

# 110年第2學期 智慧計算實驗

Computational Intelligence Laboratory

人工智慧

物聯網

電腦視覺

網路通訊

東華大學電機工程學系





課程大綱

# COURSE OUTLINE

1

智慧計算(Computational Intelligence)

2

人工智慧(Artificial Intelligence, AI) 理論與實作

3

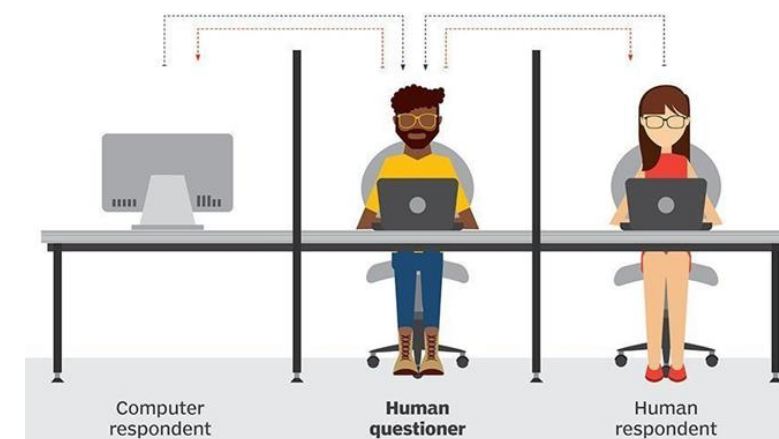
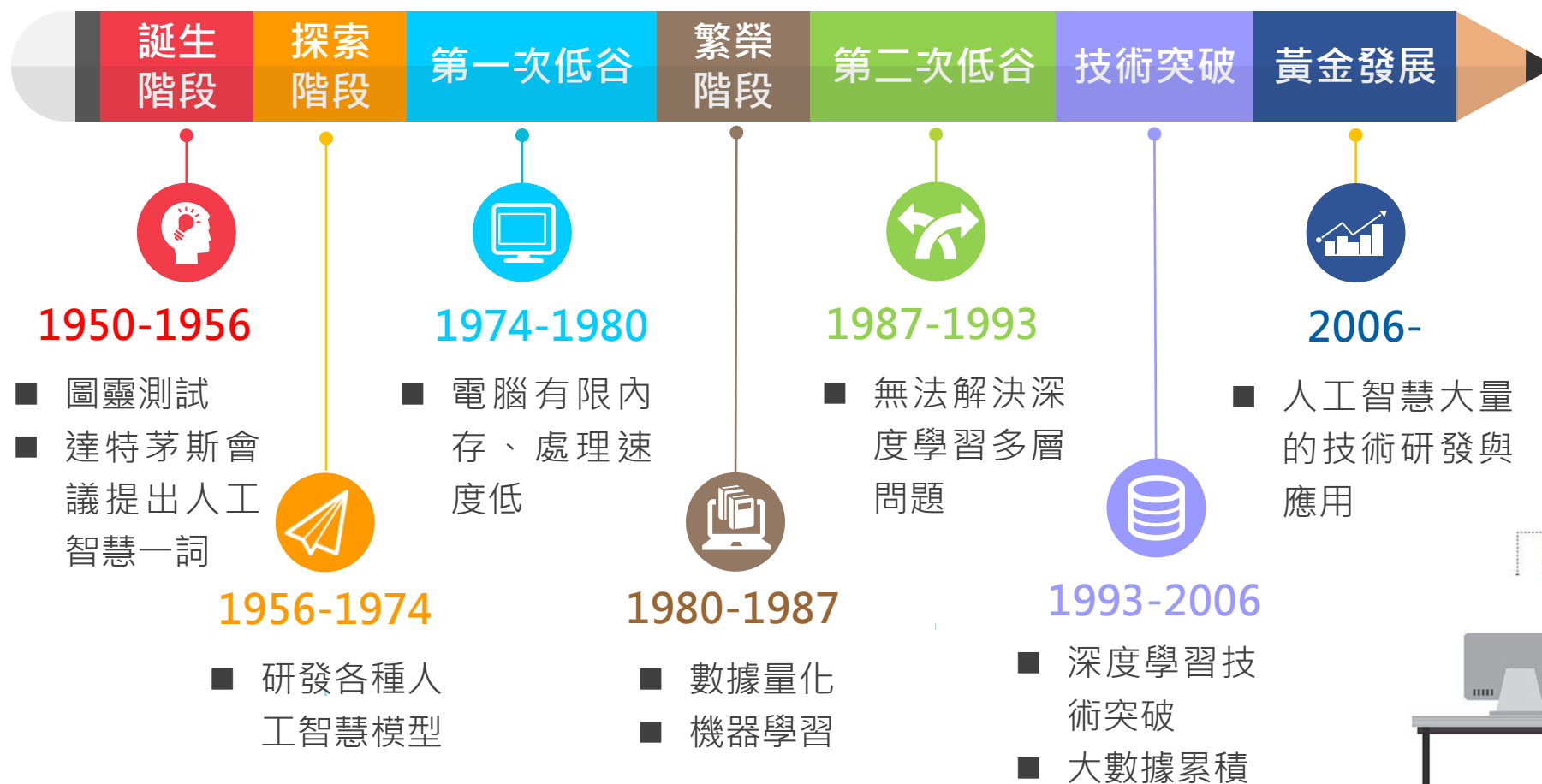
人工智慧在電腦視覺、網絡通訊與物聯網上的最新應用

註

應先修課程：程式設計(一)

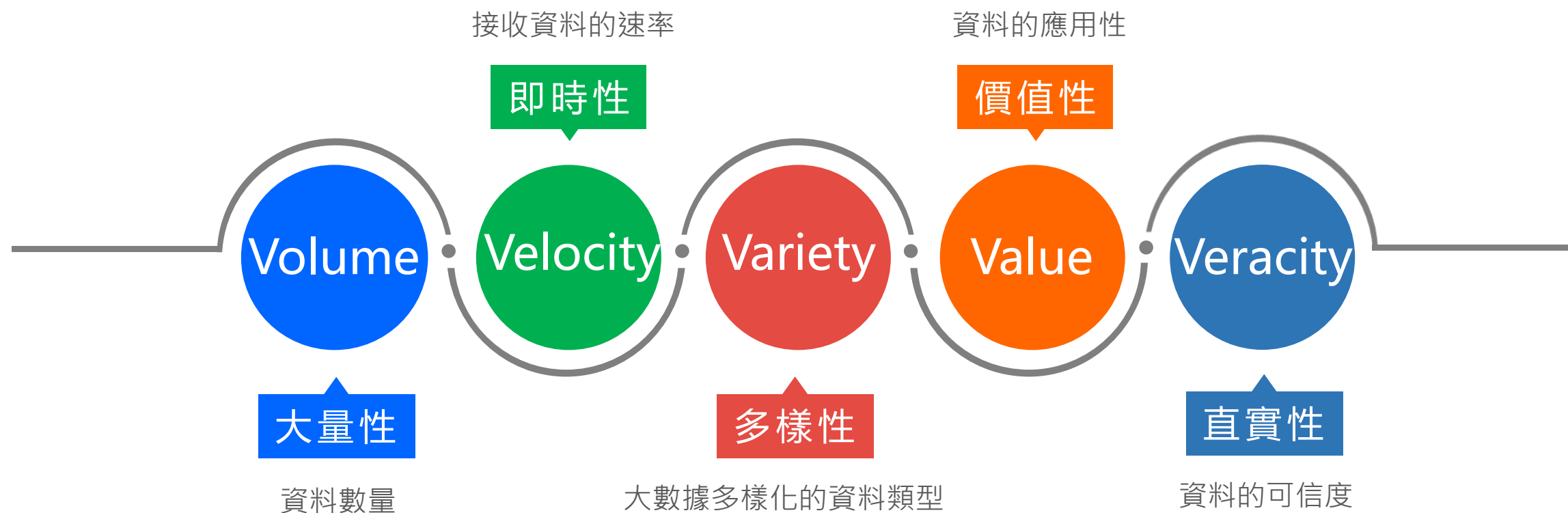


# 01 發展歷程



## 02 大數據

指更龐大且更複雜的資料集，使傳統的資料處理軟體已無力招架



## 03 人工智慧



弱人工智慧

Weak Artificial Intelligence

- 限定領域
- 解決特定問題



強人工智慧

Strong Artificial Intelligence

- 通用領域
- 能勝任人類工作



超人工智慧

Super Artificial Intelligence

- 超越人類智慧

## 04 學習任務

機器學習：透過大量的資料來讓機器(模型)，學習到如何正確預測及判斷

在數學定義上是找到一函數，能讓資料輸入函數得到輸出值，其值與實際值越相近

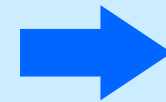
輸入

函數

輸出

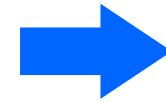
回歸

PM<sub>2.5</sub>、CO、SO<sub>2</sub>等



呼吸道感染人數

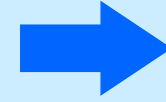
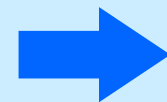
分類



貓 (0) / 狗 (1)

類別

分類  
定位



影像的像素值

目標物的類別及其位置

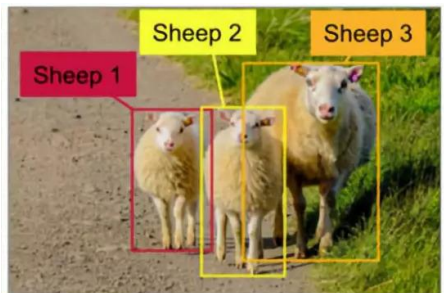


# 04 學習任務 (電腦視覺)



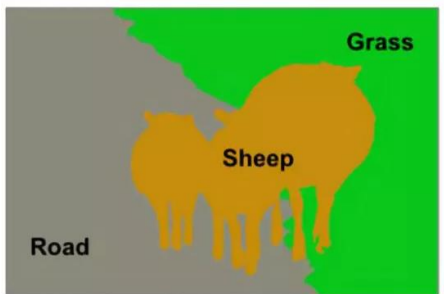
## 分類 (Classification)

- 只判別是否有該類別，無法得知位置與數量
- 輸出為類別



## 物件偵測 (Object Detection)

- 又稱為目標檢測，具有分類與定位 (回歸)
- 輸出為數個預測框與其類別

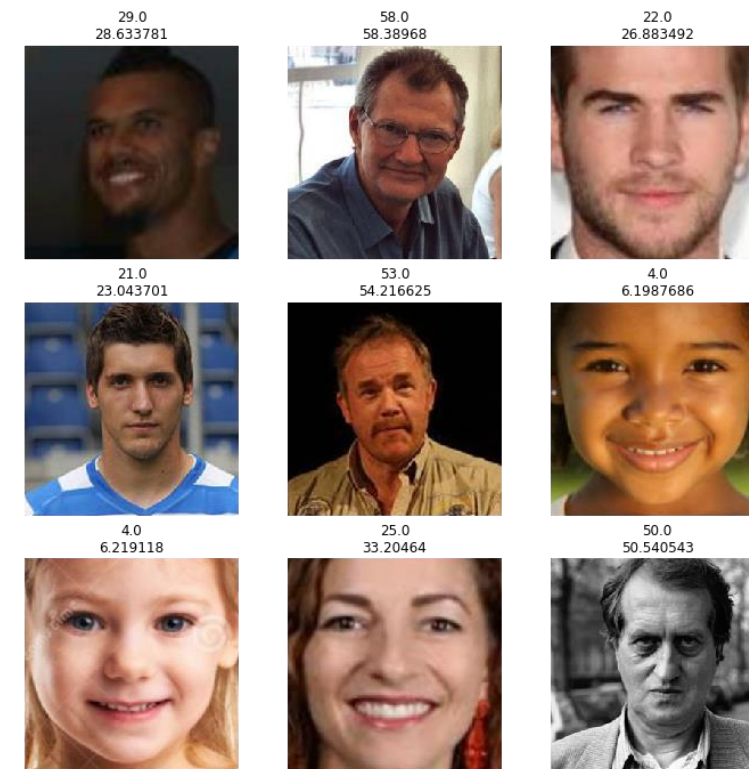


## 影像分割 (Image Segmentation)

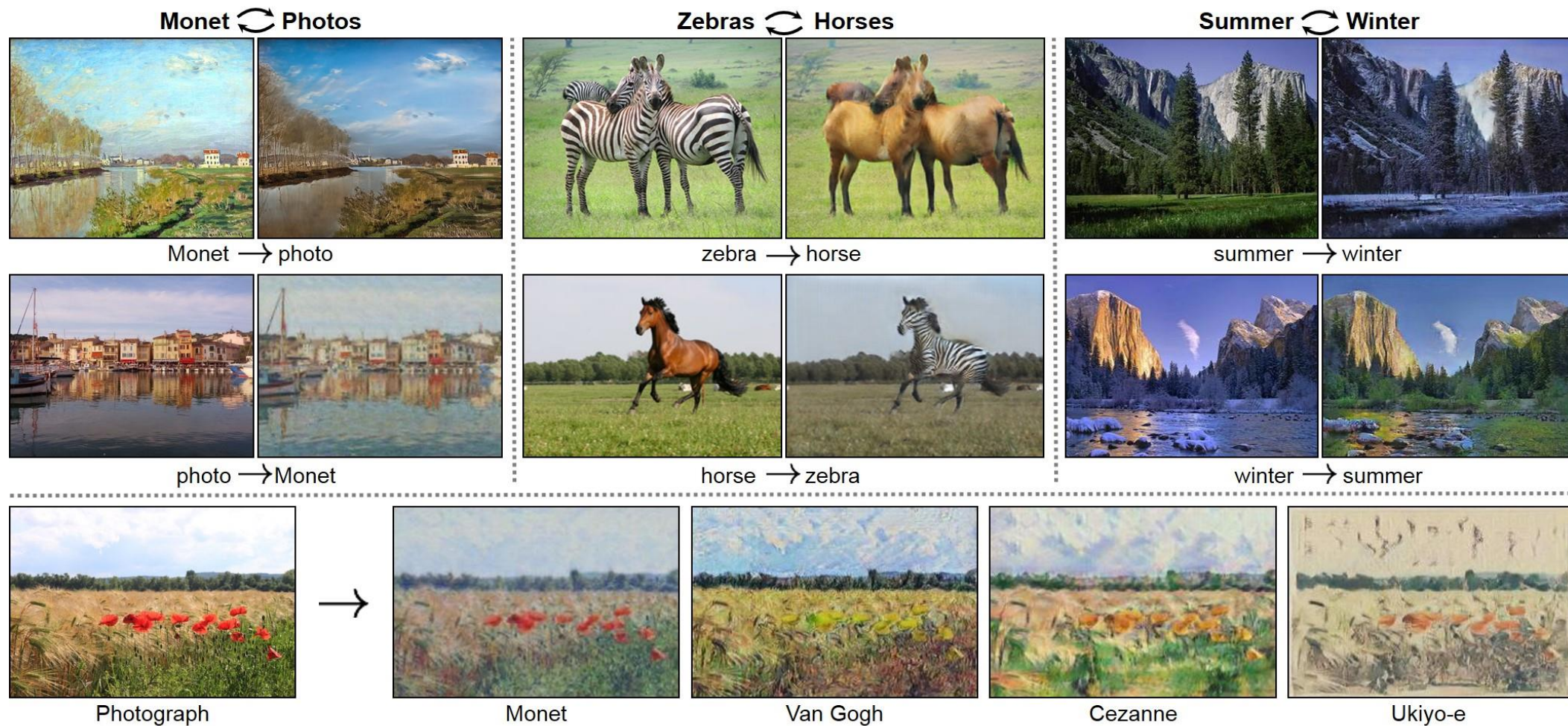
- 又有分語義分割、實例分割 與 全景分割
- 輸出為每個像素的類別

## 回歸 (Regression)

- 輸出是任一連續數值



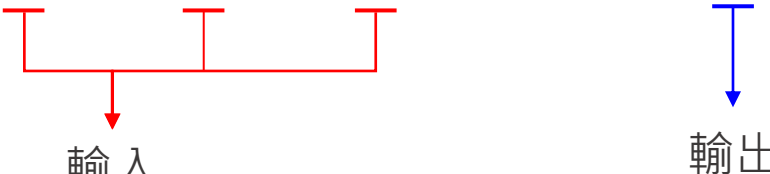
## 04 學習任務



[1] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", *IEEE International Conference on Computer Vision (ICCV)*, 2017.



