第１５章　　人工智慧

人工智慧的研究起源甚早，1943 年 McCulloch 和 Pitts 即開始從事神經系統之研究而嘗試建構神經運作之模型。**人工智慧**（Artificial Intelligence，簡稱 AI）一詞係於 1956 年由 McCarthy 教授所提出；自此，人工智慧的研究即蓬勃發展 [Negnevitsky2002]。國內人工智慧的前輩孫家麟教授曾指出：intelligence 乃是由拉丁文 legere 演變而來，本意是蒐集和組合，並從事選擇、進而能瞭解和感受。人類若能製造出一種能夠蒐集、組合、進而選擇、瞭解、和感受的機器來，我們就有了人工智慧 [孫家麟 1985]。

人工智慧的研究領域涵蓋甚廣，其著稱者包含：電腦遊戲（computer game）、專家系統（expert systems）、自然語言處理（natural language processing）、模式識別（pattern recognition）、電腦視覺（computer vision）、機械學習（machine learning）、模糊理論（fuzzy theory）、基因演算法（genetic algorithms）、類神經網路（artificial neural networks）、資料採掘（data mining）、智慧型代理人（intelligent agents）、機器人（robotics）等。本章的主要目的是藉由一些有趣的問題，引導讀者瞭解人工智慧是如何來解決各種問題，進而引發讀者對人工智慧研究的興趣。

１５－１　解決困難問題

困難問題（puzzles）是需要智慧來解決的。因此，解決難題早期被視爲是人工智慧中一個重要的研究課題。例如：有一個人帶著一隻狼、一隻羊、與一粒甘藍在河的左岸想要渡河，岸邊有一艘船只能容納一人與一物。然而，若人不在其旁，則狼會吃掉羊；相同的，羊會吃掉甘藍。請問：這個人有可能帶著他的財產安全地渡河嗎？ [Hopcroft1979]

對於上述問題，我們可用 *M*、*W*、*G* 和 *C* 分別代表人、狼、羊和甘藍。由 *MWGC* 四個符號，可以形成 16 種不同組合。亦即：*MWGC*、*MWG*、*MWC*、*MGC*、*WGC*、*MW*、*MG*、*MC*、*WG*、*WC*、*GC*、*M*、*W*、*G*、*C*、和 ∅（空集合）。我們以短線 "-" 代表河，短線的左邊代表左岸，右邊代表右岸。例如，*WC-MG* 表示目前的**狀態**（state）是：在河的左岸有狼和甘藍，右岸則有人和羊。

剛開始時，人、狼、羊和甘藍都在河的左岸，所以**起始狀態**（initial state）爲 *MWGC-∅*。題意是希望人能想出一個方法，帶著狼、羊和甘藍安全地渡河到右岸，所以**終止狀態**（final state）應爲 *∅-MWGC*。從起始狀態開始分析，人可帶著其中一物或是不帶任何一物渡河，因此，下一個可能的狀態有下列四種：

- *GC-MW*：此人帶著狼渡河到右岸，左岸則留下羊與甘藍。根據題意，此時羊會吃掉甘藍，所以這個狀態是不安全的。
- *WG-MC*：此人帶著甘藍渡河到右岸，左岸留下狼與羊。此時狼會吃掉羊，所以這個狀態也是不安全的。
- *WC-MG*：此人帶著羊渡河到右岸，左岸留下狼與甘藍。此時羊與甘藍都不會被吃掉，所以這個狀態是安全的。
- *WGC-M*：此人獨自渡河到右岸，左岸留下狼、羊與甘藍。此時，甘藍可能被羊吃掉，或羊被狼吃掉，所以這個狀態也是不安全的。

根據以上的分析，*WC-MG* 是唯一安全的狀態，我們可以從此一安全狀態繼續渡河。此時，下一個狀態又有下列兩種可能：

- *MWGC-∅*：此人又帶著羊回到左岸，與起始狀態一樣，因此形成一個迴圈。若是一直繞著這個迴圈渡河，那麼問題就無法解決。
- *MWC-G*：此人獨自回到左岸，將羊單獨留在右岸。此時羊與甘藍都不會被吃掉，所以這個狀態是安全的。

繼續以上的分析，我們可以得到如圖 15-1 的**狀態轉變圖**（state transition diagram），其中以橢圓表示狀態，狀態之間以箭頭連接，表示從一個狀態轉變到另一個狀態，箭號上標示有 *w*、*g* 或 *c*，表示此人帶著狼、羊或甘藍渡河，若爲 *m*，表示此人單獨渡河。起始狀態以 Start 箭頭表示，最後終止狀態以雙圈表示。由圖 15-1 可得知，從起始狀態到終止狀態間有路可行，表示此人可以從左岸帶著他的狼、羊與甘藍安全地渡河到右岸，整條路徑即表示其渡河的過程。

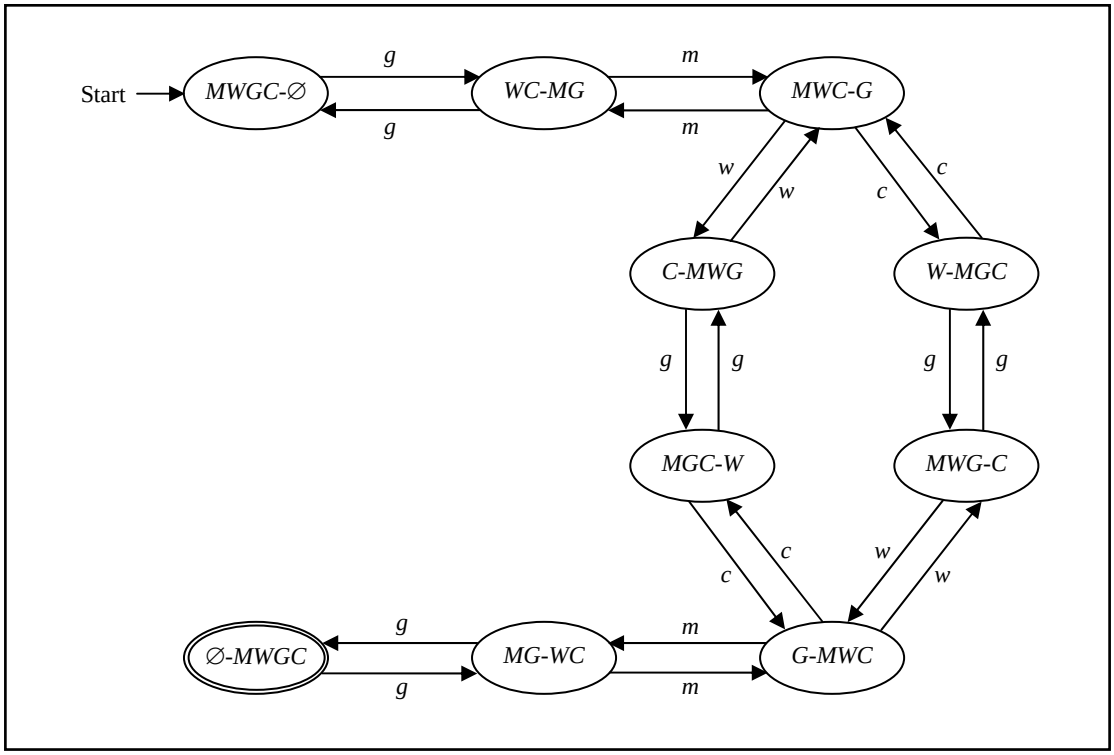


圖 15-1　人、狼、羊、與甘藍問題的狀態轉變圖

１５－２　電腦遊戲

下圍棋、象棋、或西洋棋等，都是需要高度智慧的，所以**電腦遊戲**（computer game）也是人工智慧中一項相當吸引人的研究。1997 年，IBM 的超級電腦**深藍**打敗西洋棋王，令人工智慧研究者相當振奮 [DeepBlue2005]。在本節我們以簡單的**井字遊戲**（Tic-Tac-Toe）來說明人工智慧如何來分析電腦遊戲 [Nilsson1998] 。

井字遊戲是雙人玩的，在畫有“井”字的九個空格裡，一方畫 X，一方畫 O。畫 X 的一方先下，誰先完成一列、一行、或對角線，誰就贏。在一般雙人對奕的棋戲中，人工智慧大都採用**極大極小法**（minimaxing method）來評估下一步棋應如何走。極大極小法是以目前的盤局，考慮所有可能可以走的棋步；對於每一個可能的新盤局，也考慮所有可能可以走的棋步；如是一直推衍下去，可以考量到很多步之後所有可能的盤局。當然，考量的棋步越多，勝算也就越大，然而計算量也隨著激增。我們需要設計一個**評估函數**（evaluation function）來評估每一個盤局對雙方的勝算如何。假設畫 X 的一方定爲極大（*MAX*），畫 O 的一方定爲極小（*MIN*），那麼在設計評估函數時，所得到的數值如果越大，表示對 X 的一方就越有利；相反的，如果評估函數所得到的數值越小，表示對 O 的一方就越有利。此種設計有一前提，那就是假設雙方都會朝向對自己有利的棋步走。

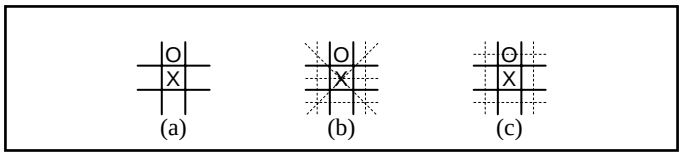


圖 15-2　盤局評估函數的計算

對於井字遊戲，假設目前考慮的盤局爲 *p*，我們可以設計一個簡單的評估函數 *e*(*p*) 如下：假如盤局 *p* 仍無法分出勝負，則 *e*(*p*) =（*MAX* 仍有可能完成的行數、列數、與對角線的總和）–（*MIN* 仍有可能完成的行數、列數、與對角線的總和）。（此處，我們以縱向爲行，橫向爲列。）假如於盤局 *p* 獲勝一方已確定爲 *MAX*，則 *e*(*p*) = ∞；相反的，假如於盤局 *p* 獲勝一方已確定爲 *MIN*，則 *e*(*p*) = – ∞。因此，假若盤局 *p* 如圖 15-2(a)所示，其評估函數值 *e*(*p*) = 6 – 4 = 2。因爲對 *MAX*（即 X 的一方）而言，尚有可能完成第一行或第三行，第二列或第三列，還有兩條對角線，合起來共有 6 種可能完成的連線，如圖 15-2(b)所示。同理，對 *MIN*（即 O 的一方）而言，只有 4 種可能完成的連線，如圖 15-2(c)所示。

