{2}x^2$$

η is the learning rate used for gradient descent.

What “Function” we‘re going to train?

- Loss Function
- 以均方誤差 (Mean Squared Error) 為例：

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

我們要訓練的 function

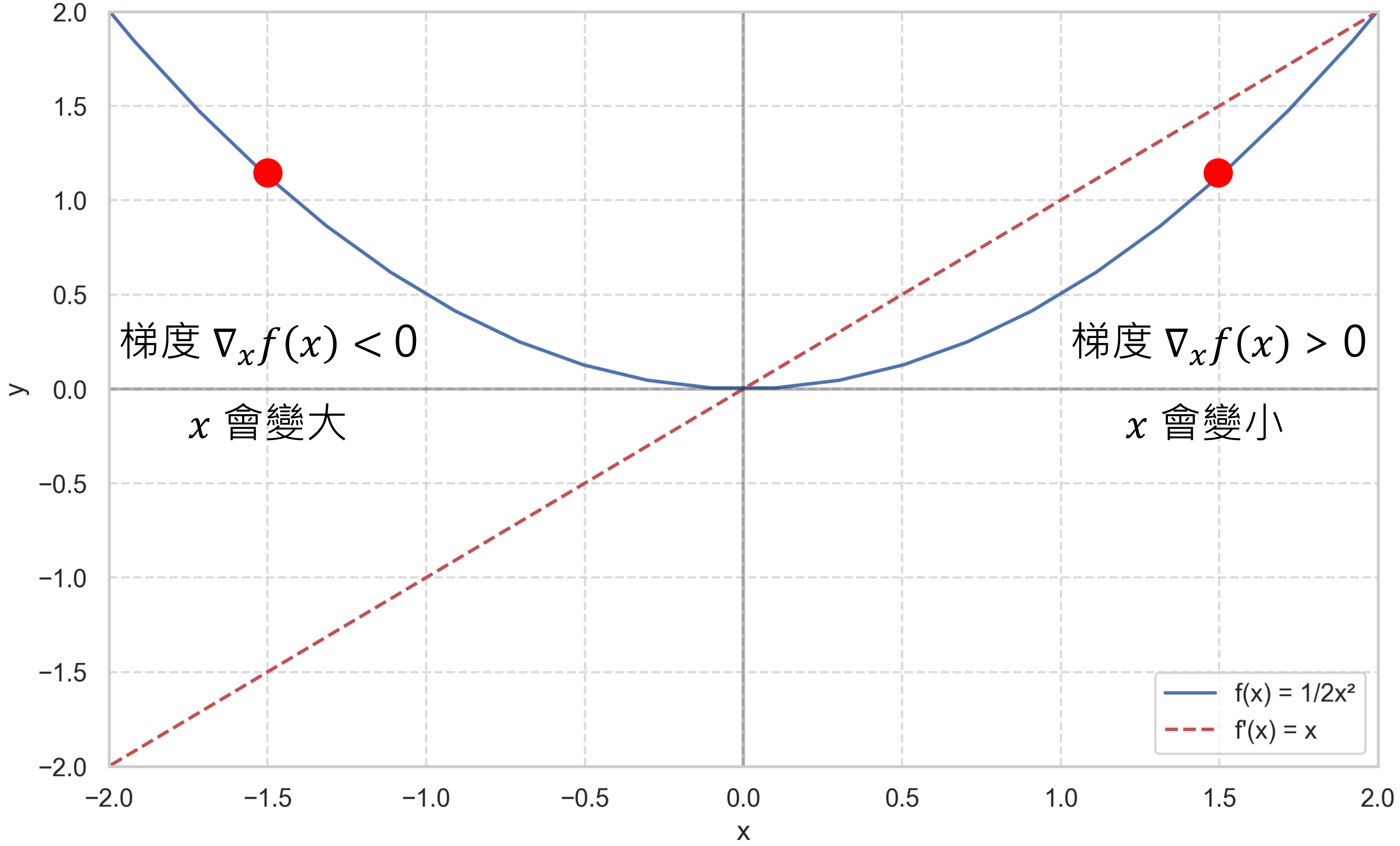
$$= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \underline{(wx_\mathrm{i} + b)}\right)^2$$

假設模型的輸出是 $wx_\mathrm{i} + b$

- 其中:
  - n 代表有 n 筆訓練資料
  - y_i 爲任一筆 ground-truth、$\hat{y}_i$ 爲任一筆 prediction (model output)

