第 1 章

統計分析的第一步、統計分佈的可視化與分析

取得資料之後的第一件事

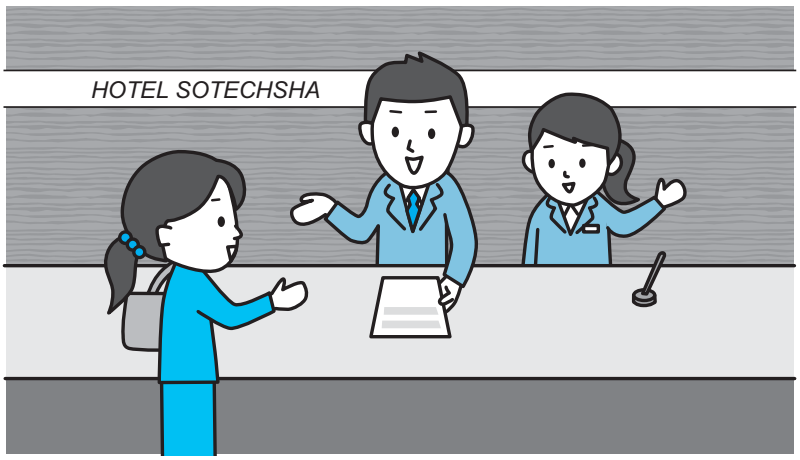
最近很常聽到「資料分析」、「機器學習」這類字眼，而這些技術的基礎其實都是「機率與統計」，但很少專業書籍介紹這些技術與「機率、統計」有什麼關係。如果缺乏機率或統計的基礎知識就隨便進行資料分析，或是使用機器學習的函數與工具，就有可能導出錯誤的分析結果。

只要具備一些簡單的基礎知識就能避免上述的錯誤發生。而且了解基礎知識之後，就能找出專屬自己的使用方法，也能知道該如何改善分析方式，所以了解基礎知識可說是非常重要的一環。

本章要介紹取得資料之後，最該先做的是哪些事情，也要介紹分析資料，取得統計值的方式，讓大家透過這一連串的步驟紮穩馬步，以便學習第2章的機器學習以及第3章的推測統計。與其說本章都是數學相關的內容，不如說本章是幫助大家做好事前準備的章節，建議大家一邊搜尋必要的知識，一邊試著動手做看看。

分析顧客行為資料的背景

某間飯店爲了提升業績，拜託你分析相關資料。這間飯店是在東京都內擁有150間客房的渡假村，在新冠疫情爆發後，來客數曾一時陷入低迷，但是在降價以及提供遠距辦公的單人房之後，來客數有慢慢回升的跡象。目前你已拿到這兩年來的住宿資料，所以可根據這份資料開始分析，也可著手整理已知的部分。



1-1 試著載入資料

分析資料的第一步就是載入資料。請執行下列的程式碼，載入**accomodation info.csv**。

載入資料 Chapter1.ipynb

```
1 import pandas as pd
2 df info = pd.read csv("accomodation info.csv",
  index col=0, parse dates=[0])
3 df info
```

圖 1-1-1 顯示資料

Out[1]:

日期	顧客ID	住宿者姓名	方案	金額
2018-11-01 00:02:21	110034	若松 花子	B	19000
2018-11-01 00:03:10	112804	津田 美加子	D	20000
2018-11-01 00:06:19	110275	吉本 美加子	D	20000
2018-11-01 00:08:41	110169	坂本 直人	B	19000
2018-11-01 00:12:22	111504	青山 零	A	15000
...	...	...	...	...
2020-10-31 23:38:51	110049	吉本 篤司	A	3000
2020-10-31 23:42:12	110127	喜嶋 浩	A	3000
2020-10-31 23:47:24	115464	藤本 明美	D	8000
2020-10-31 23:53:22	114657	鈴木 七夏	A	3000
2020-10-31 23:57:21	111407	鈴木 治	A	3000