爲了要減少盤局的個數，以減少評估函數的計算，對稱的盤局都視爲相同。例如：圖 15-3 中的 4 個盤局都視爲相同，因爲盤局 (b)、(c)、與 (d) 分別以所畫虛線而與盤局 (a) 相對稱。

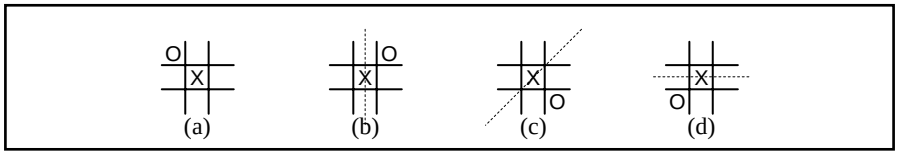


圖 15-3　對稱的盤局

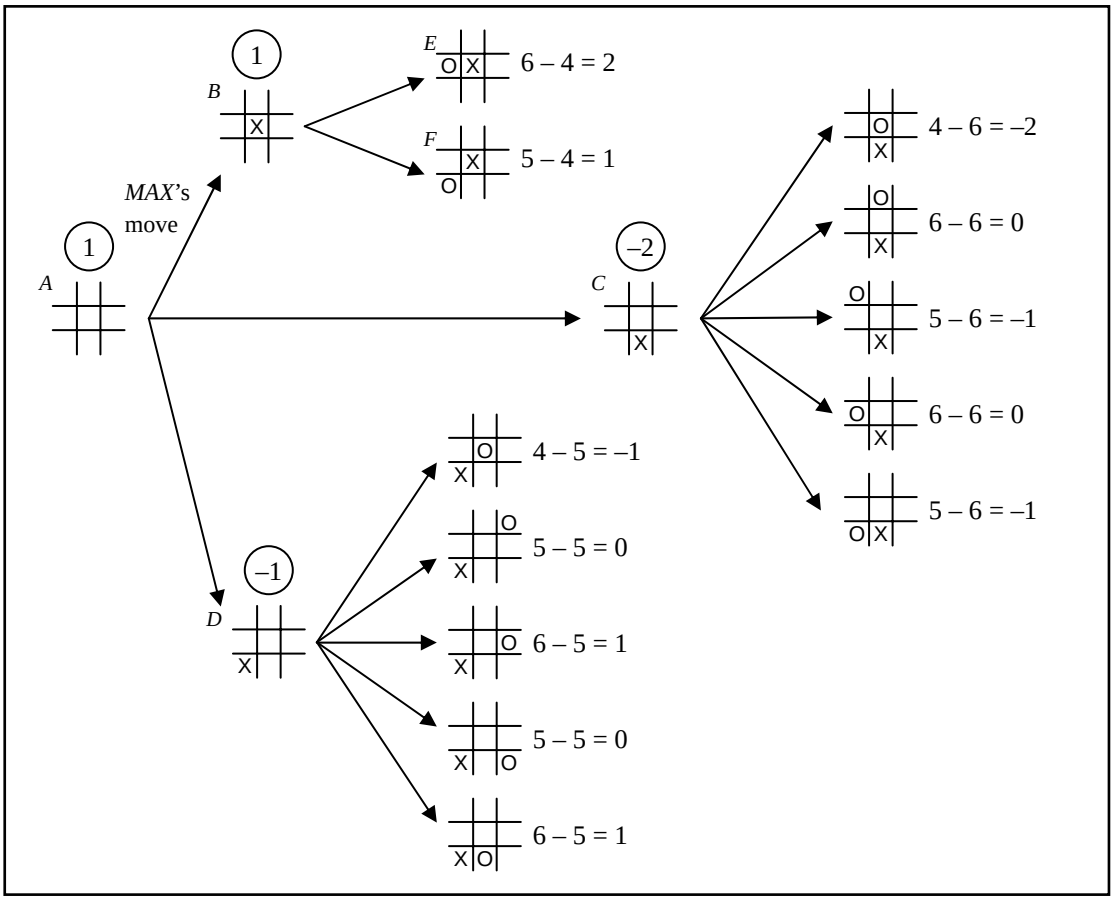


圖 15-4　井字遊戲前兩步所有盤局形成一樹狀結構

在圖 15-4，我們從空白的棋盤開始，考量下前兩步所有可能的盤局，評估函數值就列在其右。*MAX* 由空白的盤局 *A* 開始先走一步，可能成爲盤局 *B*、*C*、或

D。若爲盤局 *B*，改由 *MIN* 走下一步，可能成爲盤局 *E* 或 *F*，其評估函數值分別爲 2 和 1。對 *MIN* 而言，選擇有較小評估函數值的盤局對其較爲有利。因此，盤局 *B* 的評估值爲 1。同理，盤局 *C* 和 *D* 的評估值分別爲 –2 和 –1。最後，*MAX* 從盤局 *B*、*C*、和 *D* 中選取有較大的評估值對其較爲有利。因爲盤局 *B* 的評估值最大，故而盤局 *A* 的評估值爲 1，意即 *MAX* 走一步到盤局 *B* 的勝算最大。對於 *MAX* 或是 *MIN* 任何一方有利的棋步分數（即評估值），我們以圓圈將其圈起來。假設 *MAX* 根據上述評估，畫了 X 於井字的中間（即盤局 *B*），並且 *MIN* 在 X 的左邊畫了 O 來回應（這一步對 *MIN* 來說並不是好步）。接著 *MAX* 又往下考慮了兩步，得到如圖 15-5 的結果。

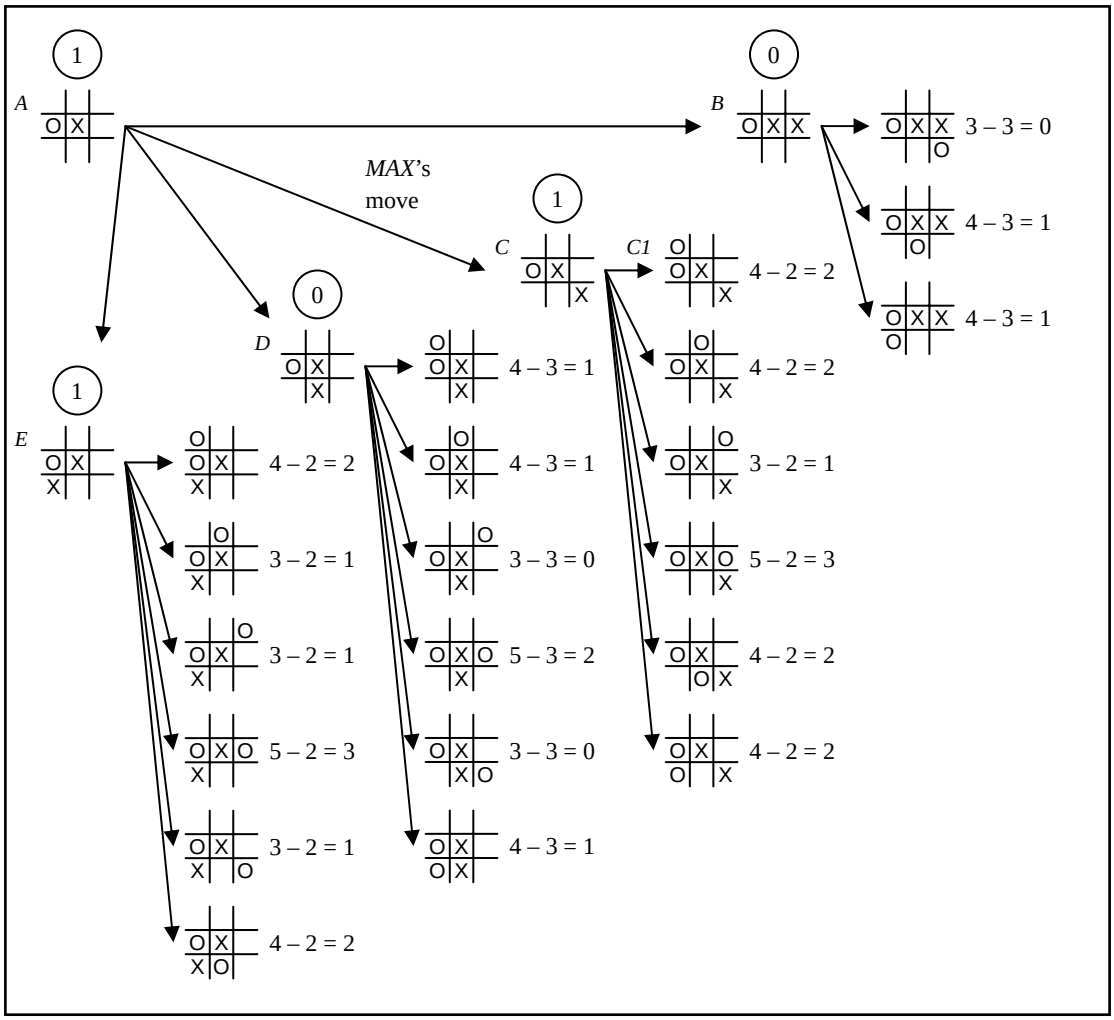


圖 15-5　井字遊戲 *MAX* 一方第二次考量

此時，有兩個棋步（圖 15-5 中的盤局 *C* 和 *E*）都得到相同的最好評估值（其值爲 1）。假設 *MAX* 如圖所標示，走到盤局 *C*。此時，*MIN* 爲了要避免立即的危險，他會在左上角畫上 O，得到盤局 *C1*。*MAX* 繼續再往下考慮兩步，得到如圖 15-6 的結果。*MAX* 的下一步，盤局 *B*、*C*、*D*、和 *E* 的評估值均爲 –∞，表示這些棋步可能會讓 *MIN* 立即贏了這盤棋。唯獨盤局 *F* 的評估值爲 1，表示還有勝算。聰明的你應該知道，X 要畫在哪裡可以獲勝。