# 05 回歸 機器學習判斷身體質量指數

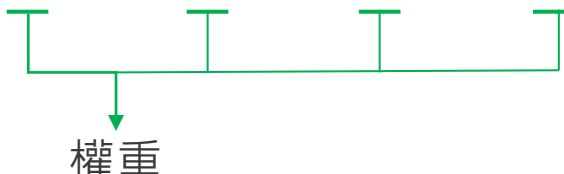
$$f(x) = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_0 = y$$


輸入

輸出

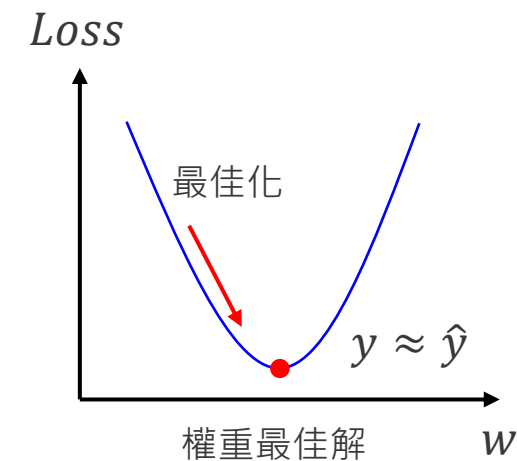
ID	身高 $x_1$	體重 $x_2$	年齡 $x_3$
1	160	47	22
2	180	90	32
3	157	55	18
⋮	⋮	⋮	⋮
999	168	65	21
1000	178	89	60

AI訓練：透過資料尋找最佳函數，即最佳權重解

$$f(x) = w_1x_1 + w_2x_2 + w_3x_3 + w_0 = y$$


權重

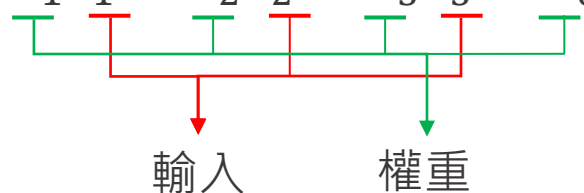
BMI ( $\hat{y}$ )	$y$
18.36	18.63
27.78	28.61
22.31	21.98
⋮	⋮
23.03	22.88
28.09	28.78



$$f(x) = -22.416 \times x_1 + 0.336 \times x_2 + 0.002 \times x_3 + 38.655$$

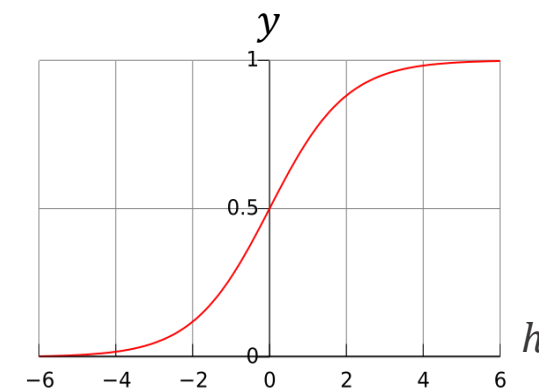
# 06 分類 機器學習判斷胖瘦

$$h(x) = w_1x_1 + w_2x_2 + w_3x_3 + w_0$$



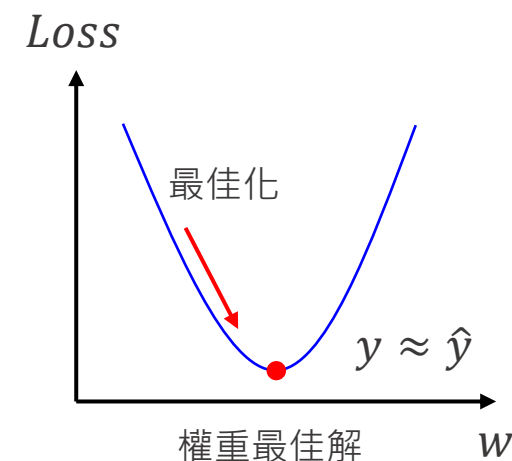
$$\text{Sigmoid}(h) = \frac{1}{1 + e^{-h}} = y$$

輸出



ID	身高 $x_1$	體重 $x_2$	年齡 $x_3$	胖1/瘦0 $\hat{y}$	$y$
1	160	47	22	0	0.1
2	180	90	32	1	0.9
3	157	55	18	0	0.2
⋮	⋮	⋮	⋮	⋮	⋮
999	168	65	21	0	0.2
1000	178	89	60	1	0.8

AI訓練：透過資料尋找最佳函數，即最佳權重解



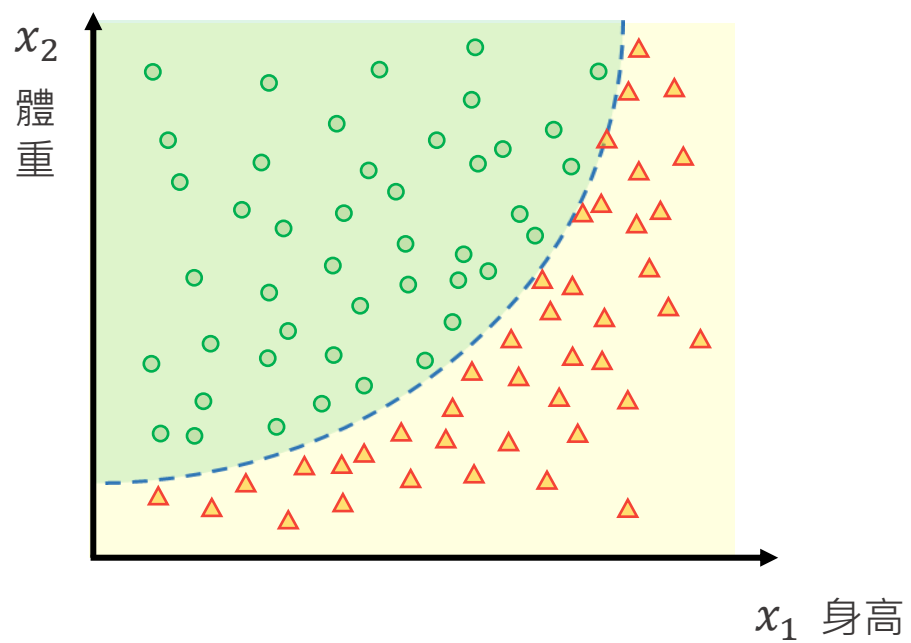
設置臨界值(Threshold)，若輸出 $y$ 大於該值則預測為1，反之，預測為0。通常設為0.5

## 06 分類 機器學習判斷胖瘦

☑ 2D散布圖：兩個輸入，函數為曲線或直線

$$h(x) = -22.416 \times x_1 + 0.336 \times x_2 + 38.655$$

$$\text{Sigmoid}(h) = 1 / (1 + e^{-h}) = y$$

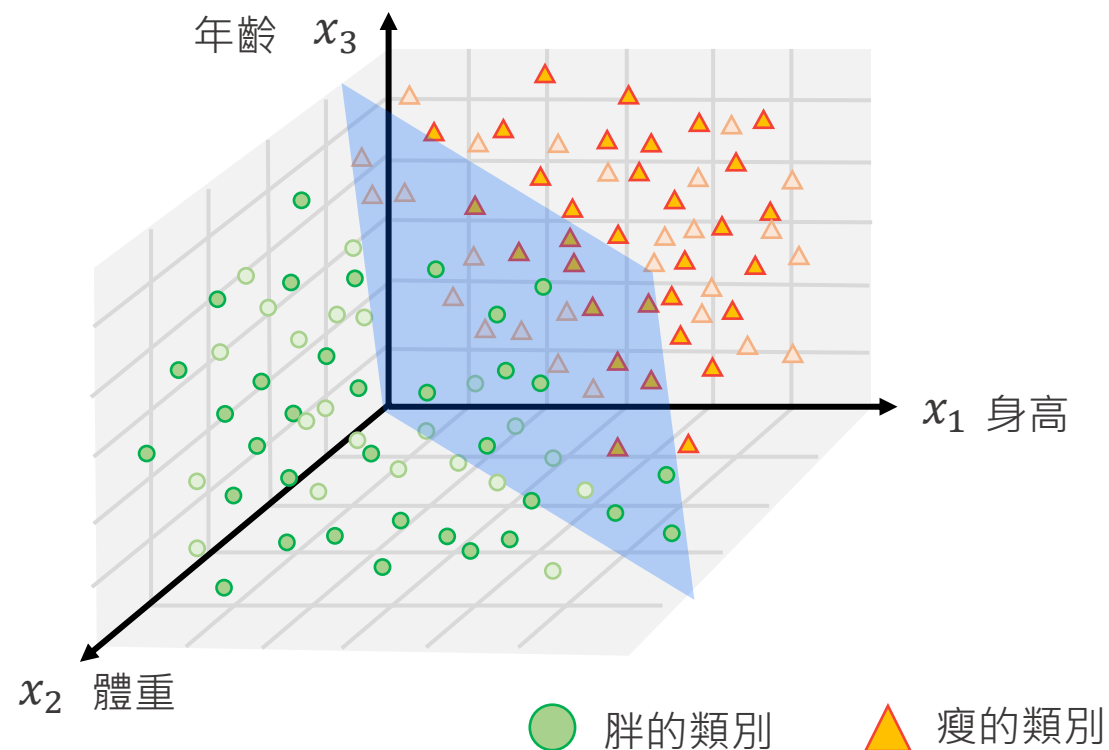


☑ 多維散布圖：多元輸入，函數為超曲面或超平面

☑ 3D散布圖：三個輸入，函數為曲面或平面

$$h(x) = -22.416 \times x_1 + 0.336 \times x_2 + 0.002 \times x_3 + 38.655$$

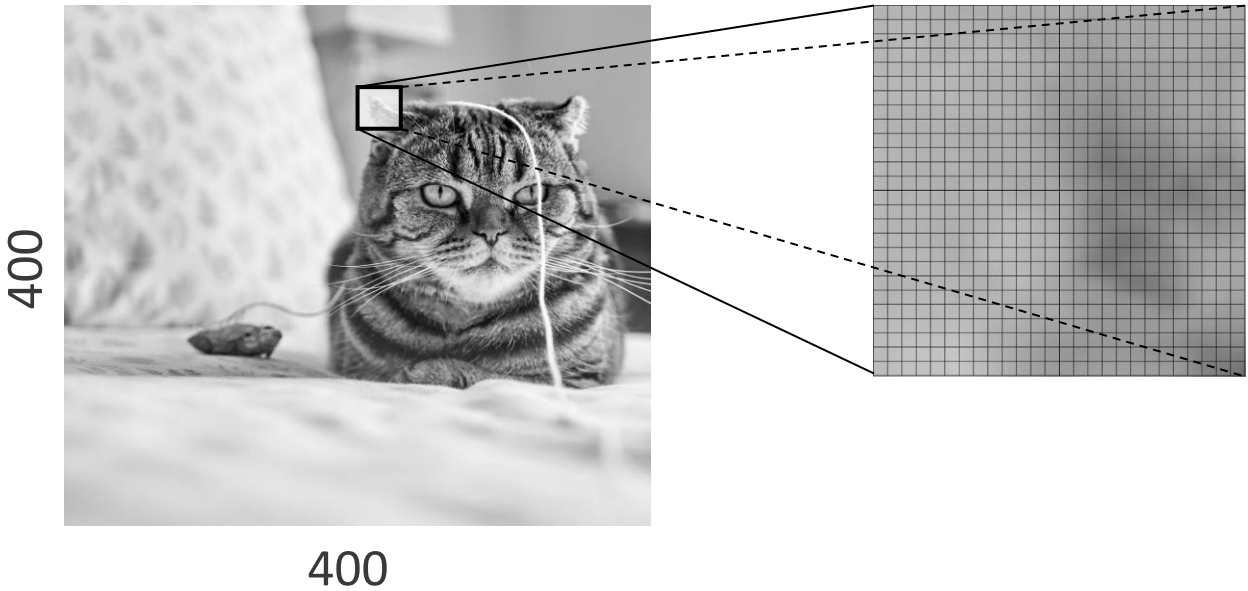
$$\text{Sigmoid}(h) = 1 / (1 + e^{-h}) = y$$



06

分類

機器學習判斷灰階影像



400

400

175	174	173	169	...	167
168	165	159	156	...	159
160	157	132	145	...	112
162	158	131	129	⋮	⋮
⋮	⋮	⋮	...	49	52
149	146	124	...	48	50

Image	Pixel 1	Pixel 2	Pixel 3	...	Pixel 400	Pixel 401	...	Pixel 159999	Pixel 160000	類別 $\hat{y}$
1	175	174	173	...	167	168	...	48	50	貓 0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮


AI訓練：透過資料尋找最佳函數，即最佳權重解

$f(x) = w_{160000}x_{160000} + \cdots + w_0 = h$

$Sigmoid(h) = \frac{1}{1 + e^{-h}} = y$



# 06 分類 機器學習判斷彩色影像



400

400

數位彩色影像每個像素值由RGB三個通道的數值所決定的

$$\begin{bmatrix} [175 & 160 & 175] & [174 & 174 & 174] & [173 & 173 & 173] & [169 & 169 & 179] & \dots & [167 & 167 & 167] \\ [168 & 165 & 167] & [165 & 165 & 175] & [159 & 109 & 189] & [156 & 156 & 156] & \dots & [159 & 169 & 159] \\ [160 & 120 & 150] & [157 & 157 & 147] & [132 & 122 & 132] & [145 & 145 & 125] & \dots & [112 & 111 & 122] \\ [162 & 102 & 80] & [158 & 158 & 148] & [131 & 100 & 131] & [129 & 102 & 119] & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ [149 & 255 & 100] & [146 & 255 & 152] & [124 & 124 & 124] & \dots & [49 & 169 & 200] & [52 & 252 & 252] \\ [48 & 148 & 255] & [50 & 250 & 240] & \dots & \dots & [48 & 148 & 255] & [50 & 250 & 240] \end{bmatrix}$$

400x400x3

$$\begin{bmatrix} 175 & 174 & 173 & 169 & \dots & 167 \\ 168 & 165 & 159 & 156 & \dots & 159 \\ 160 & 157 & 132 & 145 & \dots & 112 \\ 162 & 158 & 131 & 129 & \vdots & \vdots \\ \vdots & \vdots & \vdots & \dots & 49 & 52 \\ 149 & 146 & 124 & \dots & 48 & 50 \end{bmatrix}$$

400x400

**R**

$$\begin{bmatrix} 160 & 174 & 173 & 169 & \dots & 167 \\ 165 & 165 & 109 & 156 & \dots & 169 \\ 120 & 157 & 122 & 145 & \dots & 111 \\ 102 & 158 & 100 & 102 & \vdots & \vdots \\ \vdots & \vdots & \vdots & \dots & 169 & 252 \\ 255 & 255 & 124 & \dots & 148 & 250 \end{bmatrix}$$

400x400

**G**

$$\begin{bmatrix} 175 & 174 & 173 & 179 & \dots & 167 \\ 167 & 175 & 189 & 156 & \dots & 159 \\ 150 & 147 & 132 & 125 & \dots & 122 \\ 80 & 148 & 131 & 119 & \vdots & \vdots \\ \vdots & \vdots & \vdots & \dots & 200 & 252 \\ 100 & 152 & 124 & \dots & 255 & 240 \end{bmatrix}$$

400x400

**B**

1

純量 (Scalar)

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

向量 (Vector)

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

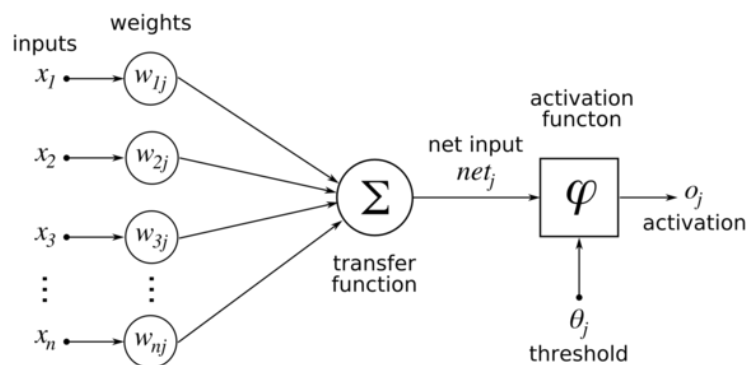
矩陣 (Matrix)

$$\begin{bmatrix} [1 & 2] & [2 & 3] \\ [3 & 7] & [4 & 5] \end{bmatrix}$$

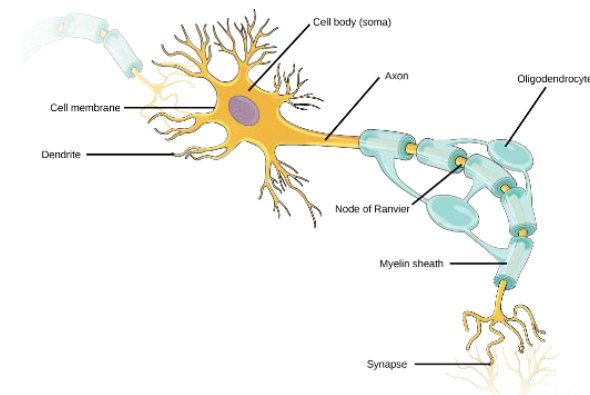
張量 (Tensor)