71722 rows × 4 columns

這份資料包含住宿報到時間、住宿者姓名、對應住宿者的顧客ID、住宿者選擇的方案（A～D四種），以及方案的費用。星期一～日的住宿費用各有不同之外，住宿費用還會隨著季節變動，尤其在新冠疫情爆發，來客數減少之後，費用更是大幅調降。

住宿者可選擇的方案包含A（無餐點）、B（附早晚餐）、C（無餐點、露天浴缸）、D（附早晚餐與露天浴缸）這四種，在新冠疫情爆發之前，在餐廳享用的早晚餐方案最受歡迎。

接著，觀察疫情爆發之後，住宿情況的變化，並進行分析。

全貌

統整最佳化問題全貌的背景

之前的倉儲公司在拜託你撰寫配送路線最佳化的程式碼後，希望你能繼續幫他們處理另一個問題。

原本希望讓「AI學習相關工作經驗與技巧」的倉儲公司窗口發現，「只要進一步了解最佳化問題，不僅能讓更多工作經驗與技巧轉換成公式，還能就此開發一套自動計算的系統」。

因此，對方請你先統整這世上所有的最佳化問題。請想想看，該怎麼做才能統整最佳化問題的全貌。

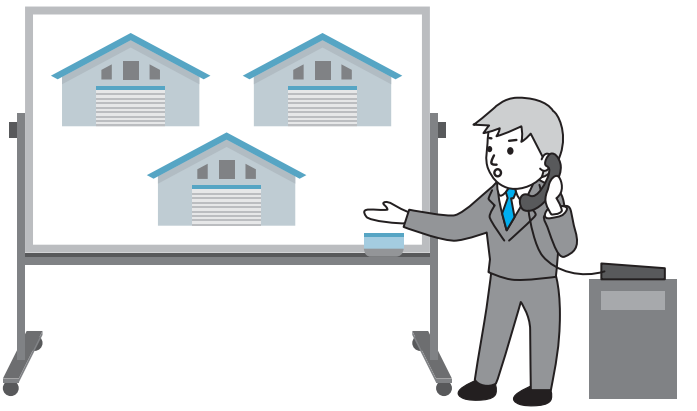


圖 5-6-3 利用貪婪演算法選擇錯誤路徑的例子

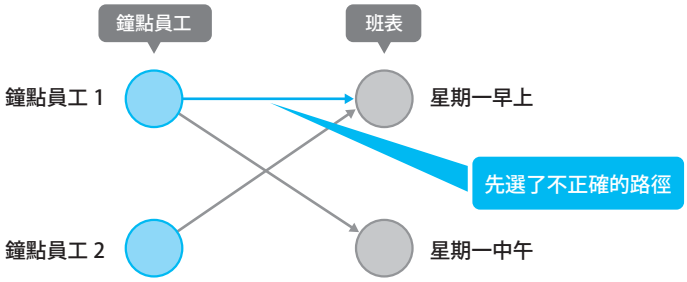


圖 5-6-1 的排班意願表是根據兩位鐘點員工意願排班的示意圖。

從**圖 5-6-2** 的正確排班範例可以發現，鐘點員工 1 被分配到星期一中午的時段，鐘點員工 2 被分配到星期一早上的時段，兩個人都按照意願分配到需要的時段。可是若使用貪婪演算法這種啟發式演算法，從鐘點員工 1 開始分配，就會出現**圖 5-6-3** 的結果，也就是鐘點員工 1 先分配到星期一早上的時段，導致鐘點員工 2 無法分配到需要的時段（換言之，很有可能導出與最佳解完全不同的答案）。

那麼該怎麼做，才能有效率地算出最大配對問題的最佳解呢？

目前已知的是，只要將 Graph Network 視爲「水路」，就能解決二分圖的最大配對問題。一開始可先將 Graph Network 畫成水路（**圖 5-6-4**），接著建立水源與出口（**圖 5-6-5**），之後再讓水從源頭流出，然後測量流水量，就能找到最佳路徑。這種計算最大流水量的問題稱爲「**最大流問題**」，只要能解決這種最大流問題，就等於能解決配對問題了。