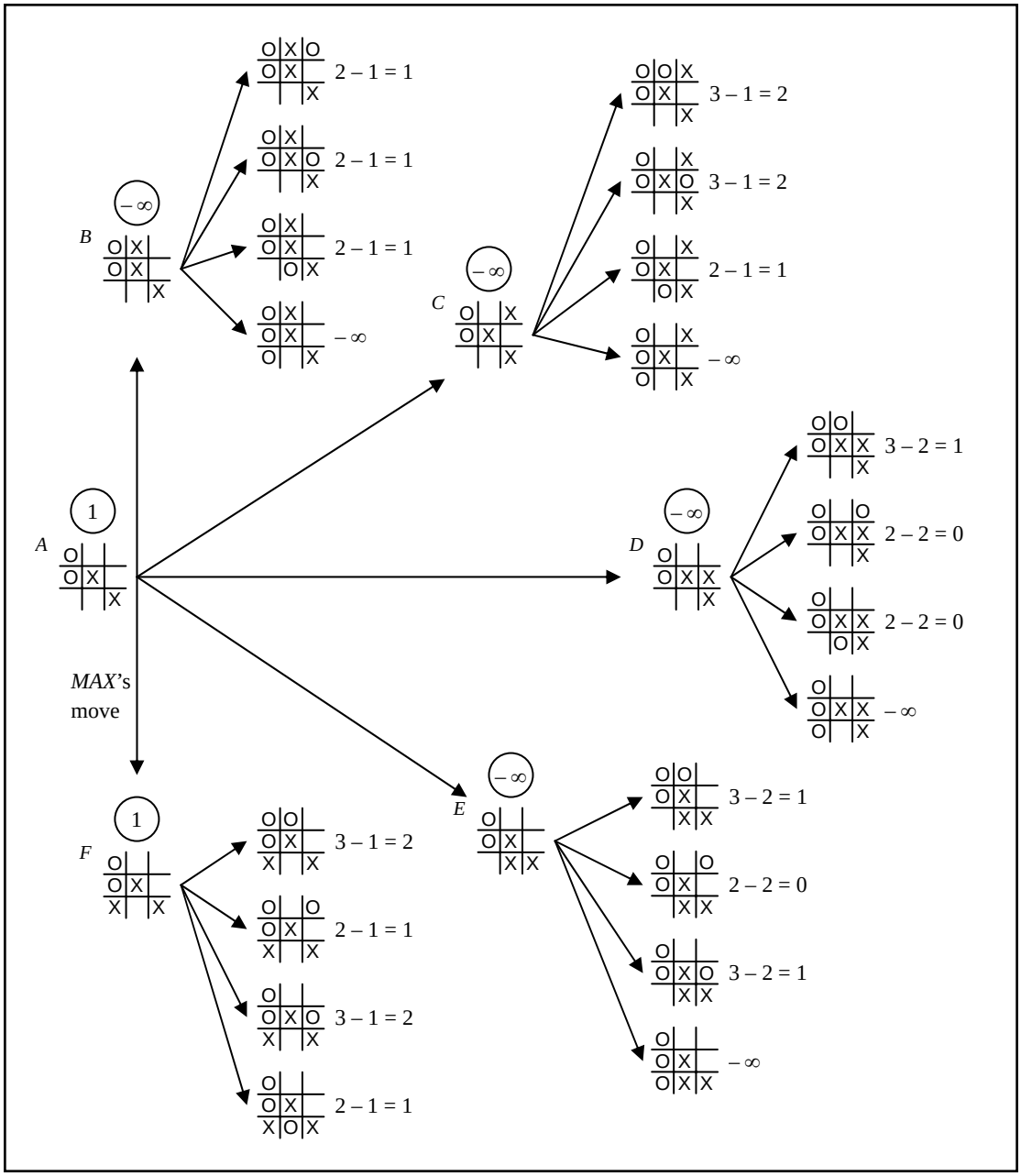


圖 15-6　井字遊戲 *MAX* 一方第三次考量

１５－３　專家系統

專家系統可算是在 1970 年代人工智慧研究中，最爲成功、也是應用最爲廣泛的領域。如圖 15-7 所示，一個專家系統基本上包含有**知識庫**（knowledge base）與**推論機制**（inference engine）等兩個部分。知識庫內儲存著某一知識範疇內的**專家知識**（expertise），並以某種**知識表達法**（knowledge representation）來表示。當使用者輸入一些已知的**事實**（facts）後，推論機制就以知識庫內的專家知識和所輸入的事實，依據數學邏輯推導出結論，並告知使用者。

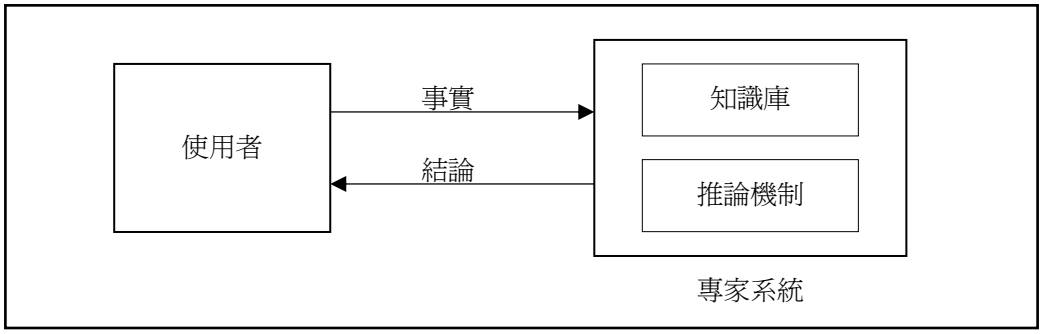


圖 15-7　專家系統基本概念

早期著名的專家系統有疾病診斷的 MYCIN 系統，由光譜來分析化學成分的 DENDRAL 系統，分析地質資料以探勘石油的 DIPMETER 系統、以及探勘礦藏的 PROSPECTOR 系統，和電腦系統組合的 XCON/R1 系統。其中以 MYCIN 系統最爲著名且最重要，其原因有三：(1) 它證實了人工智慧可以被用來解決真實世界中實際的問題；(2) 它提供了一個測試平台以測試一些新的觀念，例如：解釋、自動擷取知識、和智慧型教學等功能；(3) 它展示了專家系統**骨架**（shell）的實用性，可更簡易地建構一個專家系統 [Giarratano2004]。

知識表達法最常用到的、也是最簡單的方式爲**規則**（rules），其形式如下：

If　前提 1　前提 2　…　前提 n
Then 結論 1　結論 2　…　結論 m

其中，n 和 m 均大於 0。規則也可以用圖形方式表示，如圖 15-8 所示。其中，**前提**（antecedents）部分包含規則成立的條件，它可爲已知的事實或是其它規則的**結論**（consequents）。對於以規則方式表達的知識而言，其推論方式可分爲兩種：一種是**前向鍊結**（forward chaining），另一種是**後向鍊結**（backward chaining）。前向鍊結是依據已知的事實與先前所推導出的結論，檢視是否符合某規則的前提部分，若符合則可推得結論；一般用於從觀察到的事實中推導出可能的結論。至於後向鍊結則是由結論開始反推回去，透過規則得知需要哪些前提，才能推得結論；一般用於**查詢**（queries）時使用。

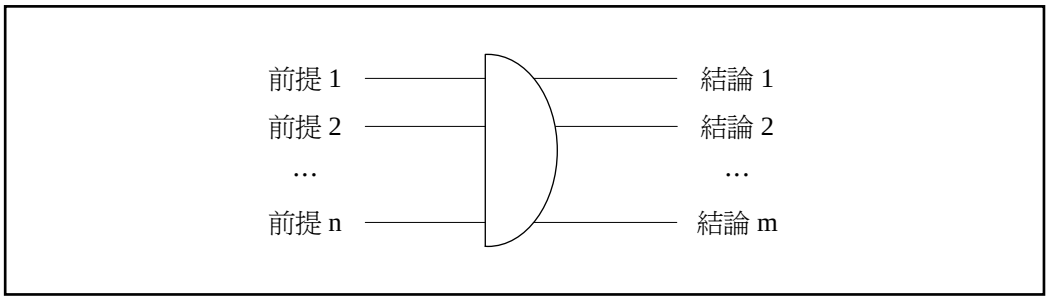


圖 15-8　規則的圖形表示法

舉例來說，假設我們有下列規則：

If ?*x* 是肉食性動物
?*x* 是金黃色
?*x* 有黑色條紋
Then ?*x* 是老虎

其中，?*x* 爲一變數，可以代入任何值。此規則意爲：有一動物，若滿足三項條件：肉食性動物，金黃色，且有黑色條紋，則可推論得知出：牠是老虎。假若我們已知有一隻動物名叫『阿丹』，牠滿足上列三項條件，根據前向鍊結，我們可以推得：阿丹是一隻老虎。假若我們要問：阿丹是否爲一隻老虎？根據後向鍊結，我們必須要進一步檢視看看，目前已知的事實與所推導出來的結論，是否滿足前提部分的三項條件。

