

# 緣由

在影像辨識中我們期望可以做到輸入為一張圖片，經過一個 `function` 之後輸出為那張圖片中物品名稱，由於該 `function` 過於複雜人力無法處理，因此誕生了 ML 的相關技術。

## 機器學習的步驟

設計機械學習的流程大致可分簡略分作下列步驟。

### 一、決定問題的種類

首先需要根據想要解決的問題決定要用何種 ML 的方法進行運用，意即我們所需要的 `function` 是什麼樣子，其大致可分作三類。

#### 1. Regression

透過給機器多筆 `training data`，透過通過 `regression` 找到的 `function` 期望他輸出一個 `scalar`（例如預測結果）。

#### 2. Classification

透過 `training data` 讓機器學會如何分類輸入的類別，又可分作兩小項。

##### (1) Binary Classification

讓機器輸出的結果是 `yes or no`。（例如判斷輸入的郵件是不是垃圾郵件）

##### (2) Multi-class classification

在該項問題中，機器要做的是選擇題，使他給出每一個選項符合輸入類別的機率（例如影像辨識）

#### 3. Structured Learning

在 `structured Learning` 裡，我們要機器輸出的是一個有結構性的東西。舉例來說，在人臉辨識的情境中，希望機器能標出不同的人的名稱。

## 二、決定 Leaning 的方法

再者我們要考慮要怎麼告訴機器我們需要找什麼樣的 function，意即訓練機器的方法。（基於手上的 data 或是想要達成的目的作決定）

### 1. Supervised Learning

用含有大量正確答案的資料告訴機器怎樣的判斷才是正確的。以影像辨識的問題來說，其 training data 可能有大量貓狗的圖片，並標註其為貓或狗，藉此讓機器知道「貓」與「狗」的不同。

### 2. Semi-supervised Learning

在 training set 內有少量的 labeled data，又有大量的 unlabeled data。在 Semi-supervised Learning 的技術裡面，這些沒有 labeled 的 data，對機器學習也是有幫助的。

### 3. Unsupervised Learning

希望機器學到無師自通，在完全沒有任何 label 的情況下，機器到底能學到什麼樣的知識。例如給機器大量動物的圖片希望他能學會「貓」與「狗」的概念。

### 4. Transfer Learning

同樣只有少量的有 labeled 的 data；但是我們現在有大量的不相干的 data（不是貓和狗的圖片，而是一些其他不相干的圖片），在這些大量的 data 裡面，它可能有 label 也可能沒有 label。

該技術要解決的問題是，這一堆不相干的 data 可以對結果帶來什麼樣的幫助。

### 5. Reinforcement Learning

在該技術中我們沒有告訴機器正確的答案是什麼，機器最終得到的只有一個分數，就是它做的好還是不好，但他不知道自己到底哪裡做的不好。其特點是 learning from critics。

### 三、限定 Model 的範圍

在讓機器怎麼找出想要的 function 之前我們需要限定 model 的搜尋範圍，讓機器在限定的 model 內找出最好的一個。

#### 1. Linear Model

最簡單的 model ( $y_i = w_{ni}x_i^n + w_{(n-1)i}x_i^{n-1} + \dots + b$ )。

優：結構簡單，可以很快地找出 best function，運算速度快。

容易設計。

缺：在處理複雜的問題上表現較差。

需要 feature transform。

#### 2. Non-linear Model

最常用的 model，其常見類別有下列各項。

(1) SVM

(2) Decision Tree

(3) K-NN

#### (4) Generative Model

用於 classification 的問題，由於在該問題中單純透過機器分類對或錯定義出來的 loss function 無法微分（不能用 gradient descent），因此採用機率的觀點來看待。意即計算 input 屬於各個 class 的機率 ( $P(C_i|x_i)$ )。

在計算  $P(C_i|x_i)$  時，我們須知  $P(x_i|C_i)$  的機率故再將 training data 的 distribution 假設為下列方法其一。