圖 5-6-4 將排班意願表畫成水路的示意圖

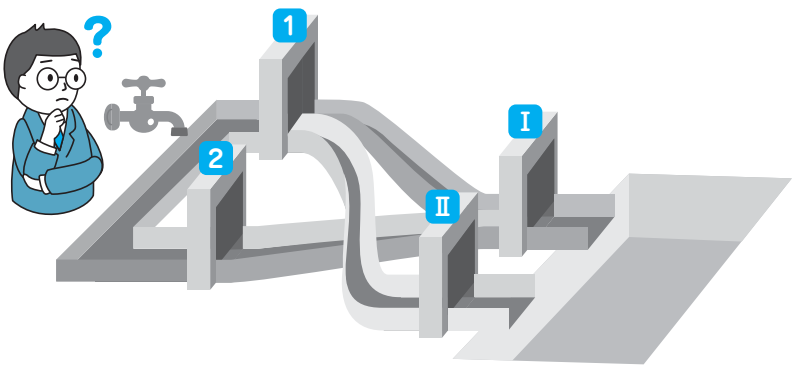
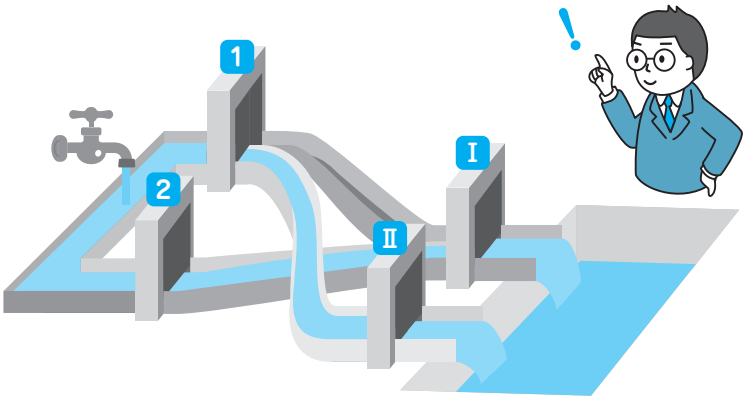


圖 5-6-5 透過讓水流入水路，求出最佳路徑的最大流問題示意圖



解決最大流問題的流程將於下列的圖說明。

圖 5-6-6 最左邊的頂點稱爲「**source**」，最右邊的頂點稱爲「**sink**」，而最大流問題就是試問從 source 流至 sink 的最大水量（flow）。

頂點之間的邊稱爲有向邊（流向固定的邊），箭頭指向的頂點稱爲「**子節點**」，位於反邊的頂點則稱爲「**父節點**」。各邊的值稱爲「**capacity**」，代表有多少水量可從這條水路經過。

只要解決最大流問題，就能在**圖 5-6-6** 的圖表中，如**圖 5-6-7** 一般，讓流至 sink 的總水量放至最大。當然也可以反過來觀察從 source 流出多少水量，但可根據每條水路能承受的水量，算出最終流至 sink 的總水量爲 12。

圖 5-6-6 解決最大流問題的圖表

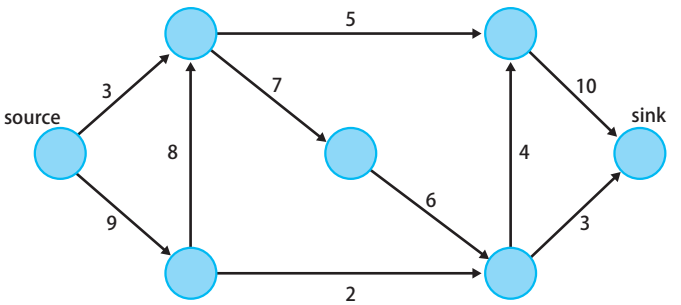
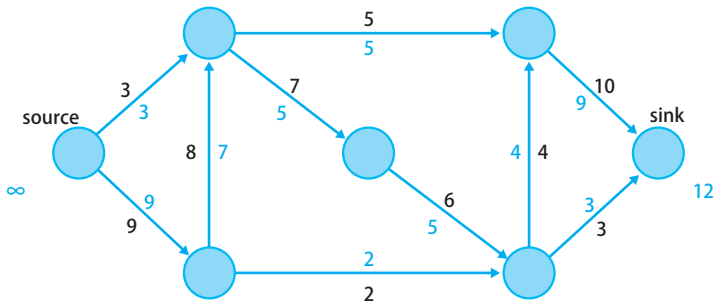