Z1: **If** ?*x* 有頭髮
Then ?*x* 是哺乳類動物

Z2: **If** ?*x* 會產奶
Then ?*x* 是哺乳類動物

Z3: **If** ?*x* 有羽毛
Then ?*x* 是鳥類

Z4: **If** ?*x* 會飛
?*x* 會生蛋
Then ?*x* 是鳥類

Z5: **If** ?*x* 是哺乳類動物
?*x* 吃肉
Then ?*x* 是肉食性動物

Z6: **If** ?*x* 是哺乳類動物
?*x* 有尖銳的牙齒
?*x* 有爪
?*x* 有銳利的眼睛
Then ?*x* 是肉食性動物

Z7: **If** ?*x* 是哺乳類動物
?*x* 有蹄
Then ?*x* 是有蹄類動物

Z8: **If** ?*x* 是哺乳類動物
?*x* 會反芻
Then ?*x* 是有蹄類動物

Z9: **If** ?*x* 是肉食性動物
?*x* 是金黃色
?*x* 有深色斑點
Then ?*x* 是豹

Z10: **If** ?*x* 是肉食性動物
?*x* 是金黃色
?*x* 有黑色條紋
Then ?*x* 是老虎

Z11: **If** ?*x* 是有蹄類動物
?*x* 有長腳
?*x* 有長脖子
?*x* 是金黃色
?*x* 有深色斑點
Then ?*x* 是長頸鹿

Z12: **If** ?*x* 是有蹄類動物
?*x* 是白色
?*x* 有黑色條紋
Then ?*x* 是斑馬

Z13: **If** ?*x* 是鳥類
?*x* 不會飛
?*x* 有長腳
?*x* 有長脖子
?*x* 是黑白相間
Then ?*x* 是鴕鳥

Z14: **If** ?*x* 是鳥類
?*x* 不會飛
?*x* 會游泳
?*x* 是黑白相間
Then ?*x* 是企鵝

Z15: **If** ?*x* 是鳥類
?*x* 很會飛
Then ?*x* 是信天翁

圖 15-9 羅碧所使用辨識動物種類的知識庫

假設有個機器人名叫『羅碧』，要到動物園去參觀。它只能觀察到動物簡單的**特徵**（features），例如：顏色、大小、以及動物是否有頭髮、會不會產奶等，但

是它沒有足夠的能力去識別動物的種類。因此，我們可以商請生物學家來幫羅碧建構一個專家系統，使它能夠利用所觀察到的基本特徵來辨識動物的種類。假設動物園內只有七種動物：虎、豹、長頸鹿、斑馬、鴕鳥、企鵝、與信天翁。此專家系統以規則的方式來建構其知識庫，其中包含有十五條規則 Z1 到 Z15，如圖 15-9 所示，並利用前向鍊結的推論機制來辨識動物的種類 [Winston1992]。

假設羅碧看到一隻動物名叫『小葉』，有如下特徵：(1) 有頭髮，(2) 會反芻，(3) 有長腳，(4) 有長脖子，(5) 是金黃色，和(6)有深色斑點。因爲小葉有頭髮，根據規則 Z1，我們可以得知：小葉是哺乳類動物。因爲小葉是哺乳類動物，而且會反芻，根據規則 Z8，可以推論得知：小葉是有蹄類動物。至此，規則 Z11 前提部分的五個條件都符合，最後可以得知：小葉是長頸鹿。整個推論過程可以用圖 15-10 表示，其中黑色方形表示爲所觀察到的特徵，白色方形爲經過前向鍊結所推得的結論。

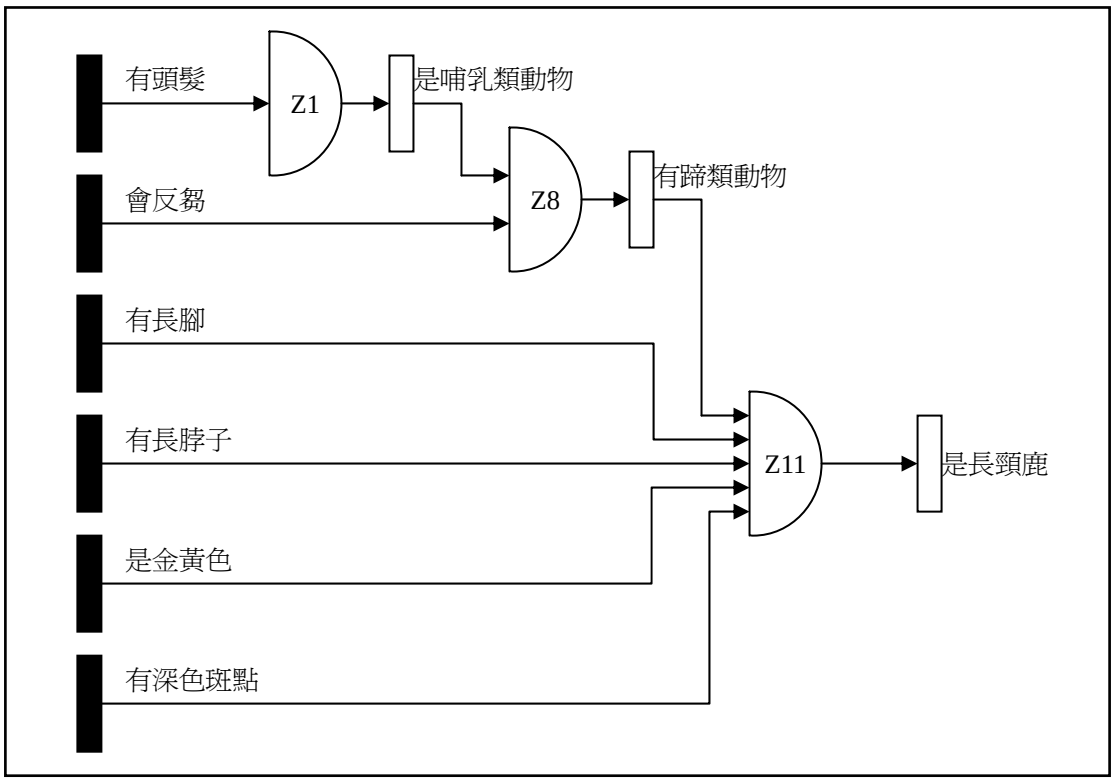


圖 15-10 推論『小葉是長頸鹿』的整個過程

１５－４　自然語言處理

自然語言（natural languages）指的是人類所使用的語言，如中文、英文等，用以區別**電腦程式語言**（computer programming languages），如 C、C++、Java 語言。自然語言的處理包含**語彙**（lexical）分析、**語法**（syntactical）分析、和**語意**（semantic）分析等三個層面。在一個英文句子中，語彙的分析是要將其**單字**（words）切割出來。語法的分析是進一步檢視單字的順序，是否符合**文法**（grammar）結構，而語意的分析則更進一步嘗試了解句子的**意義**（meaning）。

假設我們要分析 “Bill printed the file” 這個句子。在英文中，語彙分析較爲簡單，因爲英文單字都用空白或標點符號隔開，所以要切割單字出來就比較容易。上述例句，我們可以切出 Bill、printed、the、和 file 四個單字。至於語法的分析，我們必須先給定英文的文法。例如：一個英文句子（sentence，簡稱 S）可以由名詞片語（noun phrase，簡稱 NP）後接動詞片語（verb phrase，簡稱 VP）所組成，可以表達成如圖 15-11 中規則 R1 的形式 [Rich1991]。

R1: S → NP VP
R2: NP → the NP1
R3: NP → PRO
R4: NP → PN
R5: NP → NP1
R6: NP1 → ADJS N
R7: ADJS → ε | ADJ ADJS
R8: VP → V
R9: VP → V NP
R10: N → file | printer
R11: PN → Bill
R12: PRO → I
R13: ADJ → short | long | fast
R14: V → printed | created | want

圖 15-11　部分英文文法

同理，一個名詞片語可以是定冠詞 the 後接另一個新名詞片語（NP1）、或是代名詞（pronoun，簡稱 PRO）、或是專有名詞（proper noun，簡稱 PN）、或是新名詞片語（NP1），其規則爲 R2 到 R5。而新名詞片語（NP1）則是由形容詞串（adjectives，簡稱 ADJS）後接名詞所構成，其規則爲 R6。而形容詞串則可以不包含任何形容詞（以 ε 表示），或是由形容詞（adjective，簡稱 ADJ）後接形容詞串（ADJS）所構成，其規則爲 R7，其中，垂直線（"|"）代表『或是』的意思。相同地，動詞片語可以只包含動詞（verb，簡稱 V），或是由動詞後接名詞片語所組成，其規則爲 R8 和 R9。假設一個迷你字典中包含有名詞 file 和 printer，專有名詞 Bill，代名詞 I，形容詞 short、long、和 fast，動詞則有 printed、created、和 want 等，其可以規則形式表達如圖 15-11 中 R10 到 R14。

根據以上的簡單英文文法，上述例句 "Bill printed the file" 可以被分析成樹狀結構，稱爲**剖析樹**（parse tree），如圖 15-12 所示。語法分析有兩種方式：**由上而下剖析**（top-down parsing）和**由下而上剖析**（bottom-up parsing）。由上而下的語法分析是從句子（S）開始，根據文法規則，由其箭頭右邊的 NP VP 所取代，NP 又由 PN 所取代，而 PN 爲 Bill 所取代；接著 VP 由 V NP 所取代，V 爲 printed 所取代，NP 爲 the NP1 所取代，NP1 又爲 ADJS N 所取代，而 ADJS 爲 ε ，最後 N 爲 file 所取代。取代完後的字串爲 "Bill printed the file"，剛好和輸入例句一樣，所以語法剖析成功，代表語法正確。

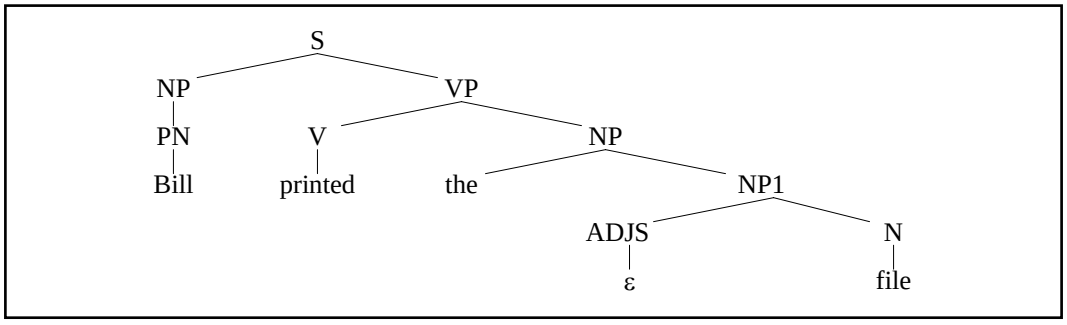


圖 15-12 例句 "Bill printed the file" 的剖析樹

由下而上的剖析是由輸入的句子開始分析，首先根據文法，Bill 可以由其左邊的 PN 取代，PN 又由 NP 所取代；接著 printed 由 V 所取代，file 由 N 所取代，ε 由 ADJS 所取代。然後，ADJS 和 N 由 NP1 所取代，接著，the NP1 又由 NP 所取代，再來，V NP 又由 VP 所取代，最後 NP VP 由 S 所取代。至此，整個句子剛好符合文法規則定義，代表句子的文法無誤。