##### a. Naive Bayes Classifier

假設每一個 dimension 的分佈函數都是一維的 Gaussian distribution，使其 covariance matrix 變成 diagonal。

優：減少需要的參數量

缺：若 feature 之間不是相互獨立的，其 bias 就會很大。

##### b. Gaussian Distribution

將 training data 的 distribution 用  $\mu$ (mean)與  $\Sigma$ (covariance)表示。

##### c. Bernoulli Distribution

優：由於假設了 distribution 的形式，因此可以補足 data 的不足。

受有問題的 data 影響較小。

priors probabilities 和 class-dependent probabilities 可從不同的來源獲得。

→減少 data 的收集量→把 prior 預測地更精確。

缺：表現較 Discriminative Model 差。

### **(5) Discriminative Model (Logistic Regression)**

同樣用於 classification 的問題。進一步假設 Gaussian Distribution 中的所有 class 都共用一個 covariance，使其變成 linear 的形式，同樣由  $w$ (weight)與  $b$ (bias)控制，最後再經過 sigmoid function 處理。

優：一般表現較 Generative Model 好。

缺：很依靠 training data，因此受有問題的 data 影響較大。

### **(6) Deep Learning (Neural Network)**

可想成將多個單一的 function 連接起來，前一個輸出為後者的輸入，通常分作 input layer、hidden layer 與 output layer。

優：適合處理人類較為直覺的任務，意即無法很明確 extract feature 的問題。

→ 影像辨識、語音辨識

缺：結構複雜，不容易設計，且運算時間較長，通常會用 GPU 加速。

(待補)

## 四、從範圍內找出好的 **function**

訓練時機器會從決定好的 model 類型中透過定義 loss function，找出最適合的 function (or network)。依據問題的種類其長相不同，該課程目前為止以下列兩項最常見。

### 1. Square Error

$$L(f) = \sum_n (\hat{y}^n - f(x_i^n))^2$$

適合用於 regression 的問題，意即該輸出僅有 scalar，基於 training data 所作的數值預測。

優：容易微分使用 gradient descent 找 optimal point。

缺：距離目標遠的時候，微分也是非常小的，移動的速度非常慢，容易卡住。

### 2. Cross Entropy

$$L(f) = \sum_n l(f(x_i^n), \hat{y}^n)$$

$$l(f(x^n), \hat{y}^n) = -[\hat{y}^n \ln f(x^n) + (1 - \hat{y}^n) \ln(1 - f(x^n))]$$

適合用於 classification 的問題，分析各選項的機率。

優：距離目標越遠，微分值就越大，參數 update 的時候變化量就越大。

## 五、找出最小的 Loss Function

透過不斷更新參數找出使 loss function 最小的一組參數。最常見有下列幾種方法。

### 1. Gradient Descent

隨機選取一組起始的參數，透過計算參數的 gradient 乘上 learning rate 的值更新參數。

$$\theta^i = \theta^{i-1} - \eta \nabla L(\theta^{i-1})$$

缺：learning rate 過大過小都會影響其表現。

無法針對每個參數分配一個 learning rate。

gradient 越大，離最低點越遠在有多個參數的情況下不一定成立。

可能不是停在 global minimum→不適用於非 convex。

### 2. Adagrad

將 learning rate 除以 root mean square，使不同參數的 learning rate 分開考慮。

$$w^{t+1} \leftarrow w^t - \frac{\eta^t}{\sigma^t} g^t$$

優：針對參數客製化。

在不增加任何額外運算的前提下，想辦法去估測二次微分的值。

缺：速度慢。

### 3. Stochastic Gradient Descent

隨機看到一個樣本點就 update 一次，其 loss function 不是所有樣本點的 error 平方和，而是這個隨機樣本點的 error 平方。

優：速度較快。

### 4. Momentum Gradient Descent

(待補)