# 1

# 簡介與概述

我們之所以寫這本書，主要是想協助機器學習（ML；Machine Learning）工程師和資料科學家，能夠順利通過 ML 系統設計的面試。對於任何想學習如何在現實世界裡善用 ML 高階概念的人來說，本書也是很有幫助的。

許多工程師都以為，像是邏輯迴歸（logistic regression）、神經網路（neural network）之類的 ML 演算法，大概就是 ML 系統的全部了。不過，真正的 ML 系統，絕不僅僅是模型的開發而已。ML 系統通常很複雜，由許多組件所組成，其中包括管理資料的資料堆疊（data stack）、讓系統可以供好幾百萬人使用的伺服基礎設施、衡量系統表現的評估管道，以及確保模型表現不會隨時間遞減的監控做法等等。

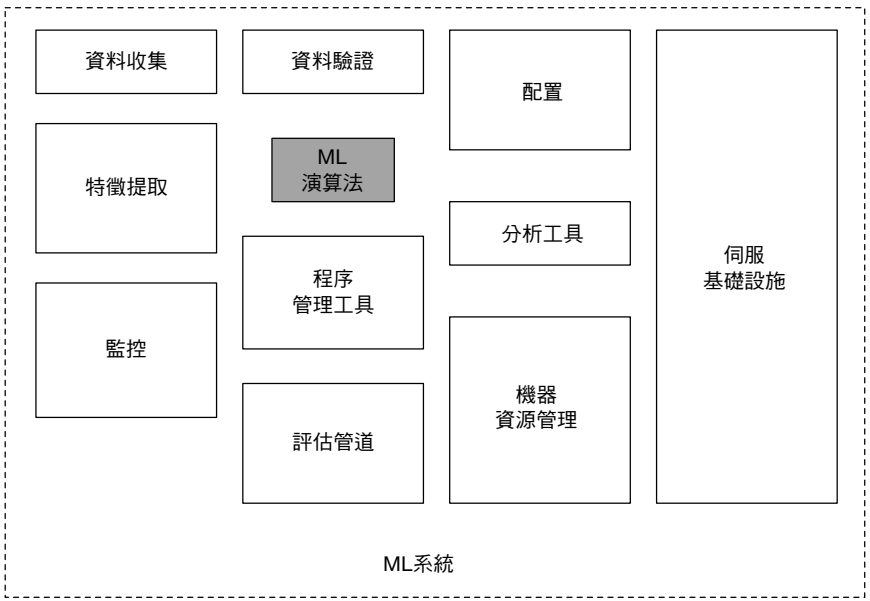


圖 1.1　一個可正式上線的 ML 系統所包含的各種組件

在 ML 系統設計面試過程中，你必須能夠回答一些開放式的問題。舉例來說，你可能會被要求設計出一套電影推薦系統，或是一個影片搜尋引擎。這類問題根本就沒有標準答案。面試官想要評估的是你思考的過程、你對各種 ML 主題的理解有多深、你設計出一整套端到端系統的設計能力，以及你在各種不同的設計選項之間，如何權衡並做出取捨。

如果想成功設計出複雜的 ML 系統，先有一套可依循的框架就很重要。非結構化的做法，只會讓整個設計流程變得非常難以依循。在本章的簡介中，我們會先提出一套本書所採用的框架，以解決 ML 系統設計相關的各種問題。這整套框架是由下面這幾個關鍵的步驟所組成：

1. 把各種要求明確化
2. 用框架把問題轉化成 ML 任務
3. 資料的準備
4. 模型的開發
5. 進行評估
6. 進行部署並提供服務
7. 監控與基礎設施相關考量