至於語意的分析則較爲複雜與困難，因爲自然語言中常常含有混淆不清的語意，必須藉由**前後文**（context）才能確定真正的意義。例如：英文句子 “I saw a man on the street”，其中介詞片語 “on the street” 可以當作主詞補語或受詞補語。若爲主詞 “I” 的補語，其意應爲：我在街上看到一個人。若爲受詞 “a man” 的補語，其意應爲：我看到一個人在街上。如何知道介詞片語是主詞補語或是受詞補語，必須由前後文才能判知，因此處理起來較爲複雜。

在早期的自然語言處理中，ELIZA [Weizenbaum1966] 是一個非常有名的程式，它模擬心理治療家 Carl R. Rogers 的精神治療法。當使用者輸入任意一個句子，電腦會以它自己的句子回答，如圖 15-13(a) 所示爲使用者與 ELIZA 的部分對話，其中 Person 代表使用者輸入的句子，而 ELIZA 代表電腦的回答 [孫家麟 1985，Rich1991]。

ELIZA 是由兩個部分所組成，一爲主程式，另一部分 Weizenbaum 視之爲腳本（script）的知識庫，它是一組規則所組成，圖 15-13(b) 爲 ELIZA 的部分規則，其中，?*x* 和 ?*y* 爲變數。假若我們有 E1 和 E2 兩條規則，當使用者輸入句子 “My brother is smart”，ELIZA 主程式會使用**樣板比對**（pattern matching）技巧來比對上述規則箭號左邊的部分。對於第一條規則，My 與 My 對應，brother 屬於 family-member，也可以對應，is 與 is 對應，smart 對應變數 ?*y*。因此，變數 ?*y* 的值變成爲 smart。對於第二條規則，My 對應變數 ?*x*，brother 對應 family-member，

剩下的句子 “is smart” 對應變數 ?*y*。以上輸入的句子，可以成功地匹配兩條規則，因此可用規則右邊的句子回答 “Who else in your family is smart?” 或 “Tell me more about your family.”。若是沒有可以比對的規則，那麼有一些預設的句子可用來回答，例如：“Please go on.” 或 “I am not sure I understand you fully.”，以繼續交談。

Person: Men are all alike.
ELIZA: In what way?
Person: They’re always bugging us about something or other.
ELIZA: Can you think of a specific example?
Person: Well, my boyfriend made me come here.
ELIZA: Your boyfriend made you come here.
Person: He says I’m depressed much of the time.
ELIZA: I am sorry to hear you are depressed.
Person: It’s true. I am unhappy.
ELIZA: Do you think coming here will help you not to unhappy?
Person: I need some help, that much seems certain.
ELIZA: What would it mean to you if you got some help?
Person: Perhaps I could learn to get along with my mother.
ELIZA: Tell me more about your family.

(a) 與 ELIZA 的部分對話

E1:	(My {family-member} is ?*y*)	→	(Who else in your family is ?*y*)
E2:	(?*x* {family-member} ?*y*)	→	(Tell me more about your family)
E3:	(?*x* me ?*y*)	→	(?*x* you ?*y*)
E4:	(I remember ?*x*)	→	(Why do remember ?*x* just now?)

(b) ELIZA 的部分規則

圖 15-13　與 ELIZA 的部分對話與部分規則

15－5　機械學習

機械學習（machine learning）的目的乃在模擬人類的學習能力。最簡單的學習即是背誦，將所要學習的事物記下來，例如背書；此類學習可用資料庫等技術

將事物記錄下來，方便以後查詢。進階的學習，可從一些**事例**（cases）中，歸納出經驗法則，此為人類智慧的更進一步表現。

機械學習可分為**符號式學習**（symbolic learning）和**連結式學習**（connectionist learning）。符號式學習是將事例依其特徵以文字或符號方式表示，然後嘗試取出其分類規則。而連結式學習則是將事例的特徵用數值形式表示，然後用類神經網路的機制來學習其分類規則。類神經網路的學習機制於 15-6 節介紹，而本節中介紹符號式學習機制。

假設你到海灘戲水，發現有些人會被太陽曬傷，有些人卻不會，因此你很好奇地想去研究：造成曬傷的原因是什麼？於是，你拿起筆來，簡單地就你所觀察到的一些特徵記錄下來，如表 15-1 所示。所觀察到的簡單特徵包含有：頭髮顏色、身高、體重、是否有塗抹防曬乳液，與最後是否有曬傷的結果。於是你想從所觀察到的 8 個**樣本**（samples）中，嘗試得到曬傷的原因 [Winston1992]。

表 15-1　曬傷研究觀察紀錄

名字	髮色	身高	體重	乳液	曬傷
Sarah	金色	普通	輕	無	是
Dana	金色	高	普通	有	否
Alex	棕色	矮	普通	有	否
Annie	金色	矮	普通	無	是
Emily	紅色	普通	重	無	是
Pete	棕色	高	重	無	否
John	棕色	普通	重	無	否
Katie	金色	矮	輕	有	否

在符號式學習中，建構一個最小的**識別樹**（identification tree）用以區分曬傷與否，是一個簡單的方法，並且可以歸納出曬傷的規則。識別樹為一樹狀結構，

其節點（node）爲特徵，其分支（branch）的標籤爲特徵值，最後樹的終端節點（terminal node，即沒有分支的節點）應只包含同一類的樣本。要建構一個最小的識別樹，需要借用資訊理論（information theory）中的**熵**（entropy）的概念，其定義如下：

$$\sum_c -\frac{n_{bc}}{n_b}\log_2\frac{n_{bc}}{n_b}$$

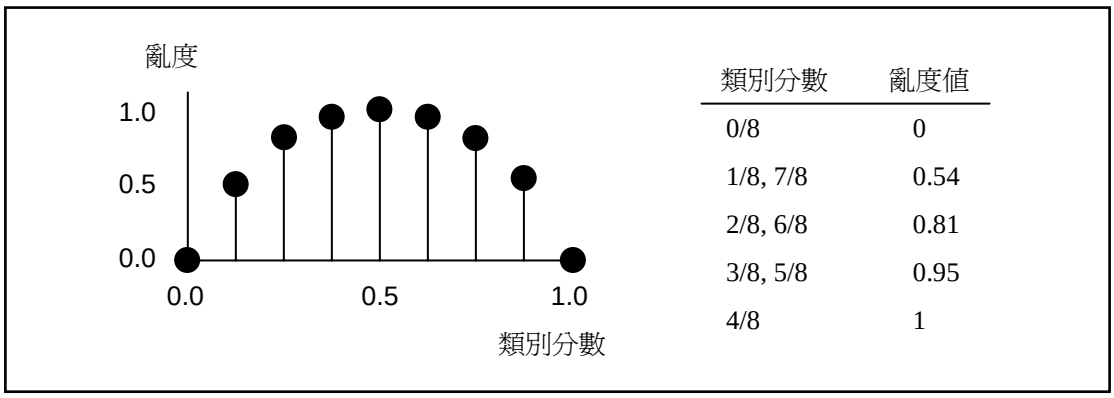
其中，n_b 是在分支 b 的樣本數，n_{bc} 是在分支 b 中類別爲 c 的樣本數。熵的定義可用爲測量物件的**亂度**（disorder）。假設目前分支 b 只有兩個類別 A 和 B，它們的樣本數都一樣，則其亂度值爲 1，是亂度的最大值，其計算如下：

$$\sum_c -\frac{n_{bc}}{n_b}\log_2\frac{n_{bc}}{n_b}=\left(-\frac{1}{2}\log_2\frac{1}{2}\right)+\left(-\frac{1}{2}\log_2\frac{1}{2}\right)=\left(\frac{1}{2}\right)+\left(\frac{1}{2}\right)=1$$

相反的，假設目前分支 b 只剩下一個類別，只剩下 A 類或是只剩下 B 類，那其亂度值爲 0，是亂度的最小值，其計算如下：

$$\sum_c -\frac{n_{bc}}{n_b}\log_2\frac{n_{bc}}{n_b}=(-1\log_2 1)+(-0\log_2 0)=(0)+(0)=0$$

如圖 15-14 所示，當從兩個類別樣本數相同逐漸移至只剩下一個類別時，其亂度值慢慢地由 1 變成 0。其中，**類別分數**爲某類別的樣本數除以所有樣本總數，如上式中的 n_{bc}/n_b。



類別分數	亂度值
0/8	0
1/8, 7/8	0.54
2/8, 6/8	0.81
3/8, 5/8	0.95
4/8	1

圖 15-14　兩個類別不同比例相混的亂度分佈情形

得到每一個分支的亂度值後，可以進一步計算由一個節點其所生出所有分支的平均亂度，其公式如下：

$$\sum_b \frac{n_b}{n_t} \times d_b$$

其中，d_b 爲分支 b 的亂度，n_t 爲所有分支的樣本數總和，n_b 如前所定義爲分支 b 的樣本數。在建構一個最小的識別樹時，我們分別以不同的特徵爲節點，計算其節點的平均亂度值，最小的平均亂度值代表分類結果較爲一致，因此應可得較小的識別樹。我們再繼續求取下一層的最小平均亂度值，一直到節點中只包含一個類別（其亂度值爲 0）爲止。