以兩個點來觀察 Gradient Descent 的特性

$$x' = x - \eta \nabla_x f(x)$$

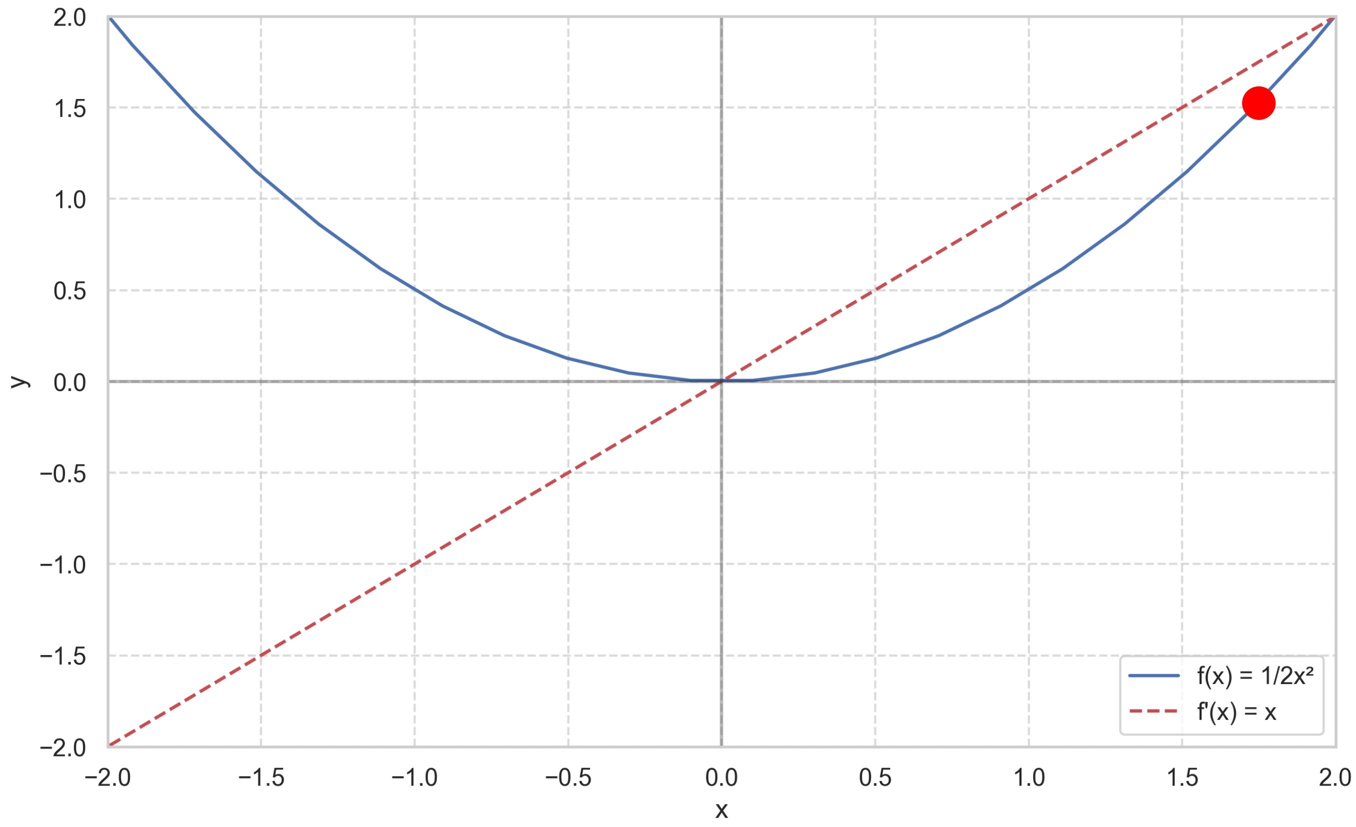


為什麼可以用位置減梯度？(1/3)