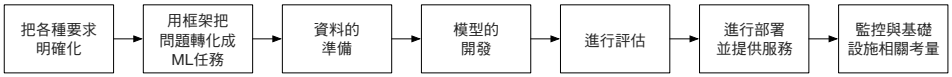


圖 1.2　ML 系統設計的步驟

ML 系統設計面試每次的過程肯定不盡相同，因為題目往往是開放式的，並沒有一體適用的做法。本書的框架主要是希望協助你建立自己的想法，不過你並不需要嚴格遵循這個框架。要懂得變通、保持彈性。如果面試官很明顯對於模型的開發比較感興趣，你也許就要特別留意他所關注的重點。

接著我們就來檢視一下，這個框架裡的每一個步驟吧。

## 把各種要求明確化

ML 系統設計方面的題目，通常會故意講得不太明確，只透露出極少的資訊。舉例來說，面試的題目可能長這樣：「設計出一個活動推薦系統」。而我們的第一個步驟，就是要提出幾個能把整件事明確化的問題。但是，究竟該問什麼樣的問題呢？呃……我們所提出的問題，應該要讓我們能夠確切瞭解真正的需求。以下就是可以協助我們找出頭緒的幾類問題：

- **商業上的目標。**如果我們被要求建立一個「短期租屋推薦系統」，或許「增加預訂數量」和「增加營收」就是其中兩個可能的動機。
- **系統所要支援的功能。**這個系統預計支援哪一些功能？這一定會影響到我們對於 ML 系統的設計。舉例來說，假設我們被要求設計出一個影片推薦系統。我們或許想知道，使用者能不能針對所推薦的影片，表達出「喜歡」或「不喜歡」的感覺（例如按讚、給評分），而這些互動資料，就可以用來標記（label）我們的訓練資料。
- **資料。**資料的來源有哪些？整個資料集有多大？資料有做過標記嗎？
- **限制。**有多少運算能力可供運用？採用的是雲端系統，還是只能在設備端運行系統？模型需不需要自動隨時間逐步改進？
- **系統規模。**有多少使用者？有多少東西（例如有多少影片）需要處理？這些指標數字的增長率又是如何呢？
- **效能表現。**預測的速度要有多快？解法是否需要即時做出反應？正確率比較重要，還是延遲的問題比較重要？

上面這份列表並不算詳盡，不過你還是可以把它當成一個起點。請注意，其他的面向（例如隱私權和道德考量）可能也蠻重要的。

到了這個步驟結束之時，我們應該就可以預期，對於系統的涵蓋範圍與要求，我們已經與面試官有了一致的看法。如果能把我們所收集到的要求與限制寫成一份清單，通常也是很棒的做法。這樣一來，就可以確保大家都有相同的認知了。

## 用框架把問題轉化成 ML 任務

在解決 ML 問題的過程中，運用框架把問題進行有效的轉化，是一個極為重要的步驟。假設面試官要求你想辦法提高影片串流平台的使用者參與度。使用者的參與度不夠，當然是個問題，不過這並不能算是一個 ML 任務。我們應該先運用框架，把它轉化成 ML 任務，然後再來解決它。

實際上，一開始我們應該先做個判斷，有沒有必要採用 ML 的做法來解決手中的問題。在 ML 系統設計面試過程中，我們或許可以假設，ML 應該是有用的做法才對。因此，接下來就可以透過以下的做法，用框架把問題轉化成 ML 任務：

- 定義 ML 的目標
- 設定系統的輸入和輸出
- 選擇正確的 ML 類別

### 定義 ML 的目標

商業上的目標也許是提高 20% 銷售額，或是提高留客率（user retention）。不過這樣的目標並不算是很好的定義，而且我們在訓練模型時，也不能只告訴它「提高 20% 的銷售額就對了」。為了讓 ML 系統能夠解決特定任務，我們必須把商業上的目標，轉化成定義很明確的 ML 目標。一個良好的 ML 目標，其實就是 ML 模型能夠解決的目標。我們先來看一些例子，如表 1.1 所示。在後面的章節中，我們還會看到更多的例子。