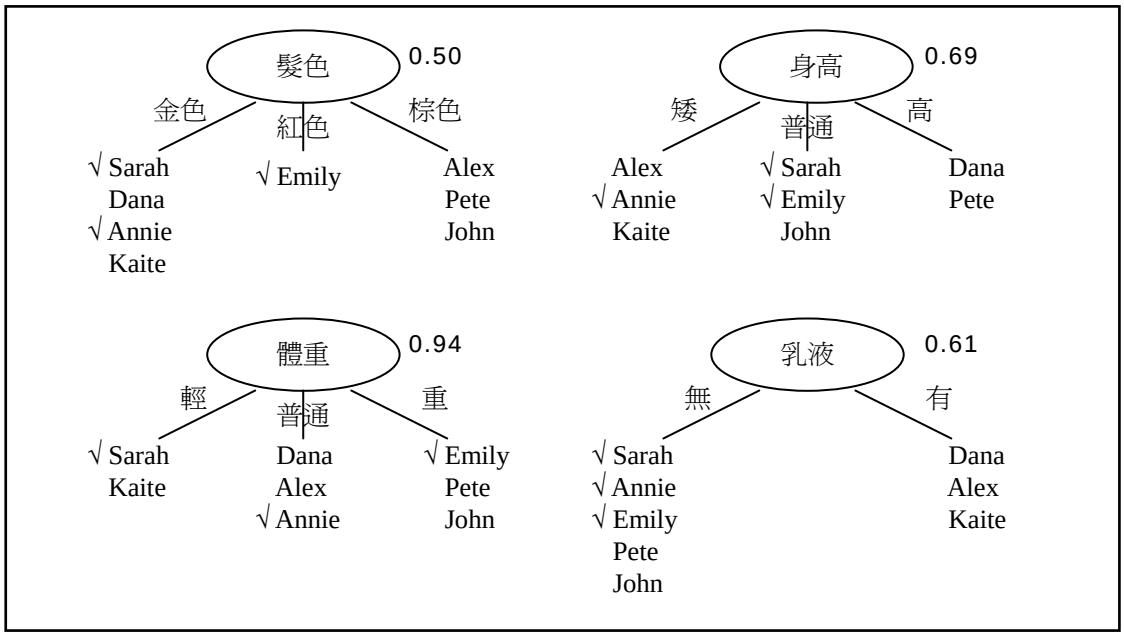


圖 15-15　全部 8 個樣本以不同特徵分類結果

以表 15-1 的 8 個樣本爲例，分別以不同特徵爲節點，其分支如圖 15-15 所示，其中打勾者（√）爲曬傷，其餘則沒有曬傷。先計算頭髮顏色這一特徵的平均亂度值，金髮有 4 人，其中有 2 人曬傷，另外 2 人沒有，紅髮只有 1 人且曬傷，棕髮有 3 人均沒有曬傷。其計算如下：

$$\sum_b \frac{n_b}{n_t} \times d_b = \frac{4}{8} \times \left(\left(-\frac{2}{4} \log_2 \frac{2}{4} \right) + \left(-\frac{2}{4} \log_2 \frac{2}{4} \right) \right) + \frac{1}{8} \times 0 + \frac{1}{8} \times 0 = 0.5$$

繼續計算其它特徵的平均亂度值，並標示於圖 15-15 不同特徵節點的右方。

因爲髮色的平均亂度值最低，所以被選爲建構識別樹的第一個特徵。由圖 15-15 中得知，頭髮爲紅色和棕色的人只有一種類別，金髮的人則有兩種類別，2 人曬傷，2 人則無。因此我們繼續針對金髮 4 人計算其它特徵的平均亂度，如圖 15-16 所示，其計算結果標示於節點右方。從以上結果可知，是否塗抹防曬乳液可得最佳結果，因此被選爲第二個測試的特徵，因其平均亂度值爲 0，代表其所有分支都只剩下一個類別，因此我們可以建構出一個最小的識別樹如圖 15-17(a) 所示。

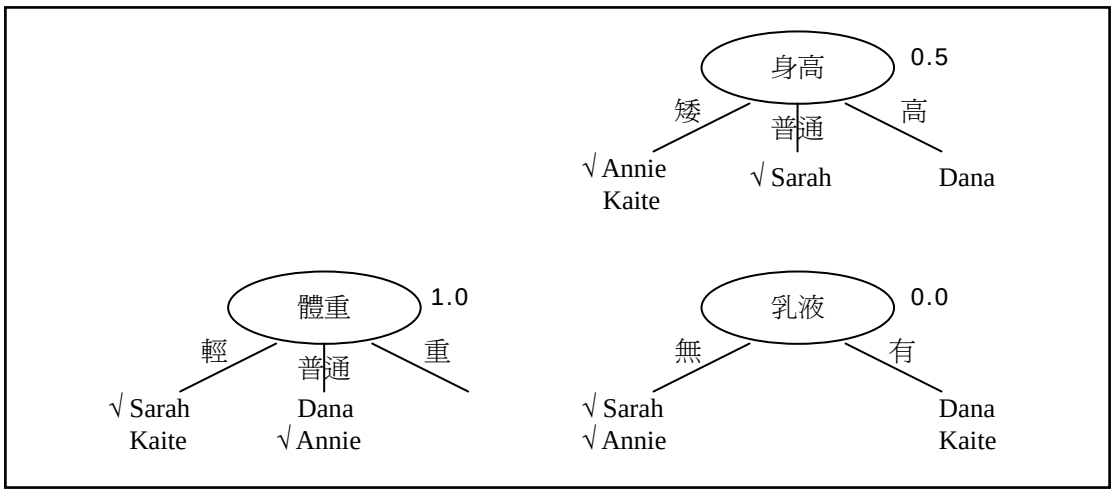


圖 15-16　金髮的 4 個樣本再以其它三種特徵分類的結果

我們可以再進一步從所建構的識別樹中得出曬傷的規則（原因）來，我們從髮色節點沿著分支往下走，直到遇見終端點爲止，因此我們可以得到四條規則，如圖 15-17(b) 所示。根據上面所得到的四條規則，我們可以從其中兩項特徵，猜測其他人有沒有可能會曬傷。

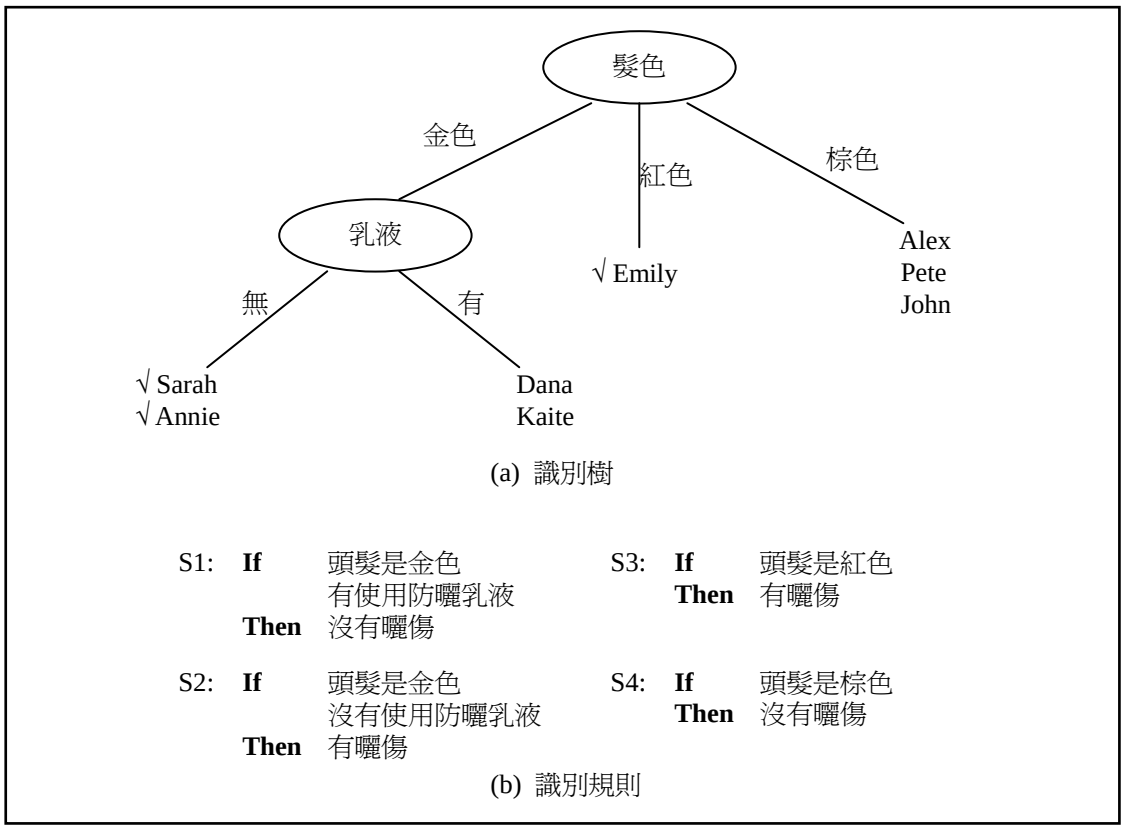


圖 15-17　曬傷檢驗識別樹與其所得規則

１５－６　類神經網路

學習是要透過我們的頭腦，因而研究大腦神經細胞的運作，可以幫助我們了解學習在腦神經是如何完成的，進而可以模擬神經細胞的運作以達到類似學習的功能。據估計人腦約有一千億（10^{11}）個神經細胞，每個神經細胞約有一千（10^{3}）根連結與其他神經細胞相連，因此人腦中約有一百萬億（10^{14}）根連結，形成一個高度連結網狀的神經網路（neural network）。科學家們相信：人腦的資訊處理工作即是透過這些連結來完成的 [葉怡成 1993]。

神經細胞的形狀與一般的細胞有很大的不同，它包括：**細胞體**（soma）：神經細胞中呈核狀的處理機構；**軸突**（axon）：神經細胞中呈軸索狀的輸送機構；**樹狀**

突（dendrites）：神經細胞中呈樹枝狀的輸出入機構；與**突觸**（synapse）：樹狀突上呈點狀的連結機構。根據神經學家的研究發現：當神經細胞透過神經突觸與樹狀突從其它神經元輸入脈波訊號後，經過細胞體處理，產生一個新的脈波訊號。如果脈波訊號夠強，將產生一個約千分之一秒 100 毫伏的脈波訊號。這個訊號再經過軸突傳送到它的神經突觸，成爲其它神經細胞的輸入脈波訊號。如果脈波訊號是經過**興奮神經突觸**（excitatory synapse），則會增加脈波訊號的速率；相反的，如果脈波訊號是經過**抑制神經突觸**（inhibitory synapse），則會減少脈波訊號的速率。因此，脈波訊號的速率是同時取決於輸入脈波訊號的速率，以及神經突觸的強度。而神經突觸的強度可視爲神經網路儲存資訊之所在，神經網路的學習即在調整神經突觸的強度。