- 梯度主要是提供參數要修正的「方向」
  - 正號或負號
- 讓參數的值改變最後順利走到最低點

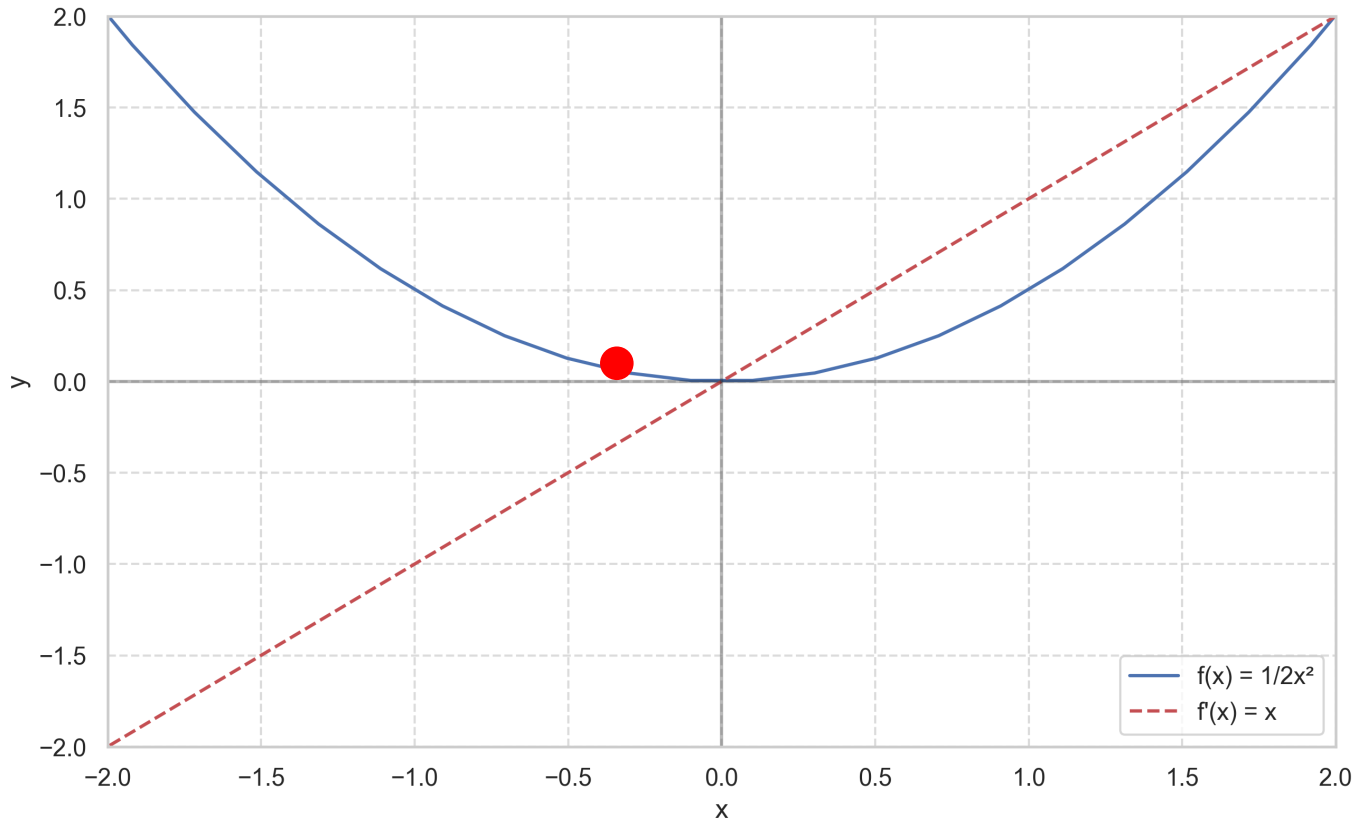
為什麼可以用位置減梯度？(2/3)

- 如果|梯度|很大的話，表示該位置還在非常陡峭的位置，通常在這時候參數的值確實需要改變多一點



為什麼可以用位置減梯度？(3/3)

- 如果|梯度|很小的話，表示該位置在函數平滑的位置，通常在這時候參數的值可能需要改變少一點



Gradient Descent 流程推導

Minimize a Regression Model

- 以均方誤差 (Mean Squared Error) 為例：

$$\mathcal{L} = \frac{1}{n}\sum\nolimits_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (wx_{\mathrm{i}} + b)\right)^2$$

- 其中:
  - $\mathcal{L}$ 代表 Loss function；n 代表有 n 筆訓練資料
  - y_i 爲任一筆 ground-truth、$\hat{y}_i$ 爲任一筆 prediction (model output)

Minimize a Regression Model

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (wx_\mathrm{i} + b)\right)^2$$

對w進行偏微分

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{1}{n}\sum_{i=1}^{n} 2\left(y_\mathrm{i} - (wx_\mathrm{i} + b)\right) \cdot (-x_\mathrm{i})$$

對b進行偏微分

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{1}{n}\sum_{i=1}^{n} 2\left(y_i - (wx_\mathrm{i} + b)\right) \cdot (-1)$$

The generalized power rule for derivatives

Assume $y = [u(x)]^n$; and $u(x)$ is differentiable.