表 1.1　把商業上的目標轉化成 ML 目標

| 各種應用 | 商業上的目標 | ML 目標 |
|---|---|---|
| 活動門票銷售 App | 提高門票的銷售量 | 最大化活動報名人數 |
| 影片串流 App | 提高使用者的參與度 | 最大化使用者觀看影片的時間 |
| 廣告點擊預測系統 | 增加使用者的點擊次數 | 最大化點擊率 |
| 社群媒體平台的有害內容偵測 | 改善平台的安全性 | 準確預測出內容是否有害 |
| 朋友推薦系統 | 提升使用者拓展人際網路的速度 | 最大化人與人建立朋友關係的數量 |

# 3

# Google 街景模糊化系統

Google 街景 [1] 是 Google 地圖裡的一種技術，可針對世界各地許多公路網提供街道級的互動式全景圖。2008 年，Google 創建了一個可以讓人臉與車牌自動模糊化（blur）的系統，藉此保護使用者的個人隱私。我們在本章設計了一個很類似 Google 街景的模糊化系統。


圖 3.1　車牌模糊化的街景圖片

## 把各種要求明確化

以下就是應試者與面試官之間相當典型的一段對話過程。

**應試者：**這個系統在商業上的目標，可以說是為了保護使用者個人隱私嗎？

**面試官：**是的。

**應試者：**我們希望設計出一套系統，可偵測出街景圖片裡所有的人臉和車牌，並在呈現給使用者看之前，先進行模糊化處理。這樣是對的嗎？我能否假設，如果使用者看到沒被正確模糊化的圖片，可以向我們回報問題？

**面試官：**可以的，這些都是合理的假設。

**應試者：**我們手頭上有沒有可運用於這個任務的已標記資料集呢？

**面試官：**你可以假設我們已經抽樣了 100 萬張圖片。圖片裡的人臉和車牌，全都已經用人工進行了標記。

**應試者：**資料集有可能並沒有包含某類人的臉孔特徵，這可能會導致系統對於特定的人類屬性（例如種族、年齡、性別等等）存在特定的偏見。這是個合理的假設嗎？

**面試官：**這是很棒的觀點。不過為了簡單起見，我們今天先不用去處理公平性和特定偏見的問題。

**應試者：**我的理解是，延遲並不會是個大問題，因為系統可以在離線的情況下進行物體偵測與模糊化處理。這樣的理解是對的嗎？

**面試官：**是的。我們可以先把現有的圖片呈現給使用者，同時以離線的方式去處理新的圖片。

這裡就來總結一下問題的陳述吧。我們想設計出一個街景模糊化系統，可以讓車牌和人臉自動模糊化。我們已經取得了一組訓練組資料，其中包含 100 萬張已標記過人臉和車牌的圖片。這套系統在商業上的目標，就是要保護使用者的個人隱私。

## 用框架把問題轉化成 ML 任務

我們會在本節用框架把問題轉化成 ML 任務。

## 定義 ML 的目標

這個系統在商業上的目標，就是把街景圖片中可以看到的車牌與人臉進行模糊化處理，以保護使用者的個人隱私。不過，保護使用者個人隱私並不是 ML 的目標。因此，我們要把它轉化成 ML 系統可以解決的 ML 目標。其中一個可以考慮採用的 ML 目標，就是準確偵測出圖片中讓人感興趣的一些物體。如果 ML 系統可以準確偵測出這些物體，我們就可以在呈現圖片給使用者看之前，先把某些物體模糊化。

為了簡潔起見，本章接下來會改用「物體」來取代「人臉與車牌」的說法。

## 設定系統的輸入和輸出

物體偵測（object detection）模型的輸入是一張圖片，其中在不同的位置可能有零或多個物體。這個模型應該可以偵測出這些物體，並輸出各個物體的位置。圖 3.2 顯示的就是一個物體偵測系統及其輸入和輸出。