圖 5-6-7 最大流問題正解的水路



接著要透過「Dinic 的演算法」解說解決最大流問題的流程。可用來解決最大流問題的演算法還有「Ford Fulkerson 的演算法」、「Push Relabel 演算法」、「Push Relabel 演算法」以及其他的演算法。

Dinic 演算法的概要如下，箇中細節將留待 **5-7** 之後的章節介紹。

1. 先使用寬度優先搜尋（BFS）決定從 source 放水的順序（**5-7**）
2. 如果水無法從任何一條水路（邊）流至 sink，就結束運算，此時流至 sink 的總水量就是最大流（最佳解）
3. 利用深度優先搜尋（DFS）的方式，讓水根據步驟 1 決定的放水方式，從 source 流至 sink，接著依照水量減掉水路的 capacity，接著再反向拉出 capacity 相同的邊（**5-8**）
4. 回到步驟❶

從下一節開始，本書要介紹寬度優先搜尋（**5-7**）與深度優先搜尋（**5-8**），之後還要試著執行最大流問題的程式碼（**5-9**）。
最後要執行二分圖配對問題的程式碼（**5-10**），藉此了解配對問題的全貌。

8-2 深度學習的運作方式

8-1 已經先帶著大家體驗了透過深度學習辨識影像的過程，接著則要介紹深度學習是如何「學習」辨識的機制。一開始會介紹深度學習的原理，之後才會執行程式，加深相關的理解。

在了解深度學習的原理時，請大家先看一下**圖 8-2-1**。這張是將「在網路構造中學習與記憶」的深度學習原理的示意圖。假設偵測到某個影像具有「橘色」與「圓形」的特徵（**圖 8-2-2**），也有會對「橘色」、「圓形」這類特徵產生反應（發火）的元素（這類元素稱爲「細胞」）。

仔細觀察這個網路，會發現其中對「橘色」、「圓形」這類特徵產生反應的細胞，與對應「橘子」的細胞相連，所以發現「橘色」與「圓形」這兩種特徵後，與「橘子」對應的細胞就會產生反應（發火），也就能將該影像辨識爲「橘子」。