# 07 深度學習 – 神經元 與 神經網路

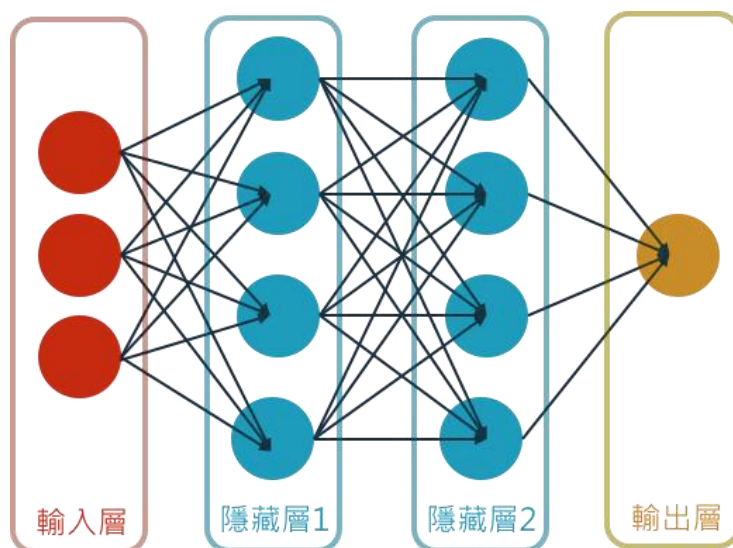
單層



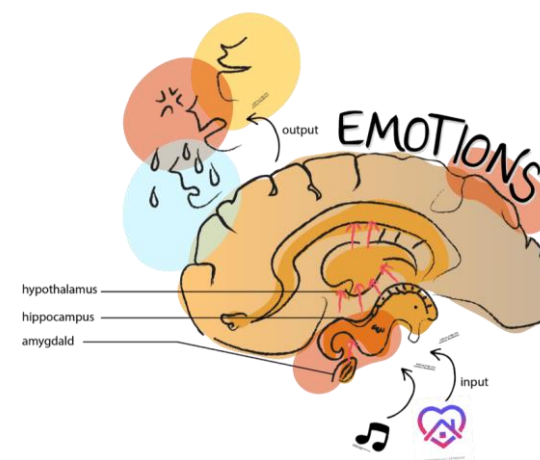
神經元



多層

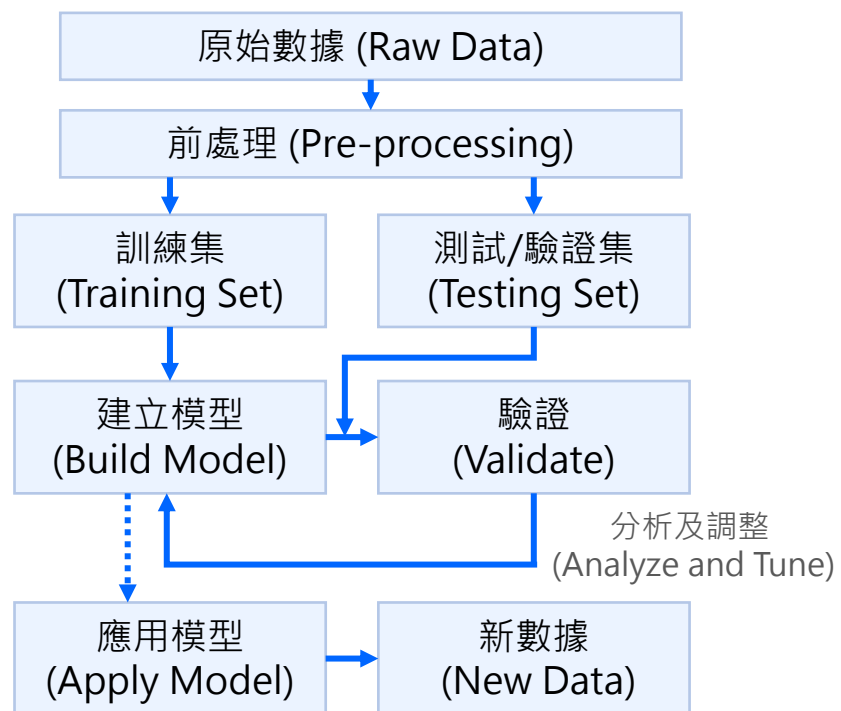


大腦



## 08 學習方法 – 監督式學習

### ☑ 監督式學習 (Supervised Learning)



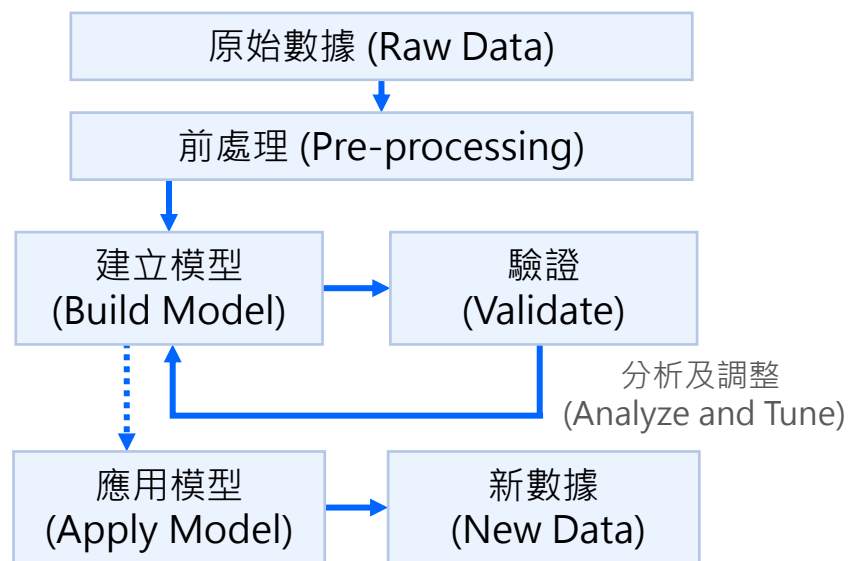
監督式學習是電腦從標籤化(Labeled)的資訊中分析模式後做出預測的學習方式

常見的監督式學習模型為：

1. 線性回歸
2. 邏輯回歸
3. 決策樹
4. 隨機森林
5. 支持向量機
6. 多層感知器
7. 卷積神經網路

## 08 學習方法 – 非監督式學習

### ☑ 非監督式學習 (Unsupervised Learning)



非監督式學習不需事先以人力處理標籤，機器面對資料時，做的處理是依照關聯性去歸類、找出潛在規則與套、形成集群

常見的非監督式學習模型為：

1. K平均演算法
2. 主成分分析
3. 孤獨森林
4. 單類別支持向量機
5. 自編碼器
6. 生成對抗網路

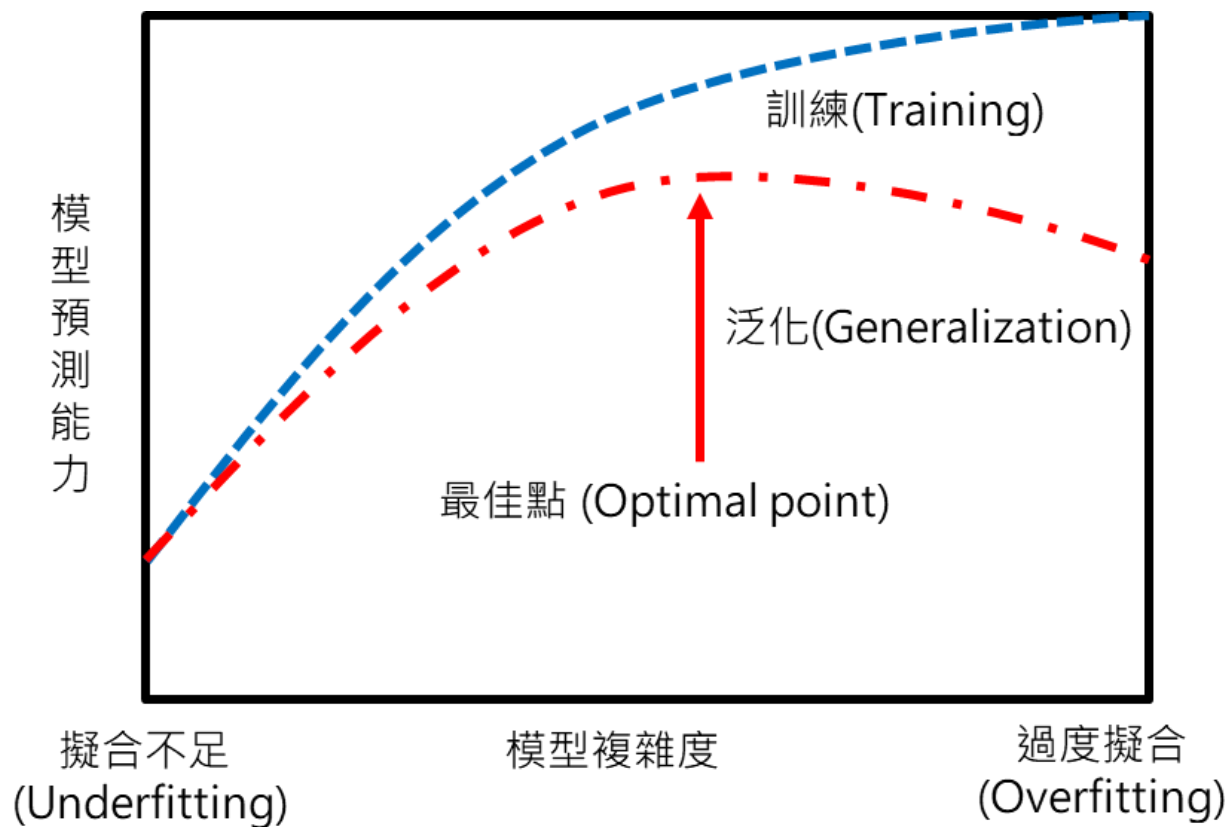


## 09 泛化能力



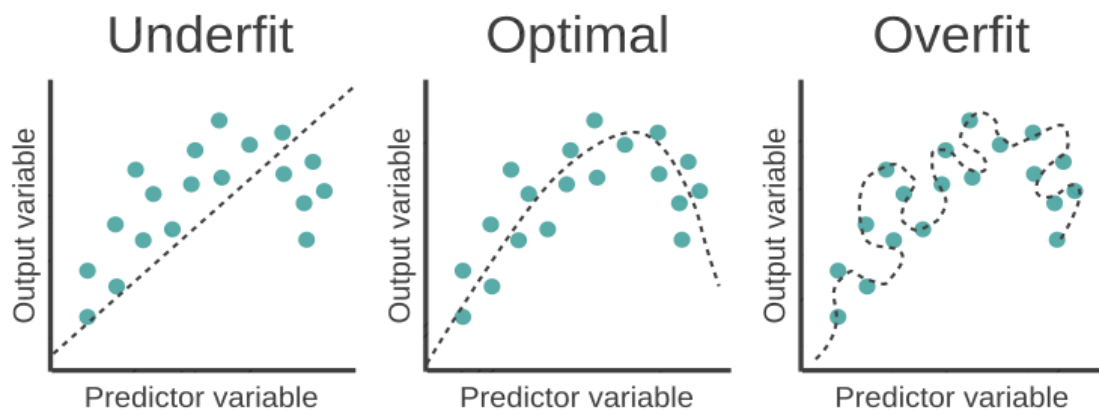
泛化能力(Generalization Ability)是指模型適應新數據的能力。以下方法可以提高泛化能力：

1. 資料樣本增大
2. 資料前處理
3. 調降低整模型**超參數**來模型複雜度

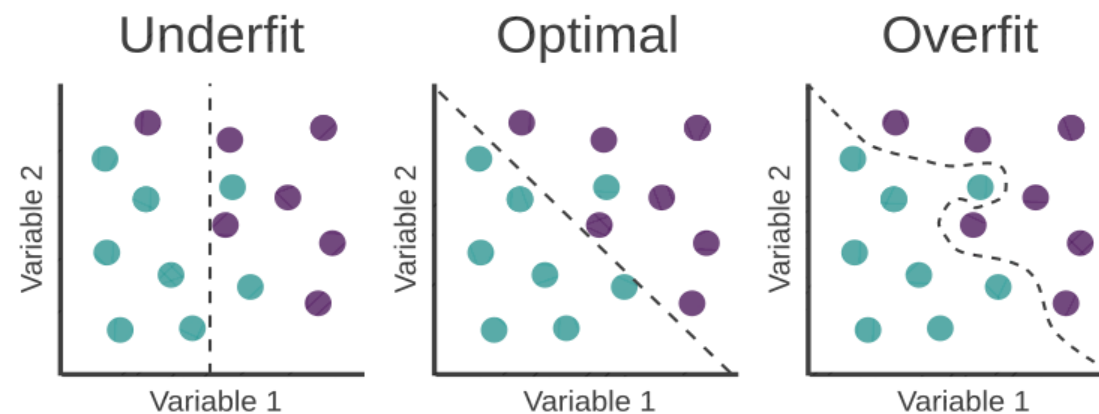


# 09 泛化能力


## ☑ 回歸



## ☑ 分類



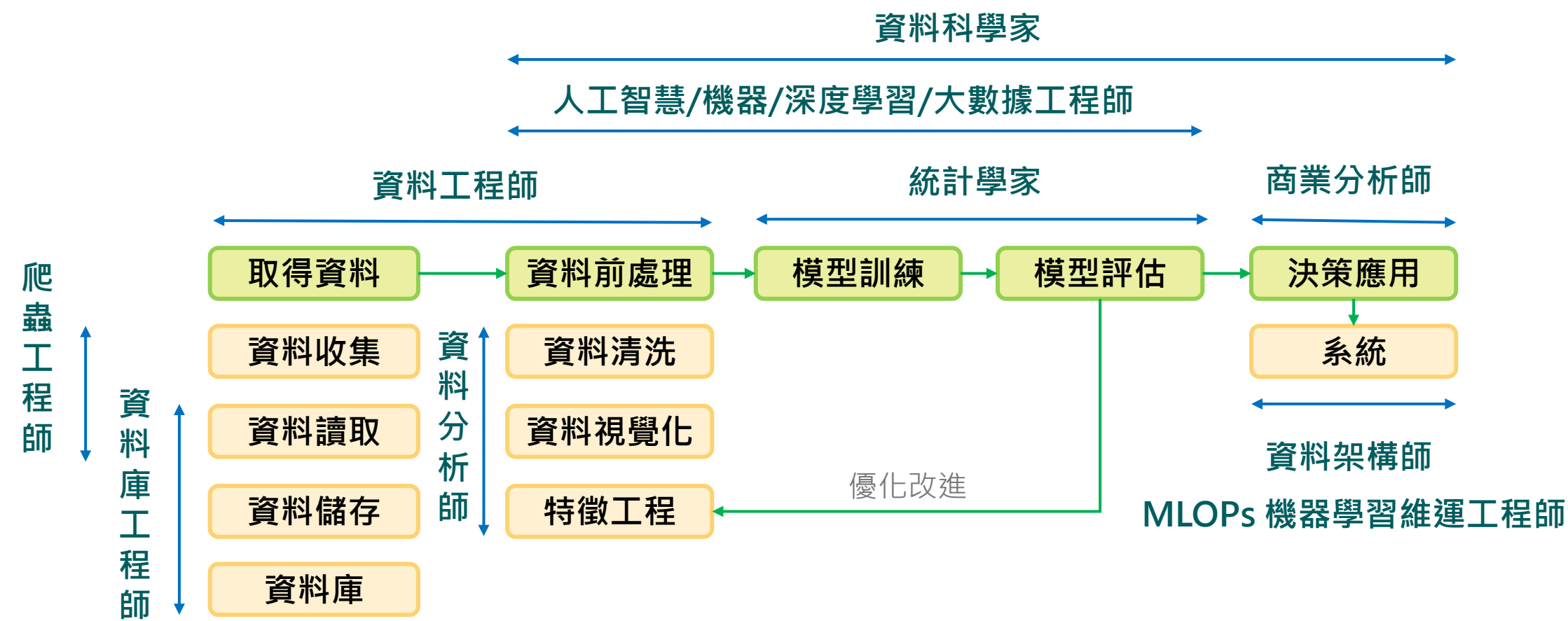
# 10 了解資料

 在數據分析與人工智慧模型訓練前的第一步，就是要了解資料與蒐集資料，目前常見的資料格式分為**結構化**、**半結構化**與**非結構化**資料：

	結構化資料 (Sturctured Data)	半結構化資料 (Semi-Structured Data)	非結構化資料 (Unstructured Data)
定義	嚴謹定義為資料可以被呈現在資料庫表格的行與欄，即已被整理過的資料	便於資料交換，其特性同時具備欄位概念與欄位可拓展性，可透過欄位查詢資料，並可根據使用者需求來增減欄位	形式自由且不遵循標準的格式規範，一團沒有組織的數據，即未經整理過的資料
優缺點	查詢資料快速，佔用存儲空間少；缺點是拓展新的欄位比較麻煩，在資料交換上的規定也比較嚴格	利於資料交換與傳輸，並可以增減欄位；缺點每筆資料的結構可能會不一致	佔用更多存儲空間，無法直接用於數據分析、未規則性的資料很難處理與整理
範例	關聯式資料庫(MySQL, Oracle等)的資料	CSV、JSON與XML	文字、圖片、音樂、影片、PDF等