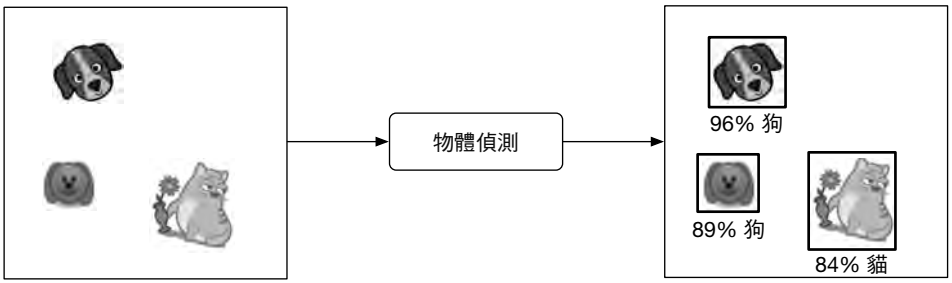


圖 3.2　物體偵測系統的輸入 / 輸出

## 選擇正確的 ML 類別

一般來說，物體偵測系統有兩項職責：

- 預測出圖片裡每個物體的位置
- 預測出每個邊界框的所屬類別（例如狗、貓等等）

第一個任務屬於迴歸問題，因為位置可以用（x, y）座標來表示，而這些座標全都是一些數值。第二個任務則可以轉化成多類別分類問題。

傳統上，物體偵測架構可分成所謂的「一階段」（one-stage）和「兩階段」（two-stage）網路。近年來，Transformer 型架構（例如 DETR [2]）也呈現出相當令人期待的成果，不過本章主要還是只探討兩階段與一階段的架構。

## 兩階段網路

顧名思義，兩階段網路會使用到兩個獨立的模型：

1. **區域提議網路（RPN；Region Proposal Network）：**掃描圖片並提出一些有可能是物體的候選區域。
2. **分類器（Classifier）：**針對每個提議的區域進行處理，然後再把它歸類為某個物體類別。

圖 3.3 顯示的就是這種兩階段的做法。

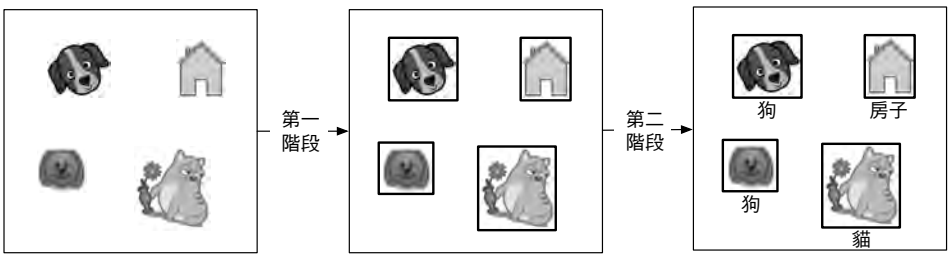


圖 3.3　兩階段網路

常用的兩階段網路有：R-CNN [3]、Fast R-CNN [4] 和 FasterRCNN [5]。

## 一階段網路

在這類網路中，兩個階段被整合了起來。只使用單獨的一個網路，就能同時生成邊界框與物體類別，而不必明確偵測出提議區域。圖 3.4 顯示的就是一階段網路的例子。

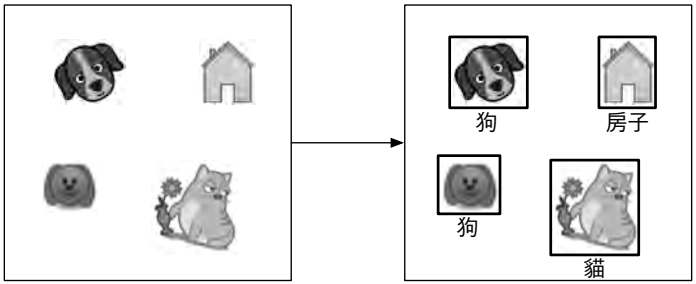


圖 3.4　一階段網路

常用的一階段網路有：YOLO [6] 和 SSD [7] 架構。

### 一階段 vs. 兩階段

兩階段網路是由兩個按照順序前後執行的組件所組成，因此速度通常比較慢，但結果通常也會比較準確。

以我們的題目來說，我們的資料集有 100 萬張圖片，如果以現代的標準來看，這並不算是非常龐大的資料量。這也就表示，使用兩階段網路並不會過度增加訓練的成本。因此，以這裡的練習來說，我們決定要從兩階段網路開始著手。如果訓練資料大幅增加，或是需要更快速的預測，也可以切換成一階段網路。