同理可證，「草莓」也是如**圖 8-2-3** 一樣，透過「紅色」與「三角形」的特徵辨識。

這種透過特徵記住「橘子」與「草莓」的學習網路稱爲神經網路，也是執行深度學習的基礎。

圖 8-2-1 學習辨識橘子與草莓的神經網路示意圖

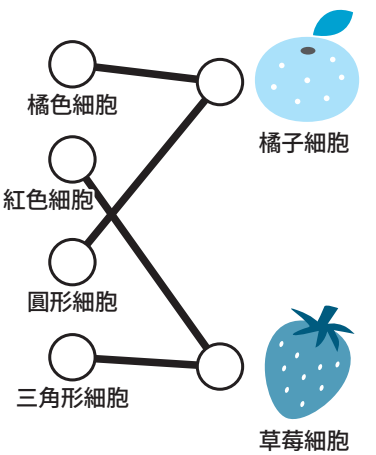


圖 8-2-2 神經網路辨識橘子的示意圖

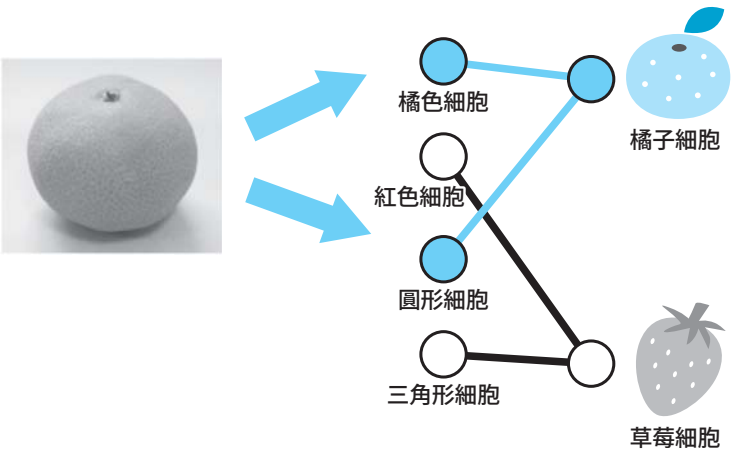
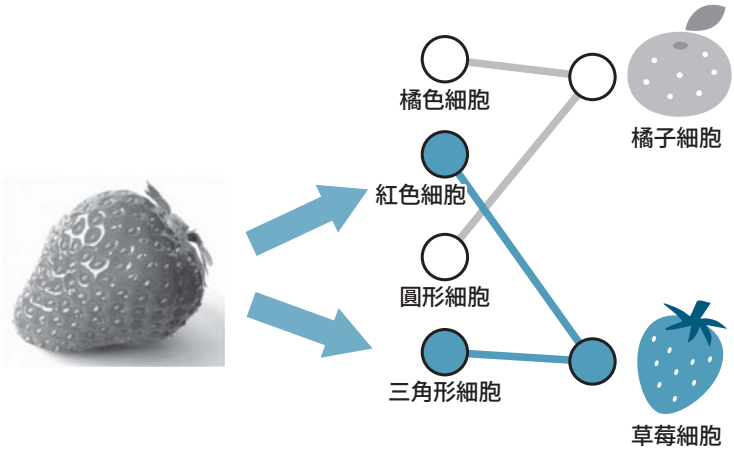


圖 8-2-3 神經網路辨識草莓的示意圖



經過上述的步驟之後，我們已經知道神經網路看到橘子或草莓的影像時，能夠正確將這些影像辨識爲「橘子」與「草莓」，不過前面的說明還有一些有待說明之處。

「辨識橘子與草莓的神經網路到底是如何學習的呢？」
「神經網路是如何辨識橘色或圓形這類特徵的呢？」
「橘子或草莓在不同的角度有不同的形狀，所以光憑橘色或圓形這類特徵怎麼能正確辨識，神經網路是如何辨識不同形狀的水果的呢？」

只要一一解開上述的疑問，就一定能更了解深度學習的機制，所以，先解決「辨識橘子與草莓的神經網路到底是如何學習的呢？」這個問題。

神經網路的學習機制與第 2 章介紹的機械學習的學習機制基本上相同。
請大家先看一下**圖 8-2-4**，其中介紹了神經網路學習前後的狀態與流程。
假設已將顏色或形狀這類資訊量化爲特徵，就能在座標空間配對顏色或形狀這些特徵，而此時爲學習前的狀態。
不過，這只是在座標空間完成配對的狀態，還無法說明橘子與草莓的特徵有何差異，所以在此時的神經網路中，所有的特徵與橘子或草莓的細胞都是平等地連接。接下來則是要慢慢地讓不同的特徵與橘子或草莓連接。