- 人工智慧所訓練的資料為結構化的張量資料，如何將非結構化轉化為結構化資料是AI前的必要步驟

# 11 AI相關的工程師





PART

2

# 線性回歸

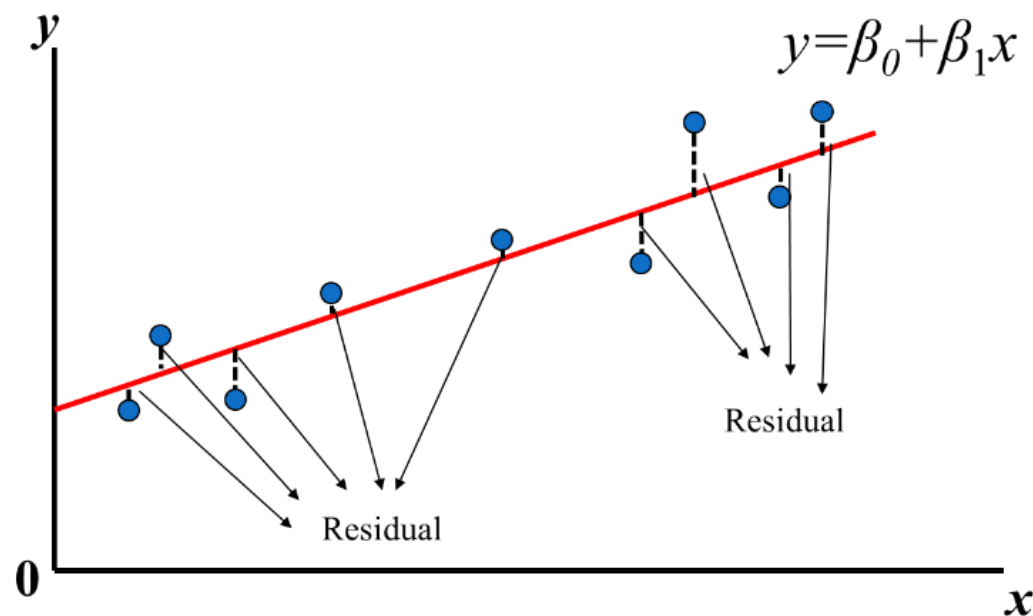
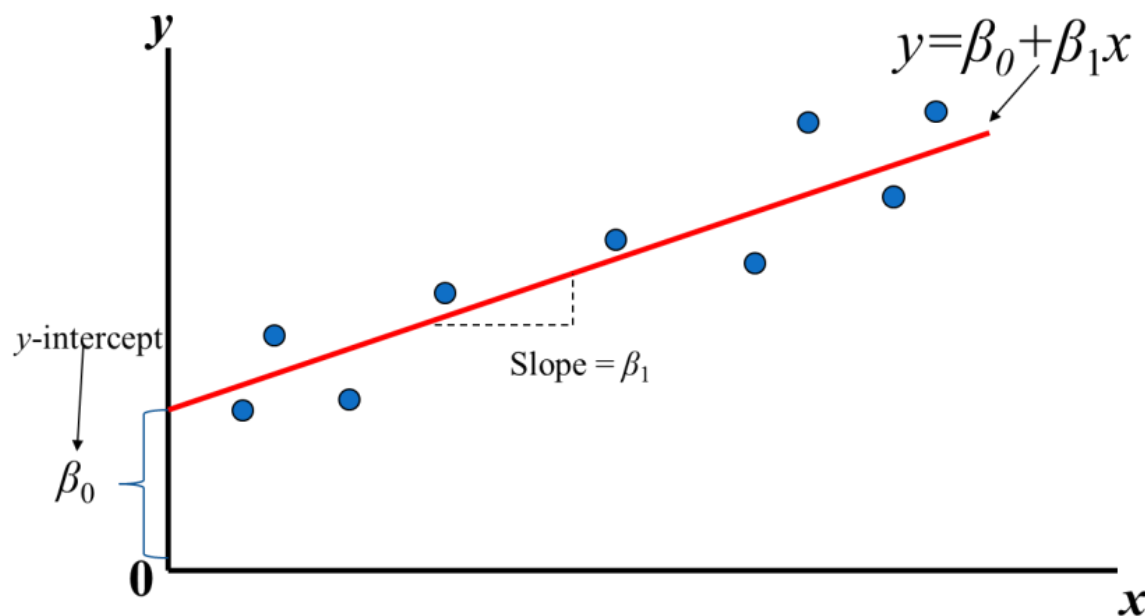
- 參數與超參數
- 損失函數
- 梯度下降法
- 學習速率
- 批量、隨機與小批量梯度下降法



# 01 參數與超參數



統計上在找多個自變數和依變數之間的關係。訓練過程是從訓練集中以梯度下降法來確定權重與偏差



- 參數(Parameter)：模型從數據中可以自動學習出的變量。例如，權重(Weights)，偏差(Bias)，即變數
- 超參數(Hyper-parameter)：確定模型的一些數值，此數值不同則模型預測能力也會不同的。超參數的數值可根據經驗確定的變量，或其他搜索演算法來決定

## 02 損失函數



損失函數(Loss Function)又稱為**目標函數**與**成本函數**，是用來估量模型的預測值 $f(x)$ 與真實值的不一致程度。機器學習的目的是找出一個預測函數，並透過設計一個目標函數能透過學習降低誤差，降低誤差是指預測值越接近真實值。

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

$n$  為資料數量， $Y^{(i)}$ 表示實際值， $y^{(i)}$ 表示預測值

# 03 梯度下降法



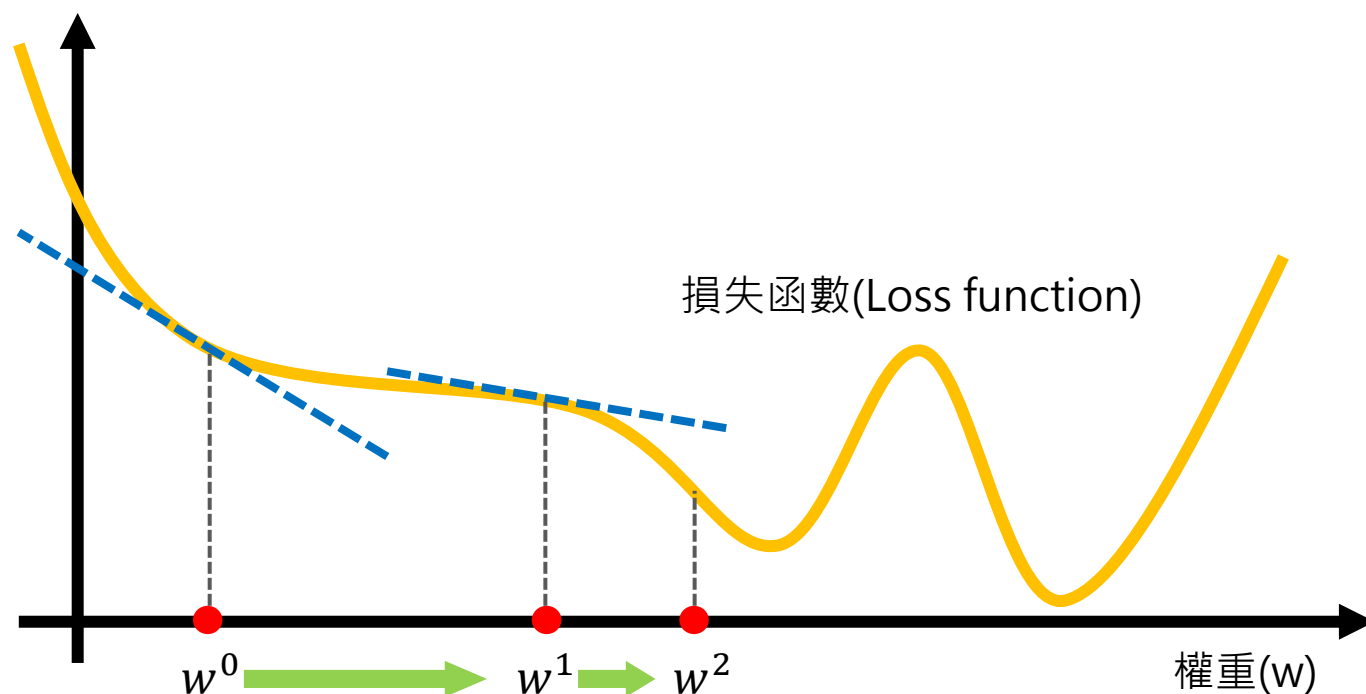
梯度下降法(Gradient descent)為常見的最佳優化方法之一，利用一階微分來逐步找尋(損失)函數的最低點，因此使用上此函數是能可被一階微分。迭代次數(Number of Iterations)為超參數代表訓練所需更新權重的次數

1. 隨機選擇一個初始值  $w^0$

2. 計算  $\frac{dL}{dw} \Big|_{w=w^0}$   $w^0 - \eta \frac{dL}{dw} \Big|_{w=w^0} \rightarrow w^1$

3. 計算  $\frac{dL}{dw} \Big|_{w=w^1}$   $w^1 - \eta \frac{dL}{dw} \Big|_{w=w^1} \rightarrow w^2$

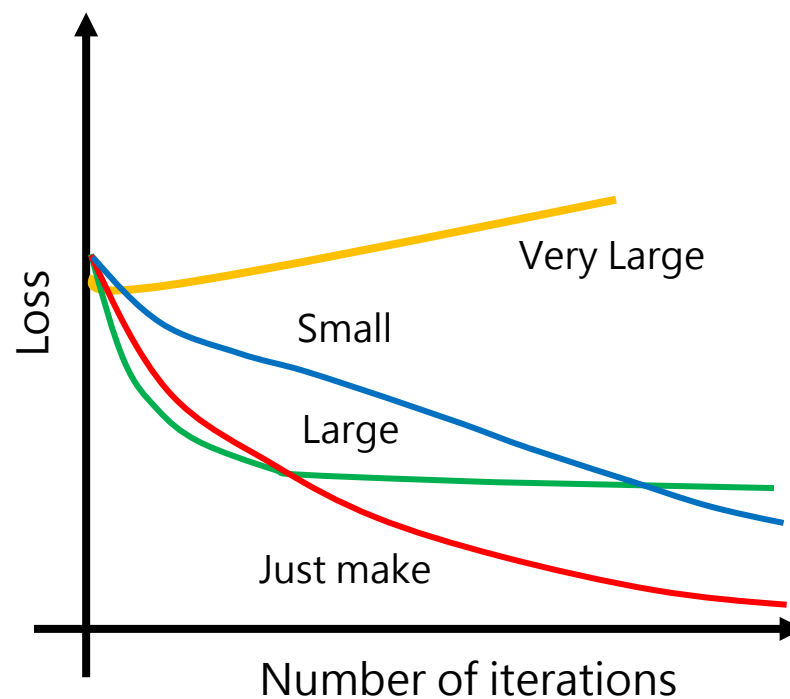
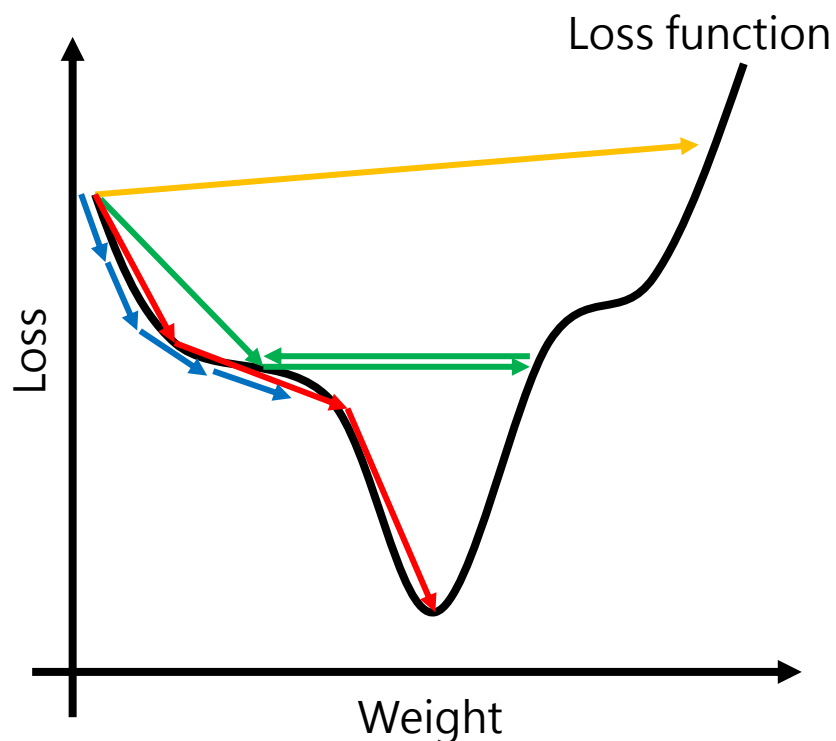
其中  $\eta$  為學習速率(Learning Rate)  
是一個超參數





## 04 學習速率

☑ 學習速率(Learning Rate)是梯度下降法的超參數，會影響損失函數收斂至最低點

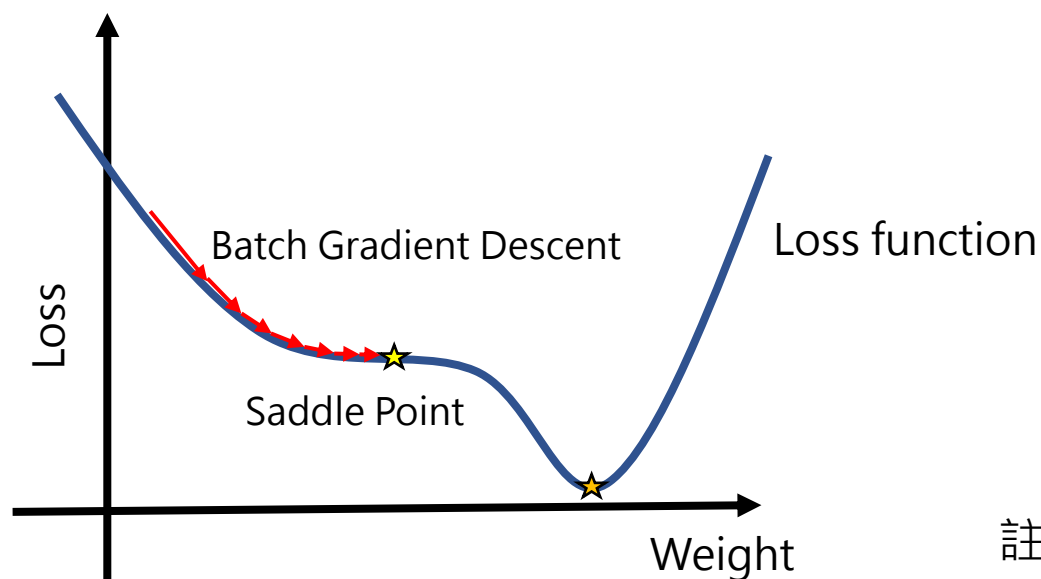


$$w^i = w^{i-1} - \eta \left. \frac{dL}{dw} \right|_{w = w^{i-1}}$$

$$\Delta w_k = -\eta \frac{\partial L}{\partial w_k}$$

## 05 批量梯度下降法

☑ 批量梯度下降法(Batch Gradient Descent)是運用**所有資料(訓練集)**來計算誤差曲面，即算出損失函數當下的斜率，並更新權重。缺點為誤差曲面(損失函數)出現「鞍點」(Saddle Point)，此時微分等於零，導致梯度下降法卡住