## 資料的準備

### 資料工程

在「簡介」的那一章，我們已經討論過一些資料工程的基礎知識。除此之外，討論一下手頭上的任務有哪些可運用的具體資料，通常是個還不錯的好主意。以這裡的問題來說，我們可取得以下這些資料：

- 已標記的資料集
- 街景圖片

接著就分別來仔細討論一下吧。

## 已標記的資料集

根據需求，我們已經有 100 萬張已標記過的圖片。每張圖片都有一個列表，其中包含一堆的邊界框，以及相關聯的物體類別。表 3.1 顯示的就是資料集裡的一些資料點：

表 3.1　已標記資料集裡的一些資料點

| 圖片路徑 | 物體 | 邊界框 |
|---|---|---|
| dataset/image1.jpg | 人臉<br>人臉<br>車牌 | [10, 10, 25, 50]<br>[120, 180, 40, 70]<br>[80, 95, 35, 10] |
| dataset/image2.jpg | 人臉 | [170, 190, 30, 80] |
| dataset/image3.jpg | 車牌<br>人臉 | [25, 30, 210, 220]<br>[30, 40, 30, 60] |

每個邊界框都是包含 4 個數字的一個列表，分別代表左上角的 X、Y 座標，然後是這個物體的寬度和高度。

## 街景圖片

這些全都是資料來源團隊所收集的街景圖。ML 系統會去處理這些圖片，偵測出其中的人臉和車牌。表 3.2 顯示的是這些圖片的詮釋資料（metadata）。

表 3.2　街景圖片的詮釋資料

| 圖片路徑 | 地點（經緯度） | 俯仰角、偏轉角、側傾角 | 時間戳 |
|---|---|---|---|
| tmp/image1.jpg | (37.432567,-122.143993) | (0, 10, 20) | 1646276421 |
| tmp/image2.jpg | (37.387843, -122.091086) | (0, 10, -10) | 1646276539 |
| tmp/image3.jpg | (37.542081,-121.997640) | (10, -20, 45) | 1646276752 |

## 特徵工程

在特徵工程這個階段，我們會先套用一些標準的圖片預處理操作（例如重新調整大小、正規化處理）。然後，我們還會運用資料擴增衍生技術，增加資料集的資料量。接著就來仔細看一下吧。

### 資料擴增

所謂的「資料擴增」（data augmentation）技術，就是稍微修改原始資料以建立一些資料副本，或是以人為方式根據原始資料衍生出新的資料，然後再添加到資料集內的一種做法。資料集的資料量增加之後，模型就可以學習到更複雜的特定模式。尤其是失衡的資料集，使用這個技術特別有用，因為它可以用來增加少數類資料點的數量。