Then $\dfrac{dy}{dx} = n \cdot [u(x)]^{n-1} \cdot u'(x)$

$$\boxed{\begin{aligned} \frac{\partial \mathcal{L}}{\partial w} &= \frac{1}{n}\sum_{i=1}^{n} 2\big(y_{\mathrm{i}} - (wx_{\mathrm{i}} + b)\big) \cdot (-x_{\mathrm{i}}) \\ \frac{\partial \mathcal{L}}{\partial b} &= \frac{1}{n}\sum_{i=1}^{n} 2\big(y_i - (wx_{\mathrm{i}} + b)\big) \cdot (-1) \end{aligned}}$$

證明： http://www.math.ncu.edu.tw/~yu/ecocal95/boards/lec12_c_95.pdf (from 中央大學數學系)

Minimize a Regression Model

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{1}{n}\sum_{i=1}^{n} 2\big(y_{\mathrm{i}} - (wx_{\mathrm{i}} + b)\big) \cdot (-x_{\mathrm{i}})$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{1}{n}\sum_{i=1}^{n} 2\big(y_i - (wx_{\mathrm{i}} + b)\big) \cdot (-1)$$

更新w：

$$w_t = w_{t-1} - \eta \frac{\partial \mathcal{L}}{\partial w_{t-1}}$$

更新b：

$$b_t = b_{t-1} - \eta \frac{\partial \mathcal{L}}{\partial b_{t-1}}$$

- η 代表 learning rate，扮演梯度下降過程的重要角色

The impact of a Learning Rate

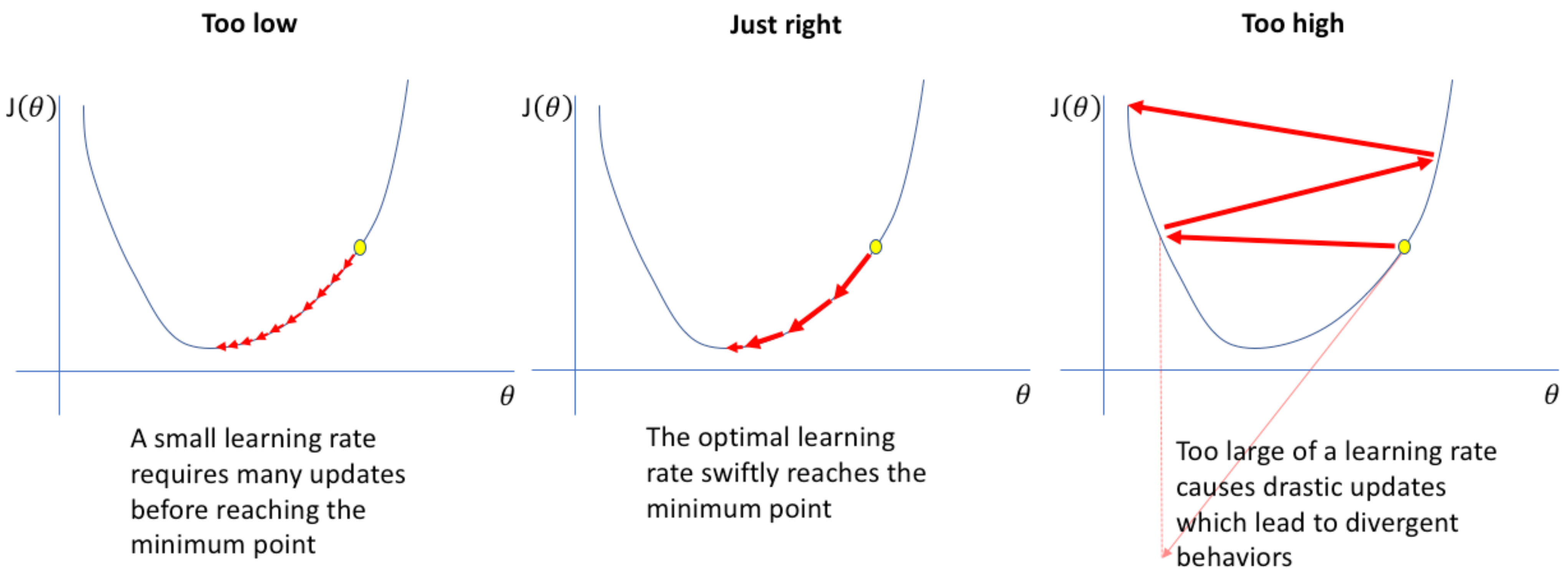


Figure source: https://www.jeremyjordan.me/nn-learning-rate/

How to choose the best learning rate?

Figure source:
https://x.com/karpathy/status/801621764144971776

More resources for Calculus

https://calcgospel.top/wp-content/uploads/2020/06/02-01-微分的定義.pdf

More resources for gradient descent

Deep Learning Book Chapter 4.3: Gradient-Based Optimization

- https://www.deeplearningbook.org/contents/numerical.html

https://www.youtube.com/watch?v=fegAeph9UaA

https://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2016/Lecture/Regression%20(v6).pdf

Thank you!

Instructor: 林英嘉

✉️ yjlin@cgu.edu.tw