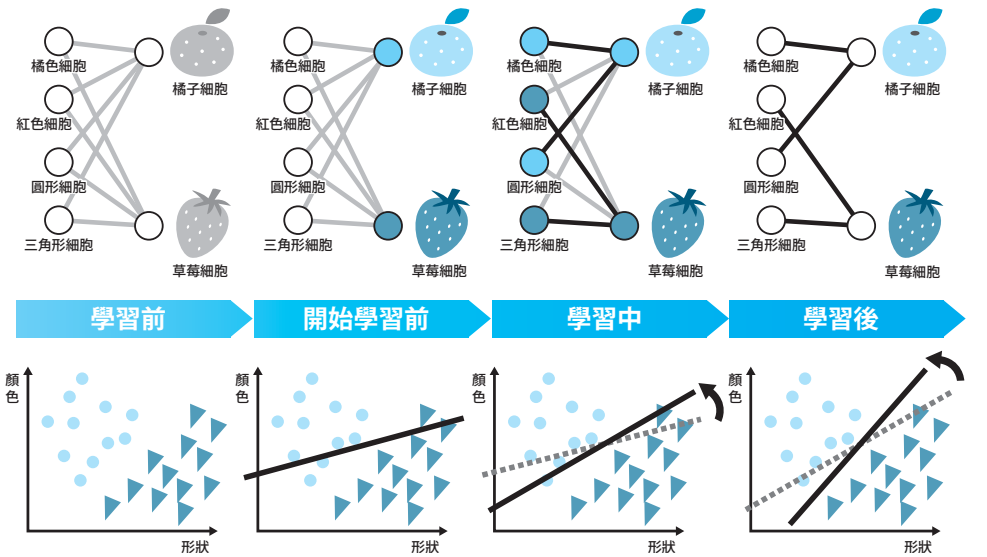
學習是在座標空間隨機畫出分界線之後開始。從圖中的「開始學習前」可以發現，與橘

子對應的橘色的●（預備了多張橘子圖片），以及與草莓對應的紅色的▼（一樣預備了多張草莓圖片）未被分界線一分爲二。

如果能在橘子與草莓之間畫出一條「正確的分界線」，將橘子與草莓一分爲二，代表學習完畢。讓分界線稍微旋轉角度（順時針或逆時針），直到可以將橘子與草莓分開時，代表這條分界線越來越接近「正確的分界線」，這時神經網路也會強化正確的特徵細胞與橘子或草莓之間的連結，以及弱化不正確的特徵細胞與橘子或草莓之間的連結。上述的流程可透過後續解說的「反向傳播演算法」實現。

學習完畢後，橘子與草莓將被分界線一分爲二，神經網路中的特徵與物體（橘子或草莓）也能正確配對。

圖 8-2-4 神經網路進行學習與記憶的示意圖



以上就是神經網路學習機制的概要。接下來要一邊解決「神經網路是如何辨識橘色或圓形這類特徵的呢？」、「橘子或草莓在不同的角度有不同的形狀，所以光憑橘色或圓形這類特徵怎麼能正確辨識，神經網路是如何辨識不同形狀的水果的呢？」這兩個問題，一邊帶著大家透過神經網路了解深度學習。

深度學習的一大特徵爲「能學習不同的版本」。以「橘子」爲例，圖 8-2-5 列出了橘子的各種樣貌，對我們人類來說，這些都是「橘子」。

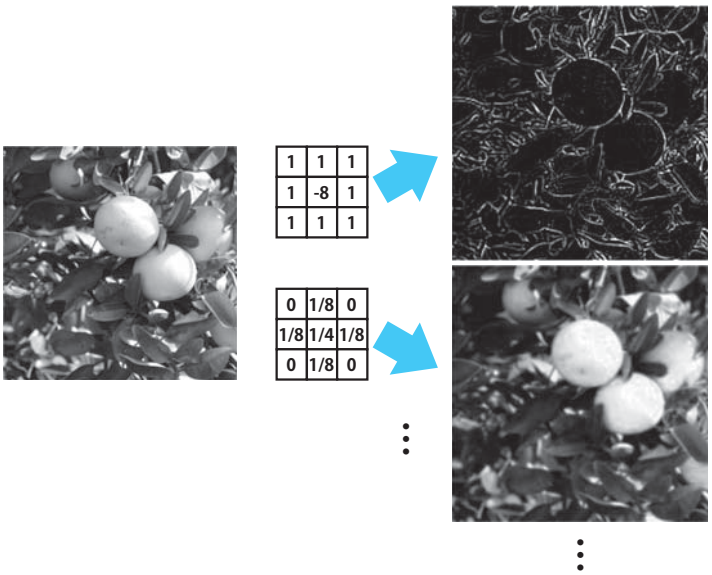
由於這些影像都具有自己的特徵，所以不能一概而論。如果能了解深度學習是如何辨識不同版本的物體，就能進一步了解深度學習的機制。