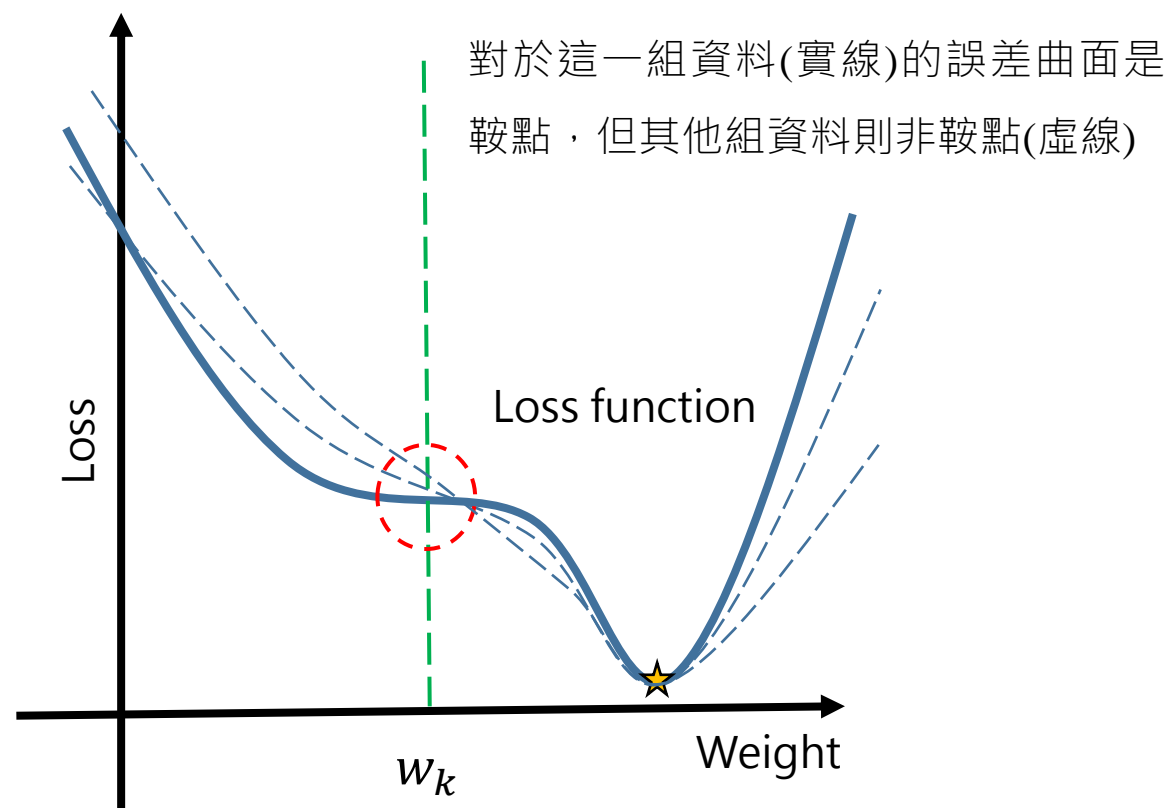
註：一個不是局部極值點的駐點稱為**鞍點**

## 06 隨機梯度下降法



隨機梯度下降法 (Stochastic Gradient Descent, SGD) 是每次只**隨機取樣一組資料** (訓練集內的任一個樣本) 來計算誤差曲面，即算出損失函數當下的斜率來更新權重

- 優點：可以避免限於鞍點
- 缺點：每次只看一組資料，若當下的誤差曲面不是很恰當的，權重更新可能有倒回去的情況發生，會導致梯度下降過程耗費更時間



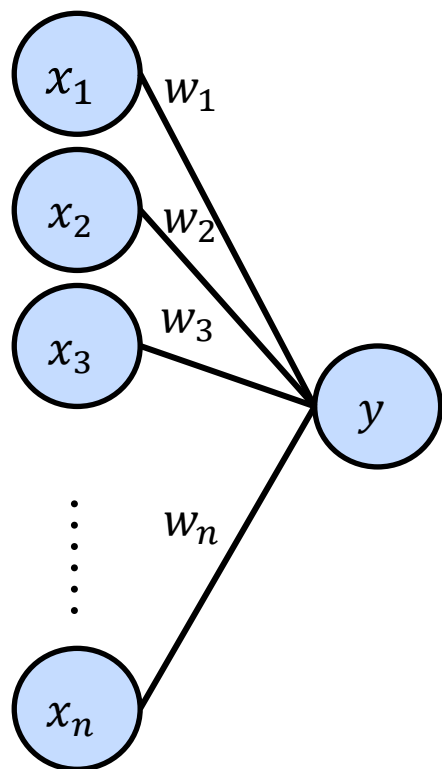
## 06 小批量梯度下降法



小批量梯度下降法(Mini-Batch Gradient Descent)是每次迭代都會從所有資料中取出一小部分(非單筆)資料來推算誤差曲面，小批量資料取樣的多寡，它又是一個超參數(Hyper-parameters)



# 06 權重更新推導



■ 線性回歸模型：

$$y = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n$$

$$\text{Loss function} = \frac{1}{2} \sum_{i=1}^n (Y^{(i)} - y^{(i)})^2$$

其中 $Y^{(i)}$ 是實際值， $y^{(i)}$ 是第 $i$ 個樣本的輸出， $n$ 為訓練集批量數量。

■ 權重更新：

$$\begin{aligned} \Delta w_k &= -\eta \frac{\partial L}{\partial w_k} = -\eta \frac{\partial}{\partial w_k} \left[ \frac{1}{2} \sum_{i=1}^n (Y^{(i)} - y^{(i)})^2 \right] \\ &= (-\eta) \times \frac{1}{2} \times 2 \times \sum_{i=1}^n [(Y^{(i)} - y^{(i)}) \left( \frac{-\partial y^{(i)}}{\partial w_k} \right)] \\ &= \sum_{i=1}^n [\eta (Y^{(i)} - y^{(i)}) \left( \frac{\partial y^{(i)}}{\partial w_k} \right)] \\ &= \sum_{i=1}^n [\eta x_k^{(i)} (Y^{(i)} - y^{(i)})] \end{aligned}$$



# 3

## PART

# 邏輯回歸



- 單變數二元分類
  - 範例：鳶尾花
- 單變數的三元分類
  - 邏輯函數
- 多變數的多元分類
  - 損失函數：交叉熵
  - 正則化

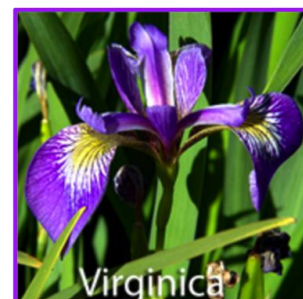
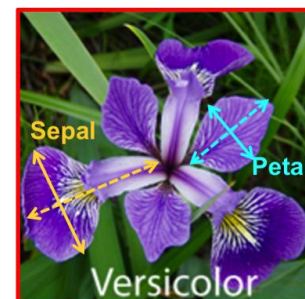
# 01 鳶尾花資料集

☑ 鳶尾花資料集(Iris Dataset) 取自美國加州大學歐文分校的機器學習資料庫

<http://archive.ics.uci.edu/ml/datasets/Iris>，資料的筆數為150筆：

特徵 (Feature)：

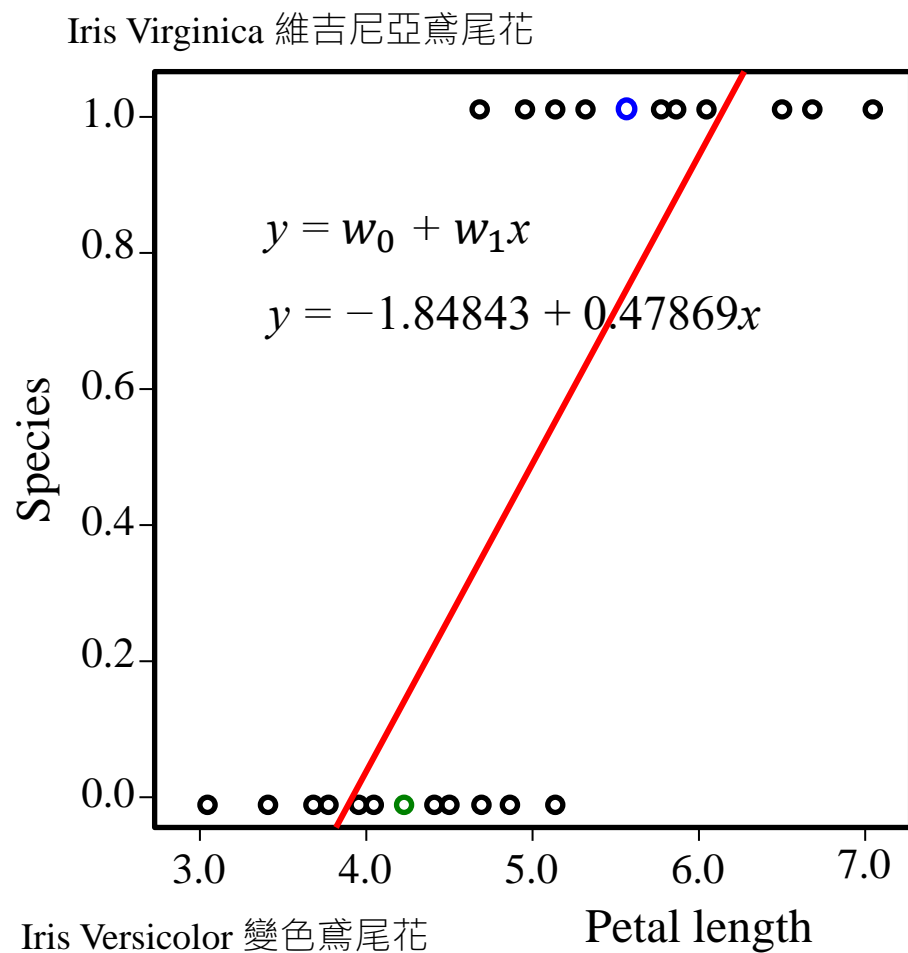
1. 花萼長度 (Sepal Length) (cm)
2. 花萼寬度 (Sepal Width) (cm)
3. 花瓣長度 (Petal Length) (cm)
4. 花瓣寬度 (Petal Width) (cm)



三個品種 - 類別/標籤 (Class/Label)：

1. 山鳶尾(Setosa)
2. 變色鳶尾(Versicolor)
3. 維吉尼亞鳶尾(Virginica)

## 02 單變數二元分類



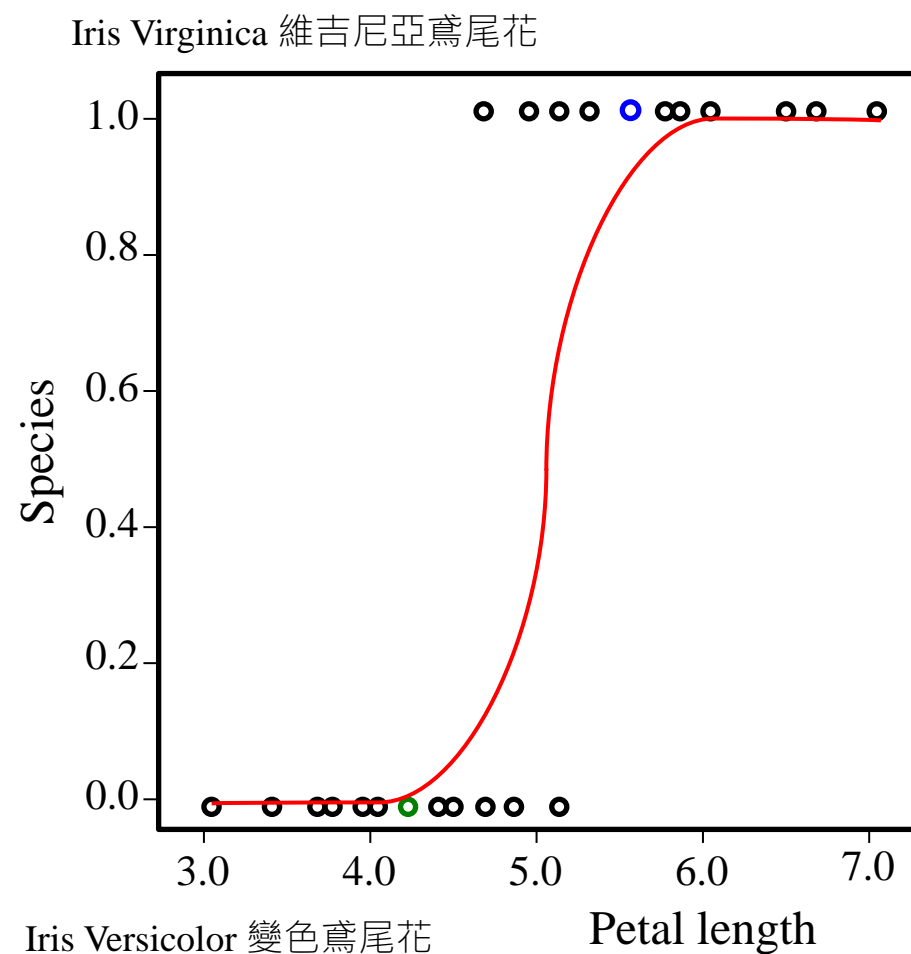
### 線性回歸作分類任務時

	$x$	$y$
	Petal length	Species
113	5.5	維吉尼亞 Virginica (1)
98	4.3	變色 Versicolor (0)
121	5.7	維吉尼亞 Virginica (1)
52	4.5	變色 Versicolor (0)
58	3.3	變色 Versicolor (0)
139	4.8	維吉尼亞 Virginica (1)
94	3.3	變色 Versicolor (0)
⋮	⋮	⋮



- 可以設置臨界值進行分類，但缺點是輸出為任一大小值

## 02 單變數二元分類



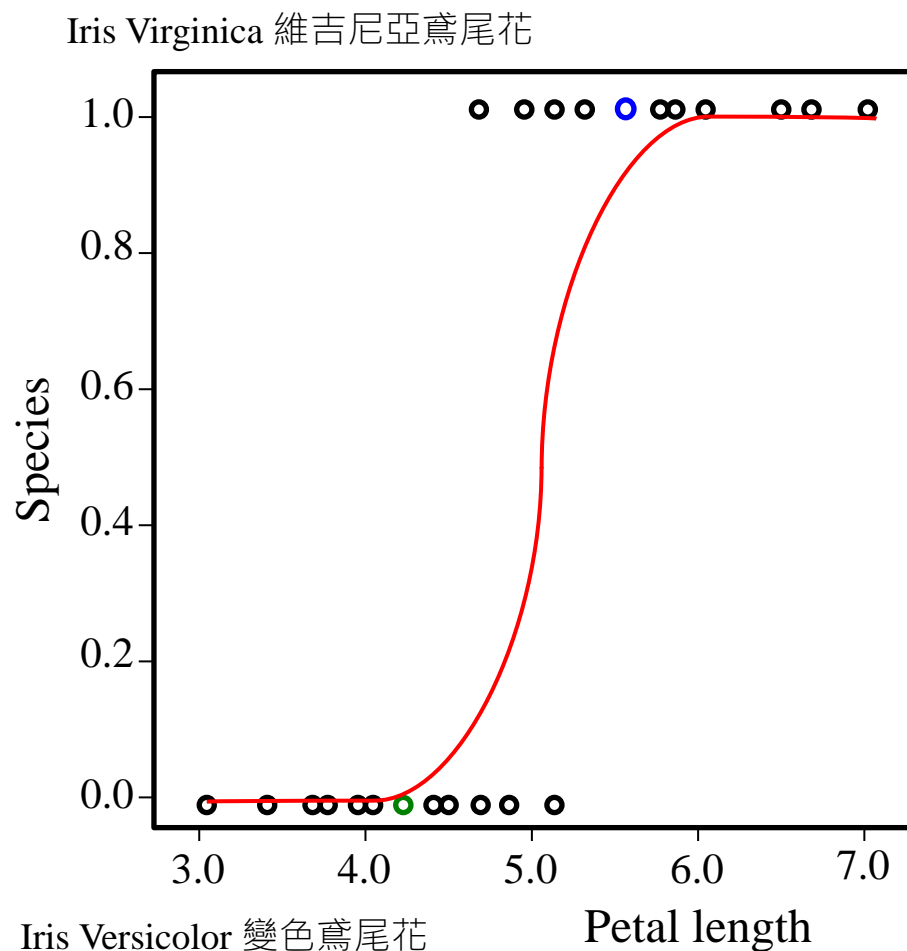
邏輯回歸作分類任務時

	$x$	$y$
	Petal length	Species
113	5.5	維吉尼亞 Virginica (1)
98	4.3	變色 Versicolor (0)
121	5.7	維吉尼亞 Virginica (1)
52	4.5	變色 Versicolor (0)
58	3.3	變色 Versicolor (0)
139	4.8	維吉尼亞 Virginica (1)
94	3.3	變色 Versicolor (0)
⋮	⋮	⋮