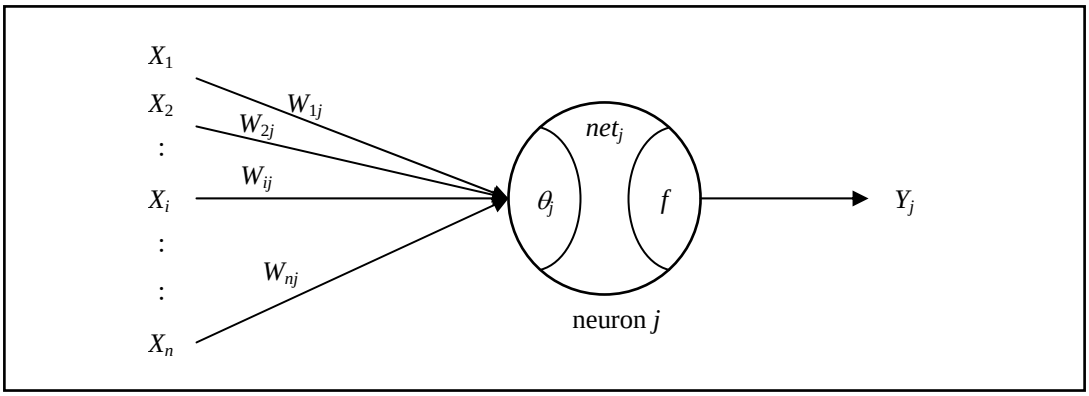


圖 15-18　人工神經細胞結構

類神經網路（artificial neural networks），或譯爲**人工神經網路**，則是指模仿生物神經網路的資訊處理系統，它是由許多人工神經細胞（又稱爲類神經元、人工神經元、與處理單元）所組成，人工神經細胞，如圖 15-18 所示。本節將探討最古老、也是最基本的類神經網路模式——**感知機**（perceptron），它是 1957 年由 Rosenblatt 所提出。感知機的基本原理是由腦神經模型所啓發，特別是 1943 年 McCulloch 和 Pitts 所共同提出的數學模型，通稱爲 MP 模型，以及 Hebb 所提出的神經元學習規則，通稱爲 Hebb 學習規則。

MP 模型的要點如下：(1) 神經元的狀態爲興奮或抑制二者之一，可用 0 表示抑制狀態，用 1 表示興奮狀態。(2) 神經元與其它神經元間的連結，可用一個**加權值**（weight）表示連結強度。(3) 神經元的狀態會經由連結輸出到其它神經元，成爲其輸入。(4) 神經元的狀態受其相連的神經元制約，當從這些神經元傳來的輸入訊號（即該神經元的狀態）經過連結以加權乘積和計算所得的值大於某**門檻值**（threshold）時，神經元的狀態將成爲興奮狀態；否則，爲抑制狀態。以公式表示爲：

$$Y_j = f(net_j) \tag{15-6.1}$$

$$net_j = \sum_i W_{ij} X_i - \theta_j \tag{15-6.2}$$

其中，W_{ij} 爲神經元 i 與神經元 j 間的連結強度，即連結加權值，X_i 爲從神經元 i 傳來的輸入訊號，θ_j 爲神經元 j 的門檻值，f 爲**轉換函數**（transfer function），通常爲一個**階梯函數**（step function），其定義如下：

$$f(S) = \begin{cases} 1 & \text{if } S > 0 \\ 0 & \text{if } S \le 0 \end{cases} \tag{15-6.3}$$

(5) 神經網路的學習過程即在調整神經元間的連結強度，即連結加權值。而 Hebb 學習規則的要點如下：調整兩個神經元間連結加權值的原則爲當第 i 個與第 j 個神經元同時處於興奮狀態時，則其連結應當加強。Hebb 學習規則與動物行爲科學中的條件反射學說一致。

感知機的網路架構有兩種，如圖 15-19 所示，一含有隱藏層，另一種則無。它們皆包括有輸入層與輸出層。輸入層用以表現網路的輸入變數，其處理單元數目依問題而定，使用線性轉換函數 $f(X) = X$，亦即輸入值即爲輸出值。隱藏層用以表現輸入處理單元間的交互影響，其處理單元數目通常以實驗方式決定其最佳數目，隱藏層可以有一層以上，也可以沒有。輸出層用以表現網路的輸出變數，

其處理單元的數目依問題而定。輸入變數形成一個輸入向量，輸出變數形成一個輸出向量。

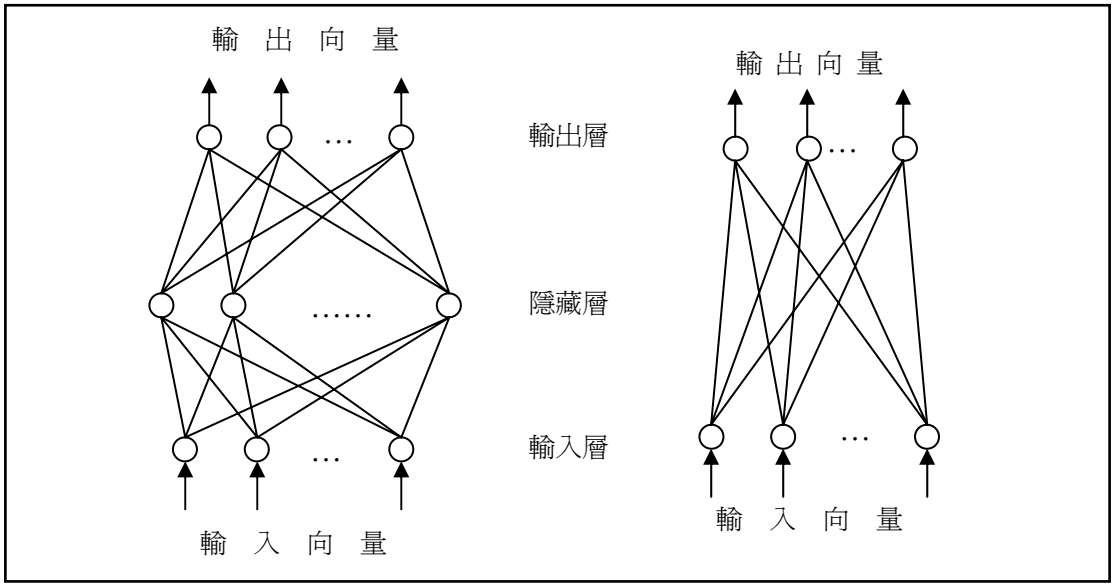


圖 15-19 感知機網路架構

我們以簡單的無隱藏層的感知機來說明類神經網路的學習機制。在神經網路的學習中，樣本資料以數值形式表示，每一個樣本都包含有輸入向量 $\boldsymbol{X} = [X_1, X_2, \ldots, X_n]$ 和目標輸出向量 $\boldsymbol{T} = [T_1, T_2, \ldots, T_m]$。一般將所有的樣本資料隨機分爲兩部分，一部分爲**訓練樣本**（training samples），另一部分爲**測試樣本**（test samples）。首先，將感知機初始化，即給定每一個連結一個隨機亂數值。然後將一個訓練樣本的輸入向量 $\boldsymbol{X}$ 輸入感知機中，並利用公式 (15-6.1) 和 (15-6.2) 計算其推論輸出向量 $\boldsymbol{Y} = [Y_1, Y_2, \ldots, Y_m]$。此網路利用由訓練樣本輸入之目標輸出向量 $\boldsymbol{T}$ 和透過網路推得的推論輸出向量 $\boldsymbol{Y}$ 相較之下的誤差，作爲修正連結中的加權值的依據，以從訓練樣本中學習隱含的輸入向量與輸出向量之對應關係。差距量 δ_j 計算公式如下：

$$\delta_j = T_j - Y_j \tag{15-6.4}$$

若 $\delta_j > 0$，表示推論輸出變數 Y_j 小於目標輸出變數 T_j，根據公式 (15-6.2) 得知

連結加權值 W_{ij} 太小，故應增加 W_{ij} 的值。相反的，若 $\delta_j < 0$，表示推論輸出變數 Y_j 大於目標輸出變數 T_j，根據公式 (15-6.2) 得知連結加權值 W_{ij} 太大，故應減少 W_{ij} 的值。加權值之改變量公式可表達如下：

$$\Delta W_{ij} = \eta \cdot \delta_j \cdot X_i \tag{15-6.5}$$

其中，η 爲學習速率（learning rate），控制每次加權值改變量的幅度。公式 (15-6.5) 中，加權值之改變量也應與輸入訊號 X_i 成正比，因爲訊號越大，其修正量也應越大。同理，輸出單元的門檻值改變量公式計算如下：

$$\Delta \theta_j = -\eta \cdot \delta_j \tag{15-6.6}$$

類神經網路的學習過程，通常以一次一個訓練樣本的方式進行，直到學習完所有的訓練樣本爲止，稱爲一個**學習循環**（learning cycle）。加權值與門檻值的修正可採用**逐步學習**（step learning）或**批次學習**（batch learning），逐步學習是每輸入一個訓練樣本，計算其加權值與門檻值的修正量後立即修改。而批次學習是在一個學習循環後，計算所有訓練樣本的加權值與門檻值的修正量後，依下列公式計算其整體修正量而修改之。

$$\Delta W_{ij} = \frac{\sum_m \Delta W_{ij}^m}{\sqrt{N}} \tag{15-6.7}$$

$$\Delta \theta_j = \frac{\sum_m \Delta \theta_j^m}{\sqrt{N}} \tag{15-6.8}$$

其中，m 表示第 m 個樣本，而 N 爲訓練樣本總數。一個網路可以將訓練樣本反覆學習多個循環，直到滿足終止條件爲止。而終止條件可訂爲執行一定數目的學習循環或是網路已收斂（即誤差不再有明顯變化）。感知機的誤差程度可用總錯誤率 E 定義如下：

$$E = \frac{N_E}{N} \tag{15-6.9}$$

其中，N_E 爲誤分類樣本總數，而誤分類樣本是指測試樣本中，其推論輸出值最大的輸出單元，與目標輸出值最大的輸出單元不是同一個單元之樣本。感知機網路演算法整理於圖 15-20，其中粗體字（如 $\boldsymbol{W}$、$\boldsymbol{\theta}$、$\boldsymbol{X}$、$\boldsymbol{T}$、與 $\boldsymbol{\delta}$ 等）表示整個矩陣或是向量。