圖片擴增（image augmentation）可以算是資料擴增其中的一種特殊類型。常用的擴增衍生技術包括：

- 隨機裁剪
- 隨機飽和度處理
- 垂直或水平翻轉
- 旋轉或平移
- 仿射變換（Affine transformations）
- 改變亮度、飽和度、對比度

圖 3.5 顯示的就是套用各種資料擴增技術之後衍生出來的幾張圖片。

NMS 是 ML 系統設計面試時經常被拿出來問的一種演算法，所以我們非常鼓勵你去充分理解它的原理 [18]。

## ML 系統設計

如圖 3.12 所示，我們提出了一個模糊化系統的 ML 系統設計圖。

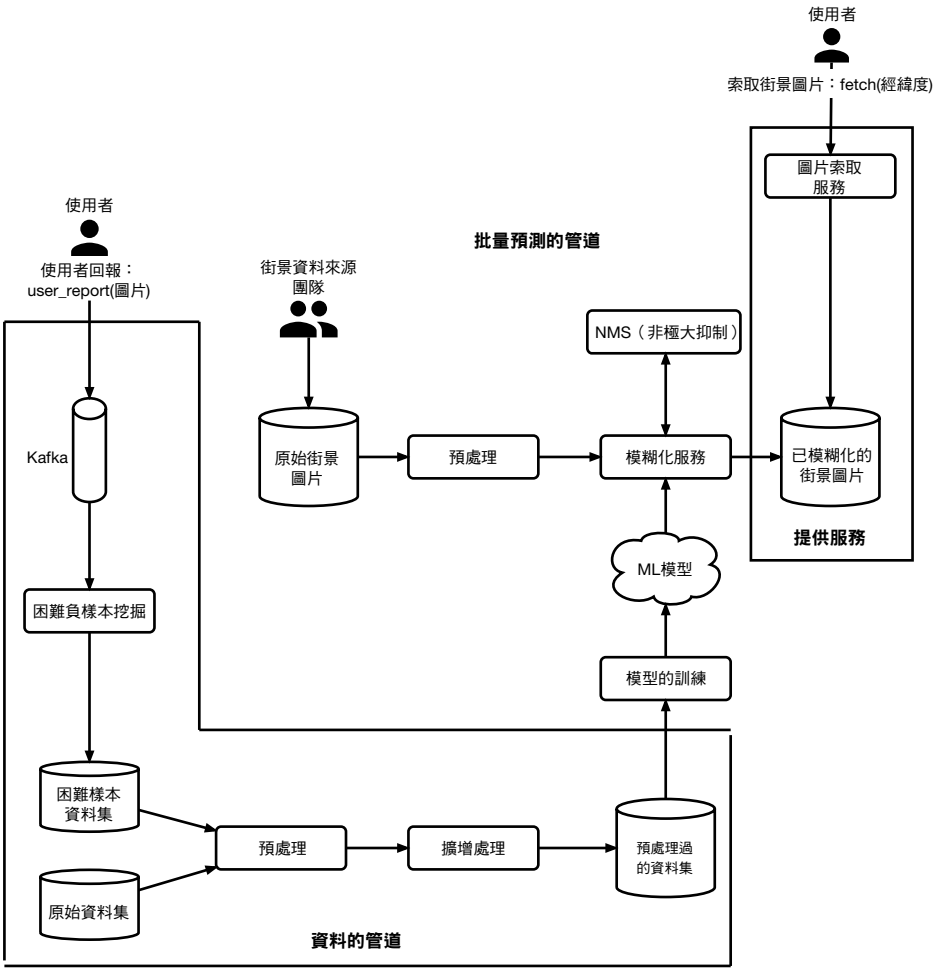


圖 3.12　ML 系統設計圖

我們就來詳細檢視每一個管道吧。

**困難負樣本挖掘。**所謂的「困難負樣本」（Hard negatives），其實就是把一些預測錯誤的樣本明確定義為負樣本，然後再把這些負樣本加到訓練組資料中。如果我們用這組更新過的訓練組資料來重新訓練模型，模型應該就會有更好的表現。

## 其他討論要點

如果時間允許，這裡還有一些可以討論的要點：

- Transformer 型物體偵測架構與一階段或兩級模型有何不同，它們各有什麼優缺點 [19]。
- 如果遇到資料集的資料量比較大的情況，可以考慮採用分散式訓練技術，來改進物體偵測的效果 [20] [21]。
- 歐洲的「一般資料保護規範」（GDPR；General Data Protection Regulation）對於我們的系統會有什麼樣的影響 [22]。
- 評估人臉偵測系統裡的特定偏向 [23][24]。
- 如何持續微調模型 [25]。
- 如何利用主動學習（active learning）[26] 或是讓人類參與其中的機器學習做法 [27] 來選出可用來投入訓練的資料點。

## 總結