- 輸出可作為機率，介於0至1之間的值

## 02 單變數二元分類

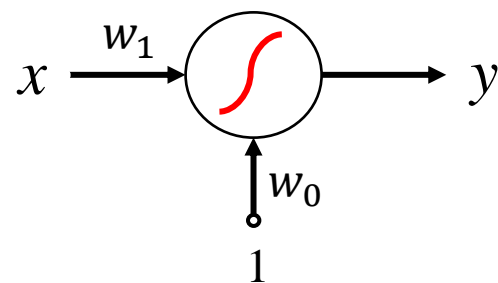


### ☑ 邏輯/乙狀函數 (Logistic / Sigmoid Function)

線性回歸  $t = w_0 + w_1x$

邏輯回歸  $y = f(t) = f(w_0 + w_1x)$

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



$$\Pr(Y=1 | x) = \frac{e^{w_0 + w_1x}}{1 + e^{w_0 + w_1x}} = f(w_0 + w_1x)$$

## 02 單變數二元分類

$$\Pr(Y=1 | x) = \frac{e^{w_0 + w_1 x}}{1 + e^{w_0 + w_1 x}} = f(w_0 + w_1 x)$$

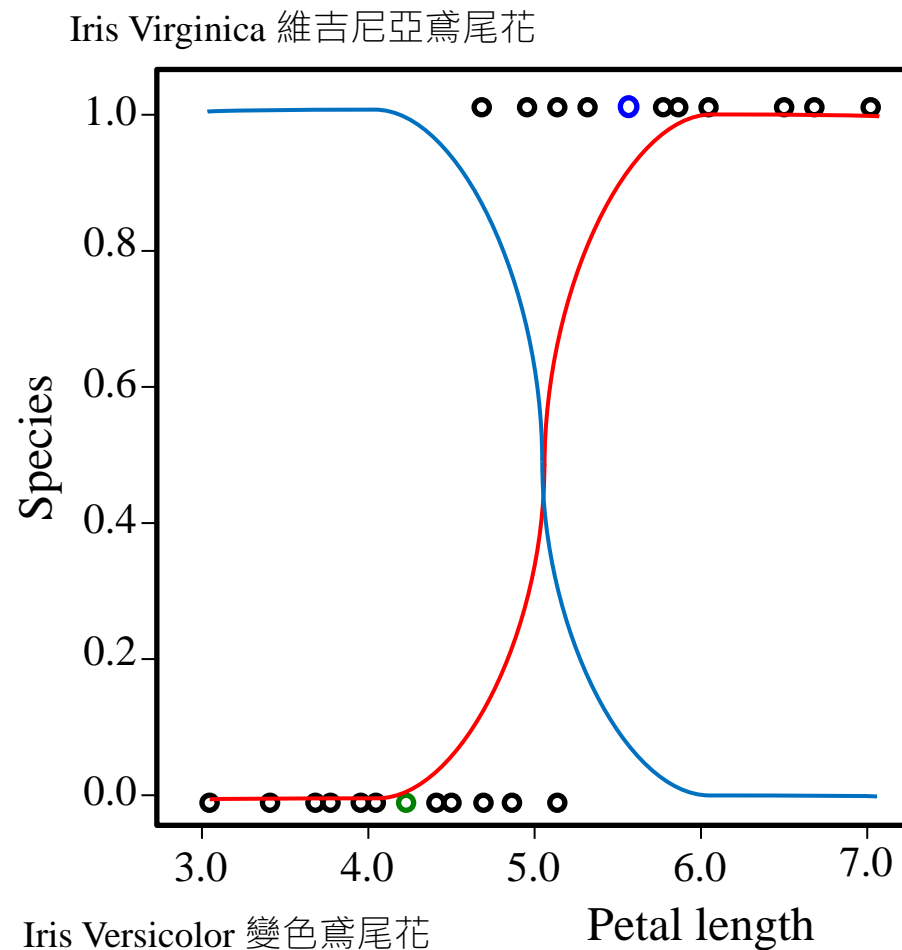
$$\Pr(Y=0 | x) = 1 - \Pr(Y=1 | x)$$

$$= 1 - \frac{e^{w_0 + w_1 x}}{1 + e^{w_0 + w_1 x}}$$

$$= \frac{1}{1 + e^{w_0 + w_1 x}}$$

$$\Pr(Y=1 | x) = \frac{e^{w_0 + w_1 x}}{1 + e^{w_0 + w_1 x}}$$

$$\Pr(Y=0 | x) = \frac{1}{1 + e^{w_0 + w_1 x}}$$





## 02 單變數二元分類

$$\Pr(Y=1 | x) = \frac{e^{w_0 + w_1 x}}{1 + e^{w_0 + w_1 x}} \quad \Pr(Y=0 | x) = \frac{1}{1 + e^{w_0 + w_1 x}}$$

$$\Pr(Y=y_{113}=1 | x_{113}) = \frac{e^{w_0 + w_1 x_{113}}}{1 + e^{w_0 + w_1 x_{113}}}$$

$$\Pr(Y=y_{98}=0 | x_{98}) = \frac{1}{1 + e^{w_0 + w_1 x_{98}}}$$

$$\Pr(Y=y_{121}=1 | x_{121}) = \frac{e^{w_0 + w_1 x_{121}}}{1 + e^{w_0 + w_1 x_{121}}}$$

$$\Pr(Y=y_{52}=0 | x_{52}) = \frac{1}{1 + e^{w_0 + w_1 x_{52}}}$$

$$\Pr(Y=y_{58}=1 | x_{58}) = \frac{e^{w_0 + w_1 x_{58}}}{1 + e^{w_0 + w_1 x_{58}}}$$

$$\Pr(Y=y_{139}=0 | x_{139}) = \frac{1}{1 + e^{w_0 + w_1 x_{139}}}$$

$$\Pr(Y=y_{94}=1 | x_{94}) = \frac{e^{w_0 + w_1 x_{94}}}{1 + e^{w_0 + w_1 x_{94}}}$$

	x	y
	Petal length	Species
113	5.5	維吉尼亞 Virginica (1)
98	4.3	變色 Versicolor (0)
121	5.7	維吉尼亞 Virginica (1)
52	4.5	變色 Versicolor (0)
58	3.3	變色 Versicolor (0)
139	4.8	維吉尼亞 Virginica (1)
94	3.3	變色 Versicolor (0)
⋮	⋮	⋮

## 02 單變數二元分類

☑ 最大似然函數 (Maximum Likelihood Function, MLF)

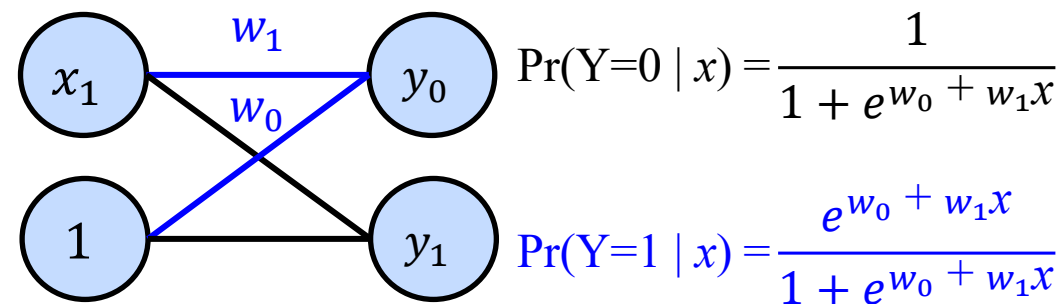
$$\begin{aligned} P(w_0, w_1) &= \prod_{y_i=1} \Pr(Y=y_i=1 \mid x_i) \prod_{y_i=0} \Pr(Y=y_i=0 \mid x_i) \\ &= \prod_{i=1}^n \Pr(Y=y_i=1 \mid x_i)^{y_i} \Pr(Y=y_i=0 \mid x_i)^{1-y_i} \\ &= \prod_{i=1}^n \Pr(Y=y_i=1 \mid x_i)^{y_i} (1 - \Pr(Y=y_i=1 \mid x_i))^{1-y_i} \\ \Rightarrow P &= \prod_{i=1}^n p(x_i)^{y_i} (1 - p(x_i))^{1-y_i} \end{aligned}$$



## 02 單變數二元分類

$$\rightarrow P = \prod_{i=1}^n p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}$$

$$\rightarrow -\log(P) = -\sum_{i=1}^n [y_i \log(p(x_i)) + (1 - y_i) \log(1 - p(x_i))]$$



### ☒ 二元交叉熵損失函數

$$\text{Binary Cross Entropy Loss} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(p(x_i)) + (1 - y_i) \log(1 - p(x_i))]$$

## 02 單變數多元分類

### ☑ 勝算比(Odds Ratio, OR)

#### 單變數的二元分類

$$\Pr(Y=1 \mid x) = \frac{e^{w_0 + w_1 x}}{1 + e^{w_0 + w_1 x}}$$

$$\Pr(Y=0 \mid x) = \frac{1}{1 + e^{w_0 + w_1 x}}$$

$$\text{Odds Ratio} = \frac{\Pr(Y=1 \mid x)}{\Pr(Y=0 \mid x)} = e^{w_0 + w_1 x}$$

#### 單變數的三元分類

$$\text{Odds Ratio} = \frac{\Pr(Y=1 \mid x)}{\Pr(Y=0 \mid x)} = e^{w_{10} + w_{11}x}$$

$$\text{Odds Ratio} = \frac{\Pr(Y=2 \mid x)}{\Pr(Y=0 \mid x)} = e^{w_{20} + w_{21}x}$$

## 02 單變數多元分類

$$\text{Odds Ratio} = \frac{\Pr(Y=1 | x)}{\Pr(Y=0 | x)} = e^{w_{10} + w_{11}x} \rightarrow$$

$$\text{Odds Ratio} = \frac{\Pr(Y=2 | x)}{\Pr(Y=0 | x)} = e^{w_{20} + w_{21}x} \rightarrow$$

$$\Pr(Y=1 | x) = \Pr(Y=0 | x) (e^{w_{10} + w_{11}x})$$

$$\Pr(Y=2 | x) = \Pr(Y=0 | x) (e^{w_{20} + w_{21}x})$$

代入

$$\Pr(Y=0 | x) + \Pr(Y=1 | x) + \Pr(Y=2 | x) = 1$$

$$\rightarrow \Pr(Y=0 | x) + \Pr(Y=0 | x) (e^{w_{10} + w_{11}x}) + \Pr(Y=0 | x) (e^{w_{20} + w_{21}x}) = 1$$

$$\rightarrow \Pr(Y=0 | x) (1 + e^{w_{10} + w_{11}x} + e^{w_{20} + w_{21}x}) = 1$$

$$\rightarrow \Pr(Y = 0 | x) = \frac{1}{1 + e^{w_{10} + w_{11}x} + e^{w_{20} + w_{21}x}}$$

$$\Pr(Y = 1 | x) = \frac{e^{w_{10} + w_{11}x}}{1 + e^{w_{10} + w_{11}x} + e^{w_{20} + w_{21}x}}$$

$$\Pr(Y = 2 | x) = \frac{e^{w_{20} + w_{21}x}}{1 + e^{w_{10} + w_{11}x} + e^{w_{20} + w_{21}x}}$$

代回去

## 02 單變數多元分類

$\Pr(\text{Species} = \text{Setosa}(\text{山鳶尾花}) | x)$

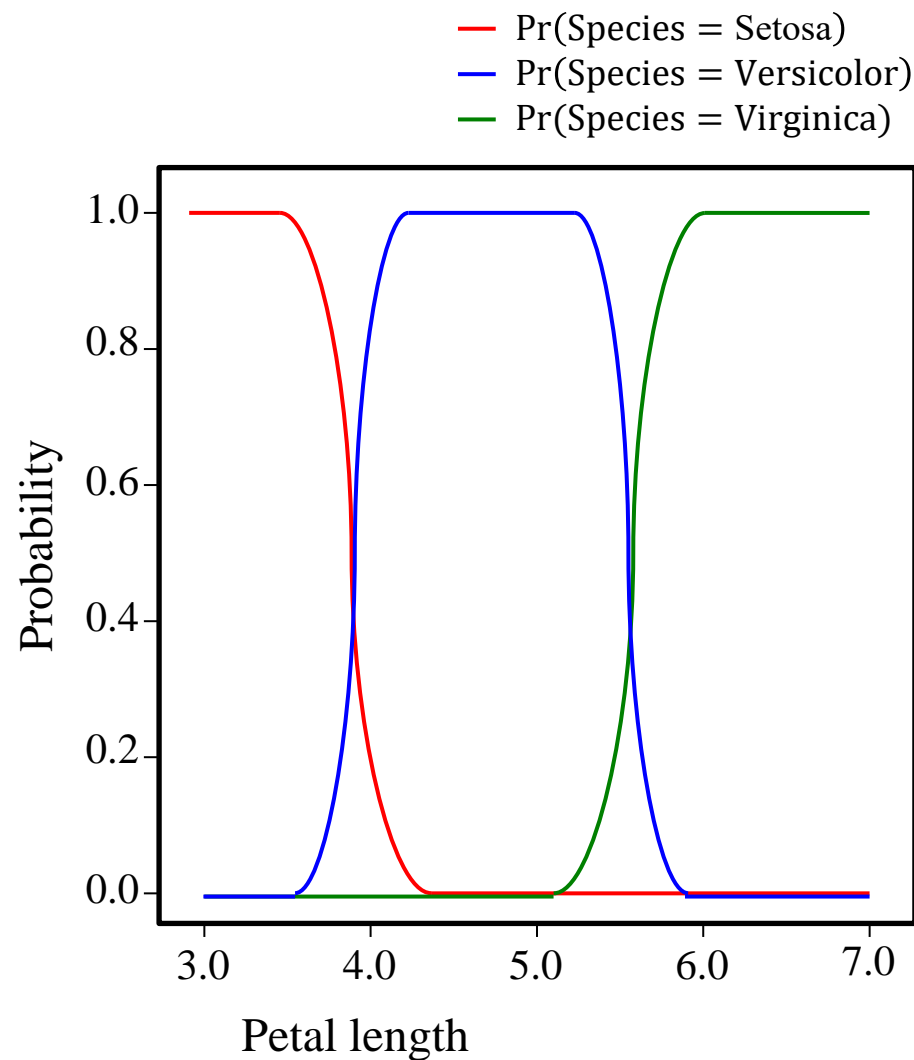
$$= \frac{e^{116.614 - 38.5797x}}{1 + e^{116.614 - 38.5797x} + e^{43.7809 - 9.0020x}}$$

$\Pr(\text{Species} = \text{Versicolor}(\text{變色鳶尾花}) | x)$

$$= \frac{e^{43.7809 - 9.0020x}}{1 + e^{116.614 - 38.5797x} + e^{43.7809 - 9.0020x}}$$

$\Pr(\text{Species} = \text{Virginica}(\text{維吉尼亞鳶尾花}) | x)$

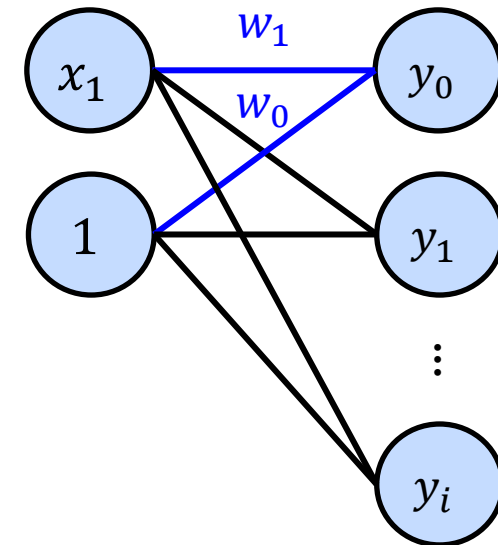
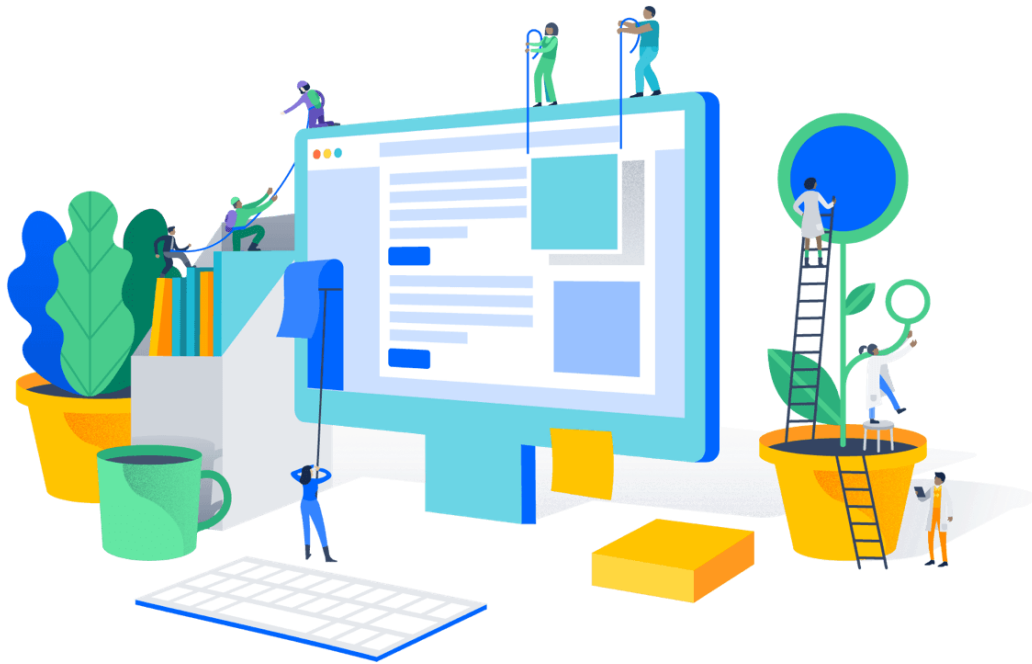
$$= \frac{1}{1 + e^{116.614 - 38.5797x} + e^{43.7809 - 9.0020x}}$$