圖 8-2-5 橘子的各種樣貌

最具代表性的深度學習就是「**卷積神經網路（Convolutional Neural Network、CNN）**」。這種卷積神經網路會在影像套用「過濾器」（卷積），讓該影像的特徵浮現，再利用該特徵進行學習。如果能了解「卷積」的原理，就更有機會解決「神經網路是如何辨識（橘子的）橘色或圓形這類特徵的呢？」這個問題。

圖 8-2-6 爲卷積的示意圖。圖片是幾百萬個「像素」的集合體，而每個像素都以 RGB 的組合呈現。

圖 8-2-6 透過過濾器對影像進行「卷積」處理的示意圖



RGB 這三種顏色分別具有 0～255 的值，所以我們才能分辨像素的顏色。若能強化這些像素與相鄰像素的差異，就能突顯物體與物體之間的界線（邊緣）。
反之，若是「扁平化像素與周圍像素的差異」，圖片就會變得模糊。

像這樣透過強調或扁平化像素與周邊像素的差異處理圖片的過程就稱爲「**卷積**」，而執行這個處理的是「**過濾器**」。

在深度學習中，這個「過濾器」是主角，而深度學習的學習就是不斷地調整這個「過濾器」，以便正確區分「橘子」與「草莓」的處理。

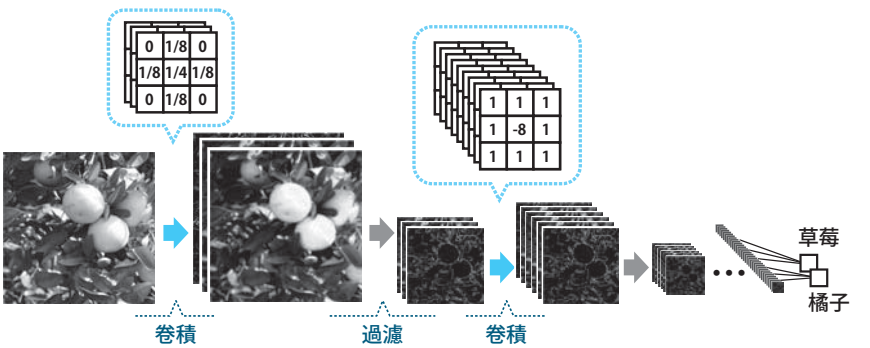
爲了進一步了解深度學習的學習過程，現在就來剖析CNN的神經網路構造（**圖8-2-7**）。深度學習的神經網路就是重複執行「**卷積**」與「**過濾**」，最後區分出「橘子」或「草莓」這類物體的「類別」的構造。所謂的「卷積」就是利用事先建立的過濾器進行的影像處理（更換影像），過濾則是壓縮影像的處理（通常會壓縮成1／2的大小）。「卷積」處理會強化像素周邊的特徵，而過濾處理則可壓縮影像，強調「更爲明顯的特徵」。

經過一層層的處理之後，不管是「細微的特徵」還是「明顯的特徵」都能一一浮現。要注意的是，執行卷積處理的過濾器是隨機設定的，所以要讓找到的特徵與橘子或草莓配對時，就如先前介紹的神經網路學習機制（**圖8-2-4**）所示，必須微調最後一層的過濾器，直到能正確區分橘子與草莓的「類別」爲止（也就是以旋轉**圖8-2-4**的分界線的方式微調過濾器）。

微調最後一層的過濾器之後，前幾層的過濾器也會跟著微調，最後所有的過濾器都會經過微調，一如**圖8-2-4**的分界線不斷調整角度一樣，過濾器的微調也是循序漸進的。

從**圖8-2-7**可以發現，越是位於右邊的階層，過濾器的數量越多。而過濾器的數量就是特徵的版本，假設有4個過濾器，代表能呈現四種橘子圖片。像這樣增加過濾器的數量，呈現橘子不同版本的特徵，正是深度學習的特徵。

圖8-2-7 深度學習的示意圖



最後要帶著大家確認實務的深度學習神經網路構造，以及透過兩相比對的方式進一步了解到目前爲止的說明。**圖8-2-8**是CNN中廣爲人知的**VGG16**神經網路。