## 02 單變數多元分類

### ☑ 二元交叉熵損失函數

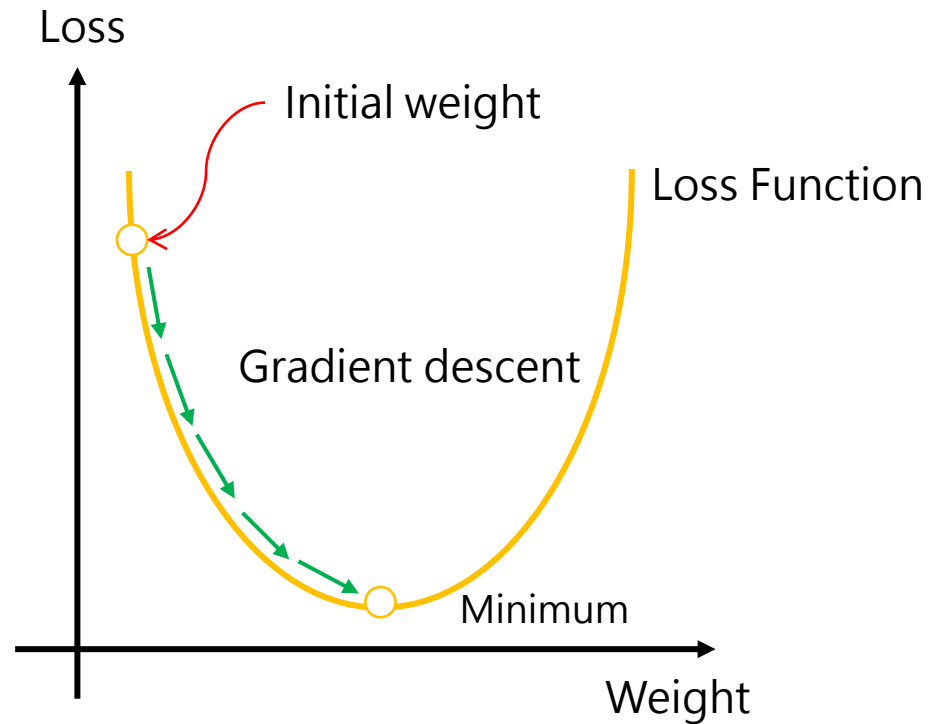
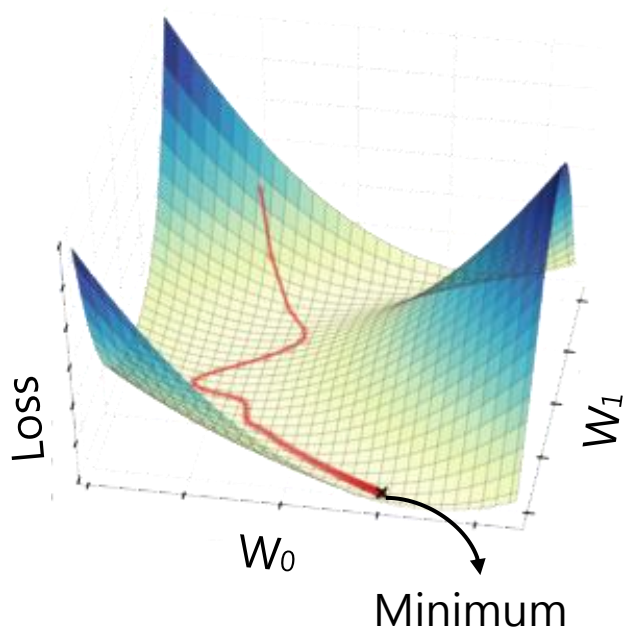
$$\text{Categorical Cross-Entropy Loss} = - \sum_{i=1}^n y_i \log(p(x_i))$$





# 03 正則化

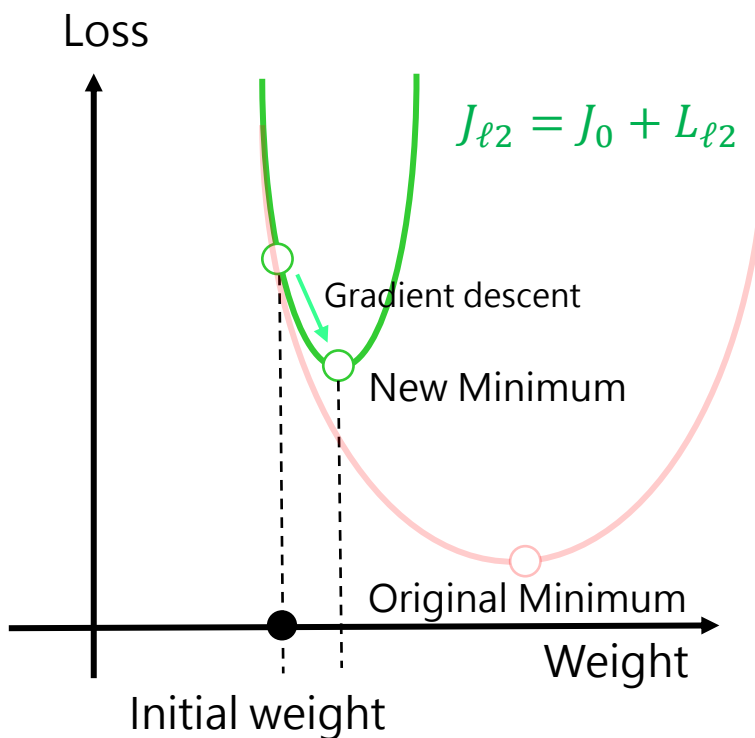
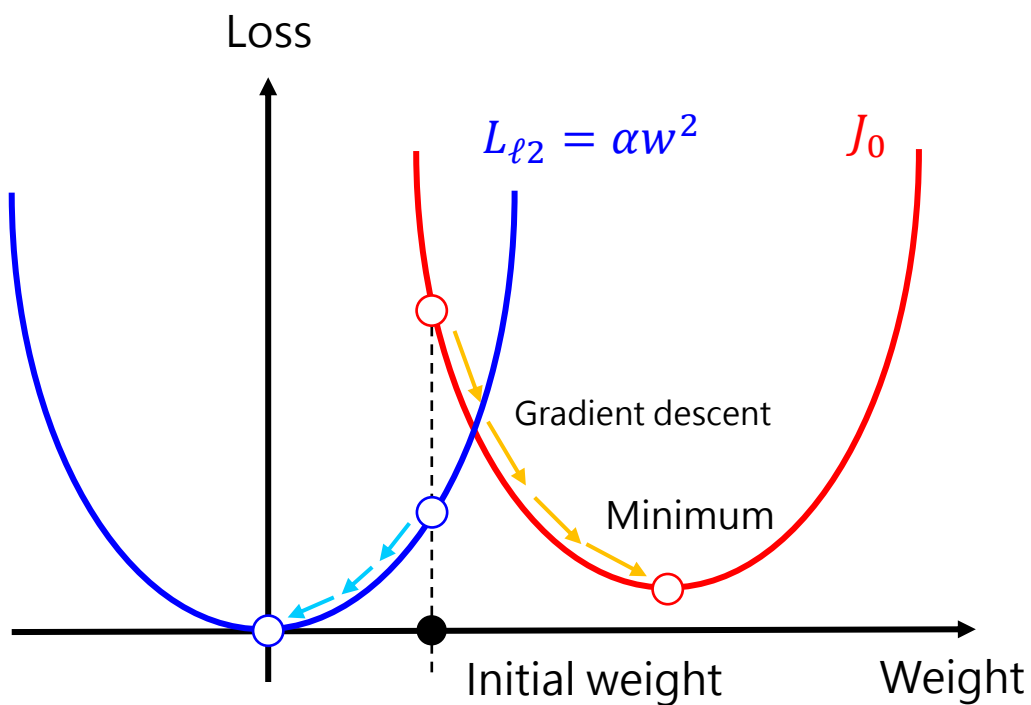
☑ 正則化 (Regularization) 是透過損失函數的改變，以避免下降至最低點，來預防過度擬和問題



# 03 正則化

☑  $J_{\ell 2} = J_0 + L_{\ell 2} = J_0 + \alpha \sum_{i=0}^n w_i^2$  其中  $J_0$  為原始損失函數， $L_{\ell 2}$  為正則化項

下圖假設只有一個權重時， $J_{\ell 2} = J_0 + L_{\ell 2} = J_0 + \alpha w^2$



PART

1

# 人工智慧安裝篇

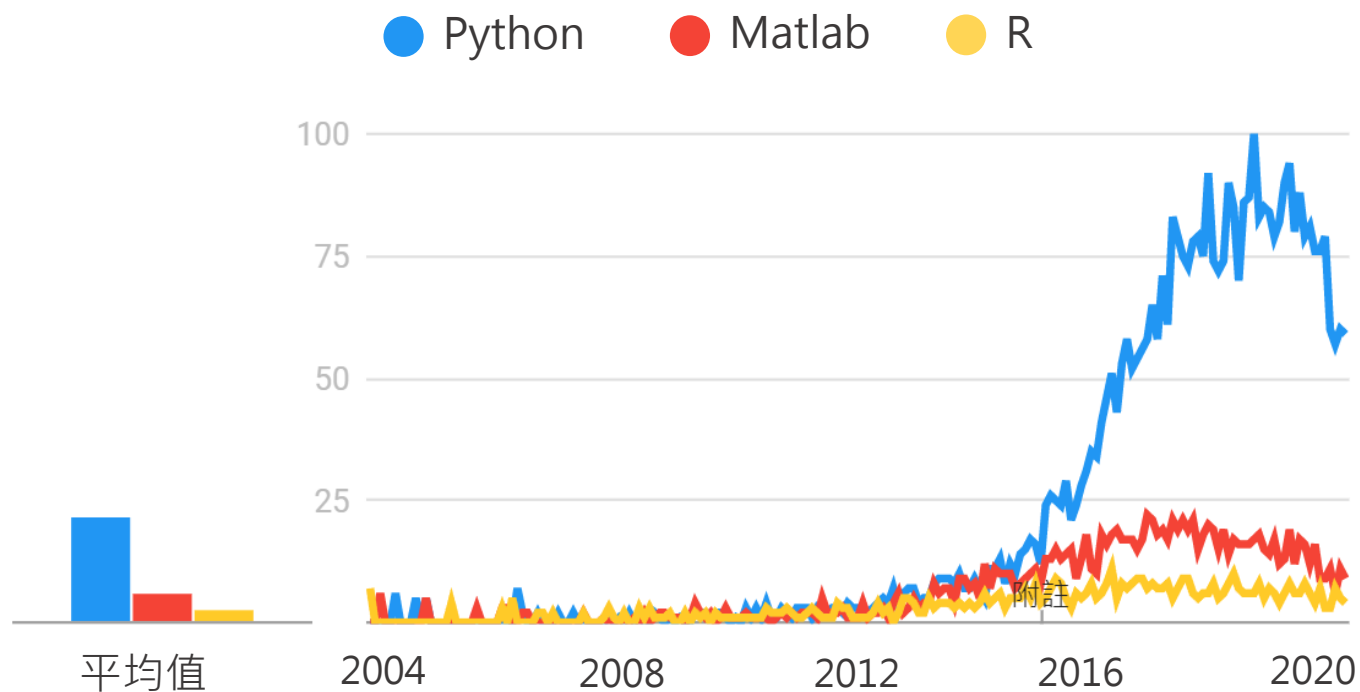
- 編程語言
- 深度學習框架
- 本機安裝Python
- Jupyter Notebook 自動補上語法
- 安裝套件



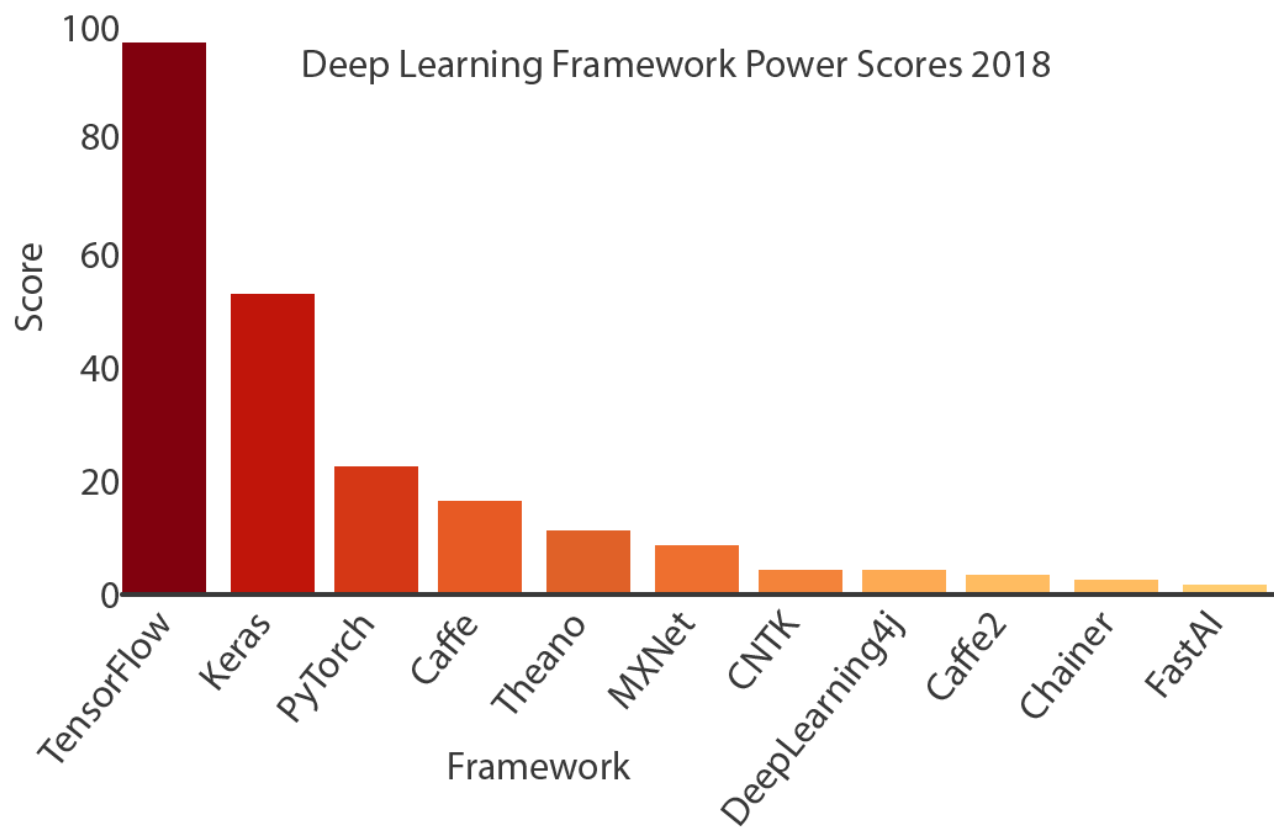
# 01 編程語言

## 01 人工智慧安裝篇

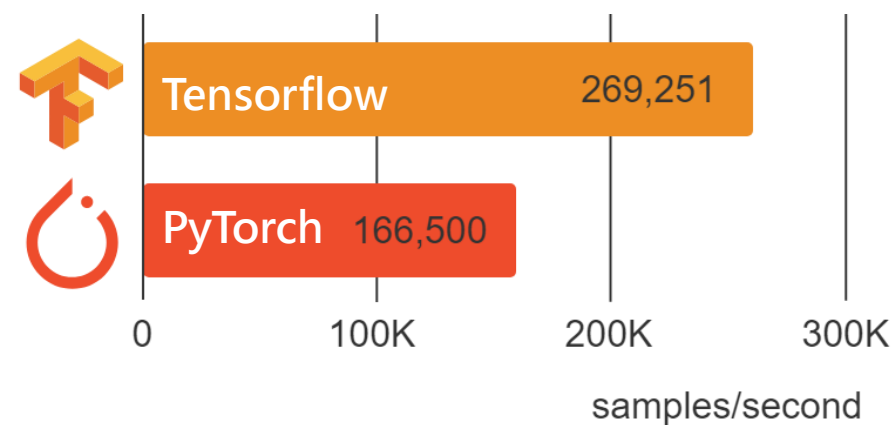
Google Trends 全球搜索詞彙：Machine and Deep Learning



## 02 深度學習框架



## 01 人工智慧安裝篇



	Tensorflow	PyTorch
Training time	01:01:54	01:40:06
Epoch time (seconds/epoch)	3.714	6.006
Training speed (samples/second)	269,251	166,500

## 03 本機安裝Python

- 從Anaconda官網中選擇下載電腦系統支援的安裝檔案(目前Anaconda的最高Python版本為3.9)  
<https://www.anaconda.com/products/individual>
- 執行安裝檔，環境變數需要打勾文字成紅色，其餘都按下一步即可
- 安裝完成後，將Anaconda以右鍵以「系統管理身分執行」
- 由於考量本教材使用的套件大多只支援**Python3.6**，所以創建Python 3.6的環境：
  1. Environments中在base(root)根環境，以左鍵點擊三角形選擇執行「Open Terminal」執行命令提示字元(CMD)
  2. 在CMD中輸入`conda create -n py36 python=3.6 anaconda`，開始創建名稱為py36與Python版本為3.6的環境
  3. 安裝完後，在Environments中點擊py36的環境，並轉換到「Home」，點擊安裝程式編譯器Jupyter Notebook「install」
  4. 完成後Environments中在py36環境，以左鍵點擊三角形選擇執行「Open with Jupyter Notebook」可以開始撰寫程式



## 04 Jupyter Notebook 自動補上語法

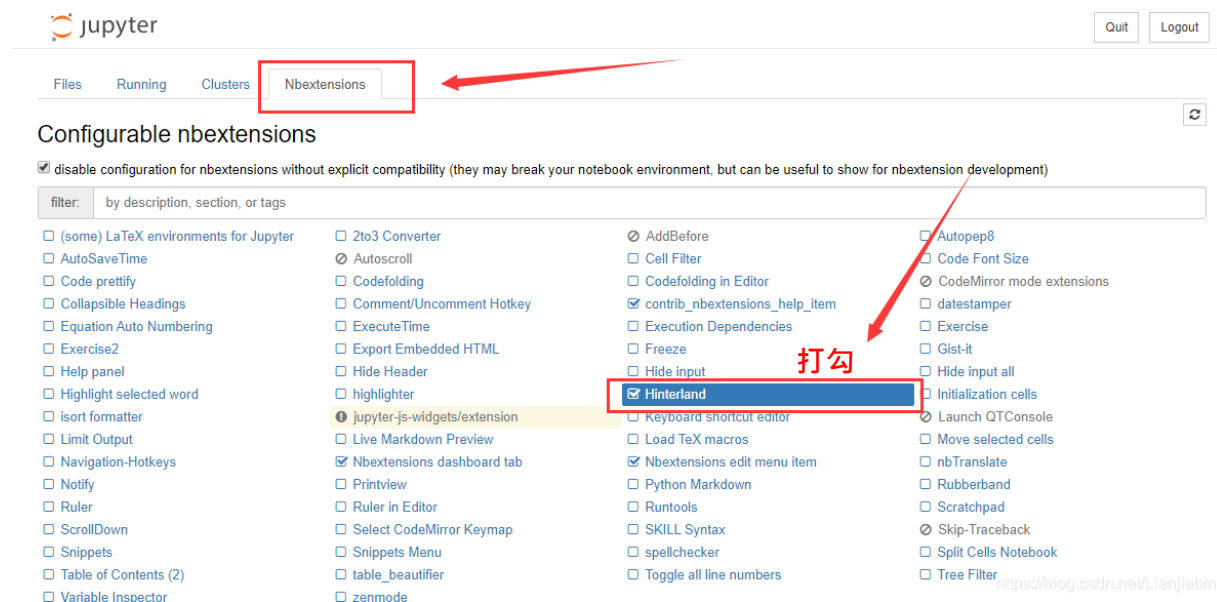
在 py36 環境中開啟命令提示字元執行下列指令：

### ■ 安裝 nbextensions

- ✓ `pip install jupyter_contrib_nbextensions`
- ✓ `jupyter contrib nbextension install --user`

### ■ 安裝 nbextensions\_configurator

- ✓ `pip install jupyter_nbextensions_configurator`
- ✓ `jupyter nbextensions_configurator enable --user`



■ 完成上述安裝後，請重新啟動 Jupyter Notebook，在彈出的Home裡面中，能看到增加了 Nbextensions 標籤頁，在這勾選Hinterland即啟動語法自動補全

- 補充：若要程式碼的左邊顯示行數，點擊「View」中的「Toggle Line Number(切換行數)」，預設為關閉



## 05 安裝套件

兩種安裝方式：

1. 可至PyPI網站（ <https://pypi.org/> ）上尋找套件名稱，並複製 `pip install package_name` 輸入至該環境的命令提示字元中



- 補充：pip 默認將 Python 包安裝到系統目錄，有時會需要系統管理訪問權限。  
添加 `-user` 指令是不需要任何特殊權限，`pip install package_name --user`
- 2. 在該環境中選擇「Not Installed」，對預安裝的套件點擊方框，並選擇「Apply」來安裝。或是使用 `conda install package_name`
- 補充：安裝方式 pip 與 conda 在環境中的依賴方面有所不同
  1. pip安裝時會在遞歸的串行循環中安裝依賴項。可能無法確保同時滿足所有套件的依賴性。如果較早安裝的套件與稍後安裝的套件具有不兼容的依賴性版本，則可能導致破壞環境
  2. conda使用可確保滿足環境中安裝的所有套件的所有要求。此檢查可能需要額外的時間，但有助於防止創建破壞環境

PART

2

# 資料讀取與處理

- 純文字文件
- 常見傳遞資料格式
- 影像
- 影片與攝影機串流



# 01 純文字文件

## ☑ 步驟：開啟 — 寫入/讀取 — 關閉

1. 開啟 `file_obj = open(file, mode= "r" )`

r 開啟檔案只供讀取，為預設值

w 開啟檔案供寫入，如果原先檔案有內容，其內容將被覆蓋

a 開啟檔案供寫入，如果原先檔案有內容，新寫入的資料將附加在後面

x 開啟一個新的檔案供寫入，如果所開啟的檔案已經存在則會產生錯誤

2. 讀檔 `file_obj.readlines()` 一次讀一行，以列表呈現

`file_obj.read()` 一次讀全部

3. 寫檔 `file_obj.write(str)`

`print(str, file=file_obj)`

4. 關閉 `file_obj.close()` 每次開啟檔案，請必要執行關閉

若使用with as 開啟檔案，以下程式執行完會自動關閉 `with open(file, mode= "r" ) as file_obj:`

## 02 常見傳遞資料格式

✓ Pandas 是一個資料處理與資料分析常用的開源套件 ( <https://pandas.pydata.org/docs/index.html> )

1. 給予資料與欄位字串，建立一DataFrame

```
df = pd.DataFrame(data, columns = [str])
```

2. 呈現DataFrame資料：前n筆資料(n預設為5)

```
df.head(n=5)
```

後n筆資料(n預設為5)

```
df.tail(n=5)
```

3. 從DataFrame中取得一欄位的資料：

```
data_column = df[str]
```

4. 儲存DataFrame資料：

- csv `df.to_csv(path, index = bool)`
- excel `df.to_excel(path, index = bool, sheet_name = str)`
- json `df.to_json(path)`
- xml `df.to_xml(path)`

index為第一欄位編號是否存取

sheet\_name 為工作表的名稱

備註：pandas 版本為 1.3.0 以上才能使用

## 02 常見傳遞資料格式

### 5. 讀取資料：

■ csv `df = pd.read_csv(path)`

■ excel `df = pd.read_excel(path, engine='openpyxl')`

openpyxl 支持較新的試算表格式

■ json `df = pd.read_json(path)`

■ xml `df = pd.read_xml(path)`

### 6. 新建欄位

`df[str] = data`

單欄新增，若str是以存在的欄位，其資料內容則會被取代

`df.insert(index,str,data)`

單欄新增，index為插入的欄位位置

`df = df.assign(str1 = data1, str2 = data2, ...)`

多欄新增

### 7. 新建資料(列)

`df2 = df2.append(df1)`

多列新增，df1的欄位名稱要與df2相同，此方法較多限制

# 03 影像



處理影像的套件常見的如下五個：

	OpenCV (cv2)	Matplotlib	Scipy
讀取資料	<code>cv2.imread(path)</code>	<code>matplotlib.image.imread(path)</code>	<code>scipy.misc.imread(path)</code>
資料型別	<code>numpy.ndarray</code>	<code>numpy.ndarray</code>	<code>numpy.ndarray</code>
顯示影像	<code>cv2.imshow(Title, img)</code>	<code>matplotlib.pyplot.imshow(img)</code> <code>matplotlib.pyplot.matshow(img)</code> <code>matplotlib.pyplot.show()</code>	<code>scipy.misc.imshow(img)</code>
儲存影像	<code>cv2.imwrite(path, img)</code>	<code>matplotlib.pyplot.imsave(path, img)</code>	<code>scipy.misc.imsave(path, img)</code>

	PIL	Tensorflow (tf), Keras
讀取資料	<code>PIL.Image.open(path)</code>	<code>tf.keras.preprocessing.image.load_img(path)</code>
資料型別	PIL	PIL
顯示影像	<code>img.show()</code>	<code>img.show()</code>
儲存影像	<code>img.save(path)</code>	<code>tf.keras.preprocessing.image.save_img(path, img)</code>

## 03 影像

1. PIL轉為陣列，可用兩種方法：`tf.keras.preprocessing.image.img_to_array(img)`，`numpy.array(img)`

2. OpenCV讀取影像的彩色通道為BGR。顯示影像也為BGR。

若要轉換彩色通道時，可以使用 `dst = cv2.cvtColor( src, code)`

- `dst` 表示輸出影像，與原始輸入影像具有同樣的資料類型與深度
- `src`表示原始輸入影像
- `code`是色彩空間轉換碼，常見的如右表 與 網站 [ [連結](#) ]：

補充：色彩空間的通道表示如下

- RGB 與 BGR：為光的三原色即紅、綠、藍色
- HSV：H為色調，S為飽和度，V為亮度
- YUV：又稱為YCrCb，Y為亮度，UV 表示色度，即色調與飽和度，分別用Cr和Cb來表示

說明	轉換碼
RGB與BGR轉換	<code>cv2.COLOR_BGR2RGB</code>
	<code>cv2.COLOR_RGB2BGR</code>
RGB/BGR與灰階轉換	<code>cv2.COLOR_BGR2GRAY</code>
	<code>cv2.COLOR_RGB2GRAY</code>
RGB/BGR與HSV轉換	<code>cv2.COLOR_BGR2HSV</code>
	<code>cv2.COLOR_RGB2HSV</code>
	<code>cv2.COLOR_HSV2BGR</code>
	<code>cv2.COLOR_HSV2RGB</code>
RGB/BGR與YUV轉換	<code>cv2.COLOR_BGR2YUV</code>
	<code>cv2.COLOR_RGB2YUV</code>
	<code>cv2.COLOR_YUV2BGR</code>
	<code>cv2.COLOR_YUV2RGB</code>

## 04 影片與攝影機串流

### 1. 初始化 `cap = cv2.VideoCapture(ID)`

初始化攝影機並存至`cap`變數中，其中ID為攝影機的ID號。預設值為-1，表示隨機選取一個攝影機。設為0為第一個攝影機，設為1為第二個攝影機，依此類推。ID也可以給予影片的位置來初始化預播放的影片

### 2. 捕捉畫面 `hasFrame, img = cap.read()`

`hasFrame` 表示捕捉是否成功，如果成功則該值為`True`，不成功為`False`。`img` 是傳回的捕捉到的畫面，如果沒有畫面被捕捉，則該值為空

### 3. 播放視訊時每一畫面的持續停留時間，該參數的單位為ms `cv2.waitKey(1)`

### 4. 釋放 `cap.release()`

### 5. 銷毀視窗 `cv2.destroyAllWindows()`

```
1 cap = cv2.VideoCapture(0)
2 while True:
3     hasFrame, img = cap.read()
4     cv2.imshow("Frame", img)
5     if cv2.waitKey(1) & 0xFF == ord('q'):
6         break
7 cap.release()
8 cv2.destroyAllWindows()
```



# 作業

## 02 資料讀取與處理

1. (20%) 打印三角形聖誕樹 ( 右圖 )，使用for迴圈打印，並儲存至純文字文件：

- 樹葉為底十個 \* 字號，依序減二，最高為兩個 \* 字號且皆置中
- 樹幹為高寬兩個 \*

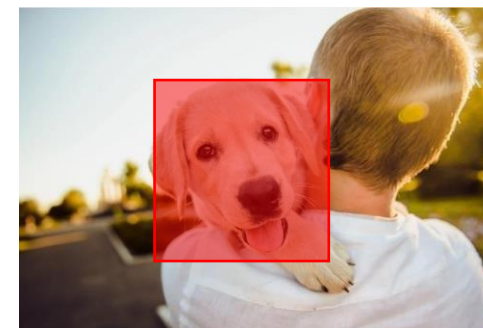
```
  **
 ****
*****
*****
*****
  **
  **
```

2. (20%) 表格 ( 左圖 ) 再新增欄位為 BMI，其數值為身高與體重所計算的，型別為浮點數取小數點兩位 ( 右圖 )

	age	city	height	weight	sex	SBP	DBP
0	23	Japan	175	68	M	120	85
1	18	Taiwan	168	55	F	114	90
2	30	USA	173	75	M	145	75
3	25	Taiwan	158	50	F	110	78



	age	city	height	weight	sex	SBP	DBP	BMI
0	23	Japan	175	68	M	120	85	22.20
1	18	Taiwan	168	55	F	114	90	19.49
2	30	USA	173	75	M	145	75	25.06
3	25	Taiwan	158	50	F	110	78	20.03



3. (20%) 右圖影像轉為陣列裁減至左上角(140, 220)至右下角(450, 520)的矩形，並用Matplotlib顯示其裁減影像並儲存

# 作業

## 02 資料讀取與處理

4. (20%) 以OpenCV讀取影像，不能使用 `cv2.cvtColor()` 而是以陣列取代方式來實現BGR轉RGB通道，並用Matplotlib顯示
5. (20%) 讀取影像並調整影像亮度提高20%與降低20%，並用Matplotlib顯示
6. (20%) 開啟攝影機：
  - 在畫面中繪製紅色正方形在畫面左上角位置(100, 100) 到 右下角位置(200, 200)
  - 將純紅色正方形改為透明度60%的紅色



PART

3

# 身體質量指數 回歸

- 讀取資料
- 特徵、標籤、特徵名稱與標籤名稱
- 資料視覺化
- 分類模型訓練



## Body Mass Index



PART

4

## 鳶尾花 分類

- 讀取資料
- 特徵、標籤、特徵名稱與標籤名稱
- 資料視覺化
- 分類模型訓練



# 01 鳶尾花資料集

## 04 鳶尾花 分類

```
1  from sklearn import datasets
2  iris = datasets.load_iris()
3  print(iris.keys())
4
5  features = iris.data
6  label = iris.target
7  features_name = iris.feature_names
8  label_name = iris.target_names
9
10 print("標籤類別數量 :", Counter(label))
11 print("特徵名稱 :", features_name)
12 print("標籤名稱 :", label_name)
```



PART

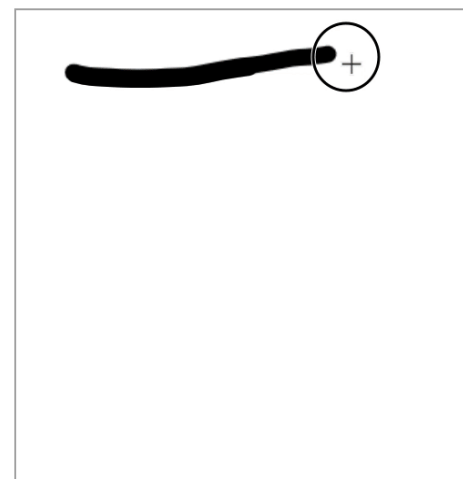
5

# 數字手寫影像 分類

- 讀取資料
- 特徵、標籤、特徵名稱與標籤名稱
- 資料視覺化
- 分類模型訓練

# 01 數字手寫影像資料集

## 05 數字手寫影像 分類



Draw a single digit (0-9) in the empty box.

Clear