學習過程：

1. 設定網路參數。
2. 以均佈隨機亂數設定加權值矩陣 $\boldsymbol{W}$，與偏權值向量 $\boldsymbol{\theta}$ 初始值。
3. 輸入一個訓練樣本的輸入向量 $\boldsymbol{X}$ 與目標輸出向量 $\boldsymbol{T}$。
4. 計算推論輸出向量 $\boldsymbol{Y}$。
5. 計算差距量 $\boldsymbol{\delta}$。
6. 計算加權值矩陣修正量$\Delta\boldsymbol{W}$，以及偏權值向量修正量 $\Delta\boldsymbol{\theta}$。
7. 更新加權值矩陣 $\boldsymbol{W}$，以及偏權值向量 $\boldsymbol{\theta}$。
8. 重複步驟 3 至步驟 7 直至到收斂或執行一定數目的學習循環。

回想過程：

1. 設定網路參數。
2. 讀入加權值矩陣 $\boldsymbol{W}$ 與偏權值向量 $\boldsymbol{\theta}$。
3. 輸入一個測試樣本的輸入向量 $\boldsymbol{X}$。
4. 計算推論輸出向量 $\boldsymbol{Y}$。

圖 15-20　感知機的學習過程與回想過程

以下我們以表 15-2 的 4 個訓練樣本說明感知機的學習過程。樣本的輸入向量 $\boldsymbol{X}$ 包含有兩個輸入變數 X_1 和 X_2，輸出向量 $\boldsymbol{T}$ 只包含一個輸出變數 T，因此我們可以設計一個簡單不含隱藏層的感知機網路架構如圖 15-21 所示，包含兩個輸入單元與一個輸出單元，編號標示於其旁。令初始加權值 $W_{13} = 1.0$，$W_{23} = -1.0$，輸出單元的門檻值 $\theta_3 = 0.0$。假設網路採批次學習，其學習速率 $\eta = 0.5$。在此範例中，因爲只有一個輸出，爲簡便起見，我們將其下標省略，如用 θ 代表 θ_3，Y

代表 Y_3 等。

表 15-2　訓練樣本

樣本 (m)	X_1	X_2	T
1	–1	–1	0
2	–1	1	1
3	1	–1	1
4	1	1	1

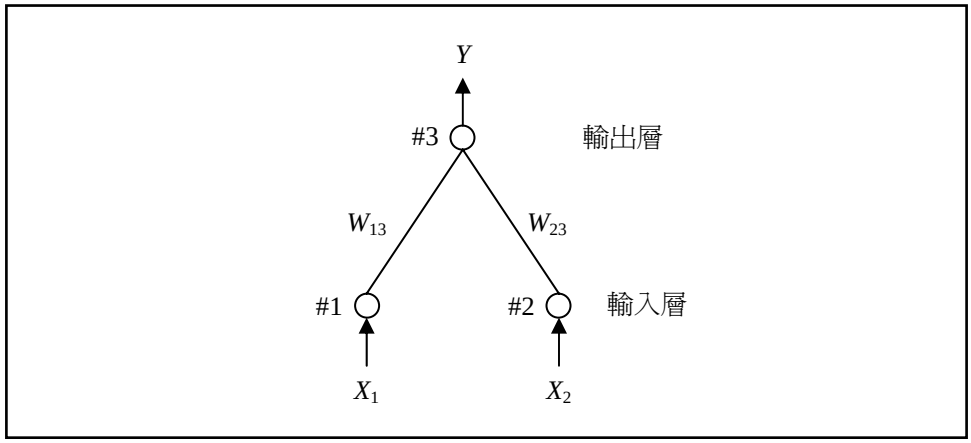


圖 15-21　感知機網路架構

第一個樣本 $X_1 = -1$，$X_2 = -1$，$T = 0$ 的學習過程如下：首先計算推論輸出值 Y，由公式 (15-6.2) 得

$$net = W_{13} \cdot X_1 + W_{23} \cdot X_2 - \theta = (1.0)(-1) + (-1.0)(-1) - 0.0 = 0.0 \text{ 。}$$

由公式 (15-6.1) 與 (15-6.3) 得知

$$Y = f(net) = f(0.0) = 0 \text{ 。}$$

接下來，計算輸出層差距量 δ，由公式 (15-6.4) 得

$$\delta = T - Y = 0 - 0 = 0 \text{ 。}$$

最後，計算加權值修正量 ΔW_{13} 和 ΔW_{23} 及偏權值修正量 $\Delta\theta$，由公式 (15-6.5) 和 (15-6.6) 得

$$\Delta W_{13}^{1} = \eta \cdot \delta \cdot X_1 = (0.5)(0)(-1) = 0 \text{ ，}$$

$$\Delta W_{23}^{1} = \eta \cdot \delta \cdot X_2 = (0.5)(0)(-1) = 0 \text{ ，}$$

$$\Delta\theta^{1} = -\eta \cdot \delta = -(0.5)(0) = 0 \text{ 。}$$

其它三個樣本的學習過程如表 15-3 所示。

表 15-3　感知機的訓練樣本學習過程

m	X_1	X_2	T	net	Y	δ	ΔW_{13}^{m}	ΔW_{23}^{m}	$\Delta\theta^{m}$
1	–1	–1	0	0.0	0	0	0.0	0.0	0.0
2	–1	1	1	–2.0	0	1	–0.5	0.5	–0.5
3	1	–1	1	2.0	1	0	0.0	0.0	0.0
4	1	1	1	0.0	0	1	0.5	0.5	–0.5

根據公式 (15-6.7) 和 (15-6.8) 計算批次學習的加權值與門檻值如下：

$$\Delta W_{13} = \frac{\sum_{m=1}^{4} \Delta W_{13}^{m}}{\sqrt{N}} = \frac{(0.0) + (-0.5) + (0.0) + (0.5)}{\sqrt{4}} = \frac{0.0}{2} = 0.0 \text{ ，}$$

$$\Delta W_{23} = \frac{\sum_{m=1}^{4} \Delta W_{23}^{m}}{\sqrt{N}} = \frac{(0.0) + (0.5) + (0.0) + (0.5)}{\sqrt{4}} = \frac{1.0}{2} = 0.5 \text{ ，}$$

$$\Delta\theta = \frac{\sum_{m=1}^{4} \Delta\theta^{m}}{\sqrt{N}} = \frac{(0.0) + (-0.5) + (0.0) + (-0.5)}{\sqrt{4}} = \frac{-1.0}{2} = -0.5 \text{ 。}$$

於是，我們更新網路的連結加權值與門檻值如下：

$$W_{13} = W_{13} + \Delta W_{13} = (1.0) + (0.0) = 1.0$$

$$W_{23} = W_{23} + \Delta W_{23} = (-1.0) + (0.5) = -0.5$$

$$\theta = \theta + \Delta\theta = (0.0) + (-0.5) = -0.5$$

繼續以上步驟，直到網路收斂爲止。

我們可以稍微研究一下，在上個範例中，兩個輸入變數 X_1 和 X_2 可以表示成二維平面中的水平與垂直軸，於是四個樣本可以表示成落在平面上四個點，其中第一個樣本其輸出爲 0，我們用黑點表示，其它三個樣本輸出爲 1，我們以白點表示，因此樣本有黑白兩個類別。在計算網路輸出 Y 時，$net = W_{13}X_1 + W_{23}X_2 - \theta = 0$ 剛好是一條直線，代表一條邊界線將平面分割爲兩部分。因此網路的學習即在自動找到一條邊界線能將黑白兩類樣本點分開。此條邊界線初始設定爲 $X_1 - X_2 = 0$，爲一對角線，並沒有將黑點與白點分開，如圖 15-22 所示。經過第一個批次學習後，此直線往逆時針方向修正成 $2X_1 - X_2 = -1$。此網路在經過幾個批次學習後會收斂，此時邊界線會將黑點與白點分開。

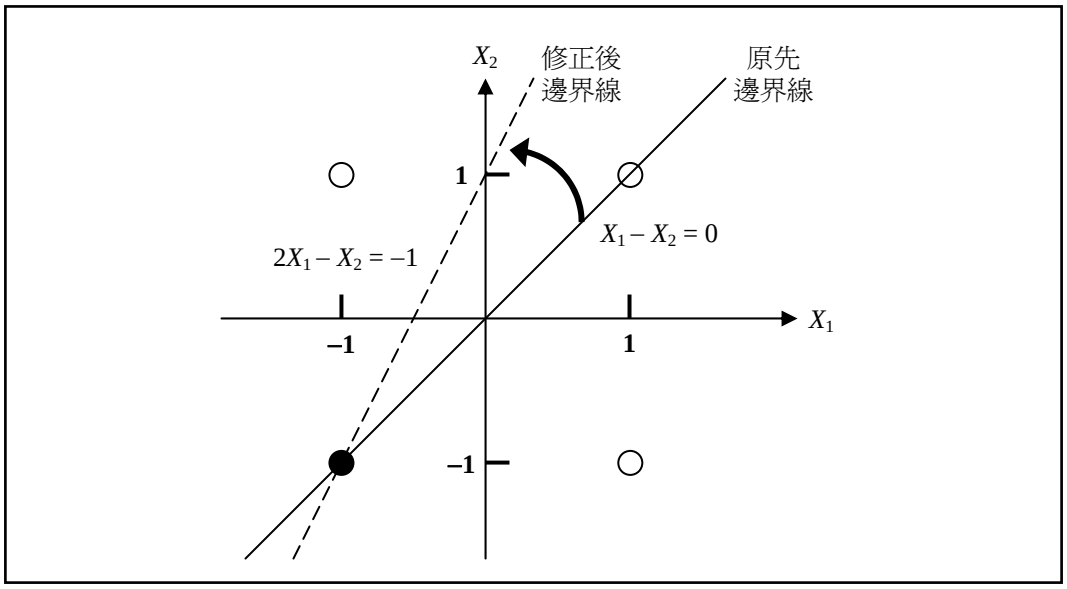


圖 15-22　經過第一次批次學習後的邊界線修正的結果

１５－７　基因演算法

基因演算法（genetic algorithms），或稱爲**遺傳演算法**，是由 John Holland 於

1970 年代所提出的，他利用電腦來模擬達爾文（Charles Darwin）的**天演論**「物競天擇，適者生存」的概念，用以解決**最佳化**（optimization）問題，以求得較佳解。在生物的**染色體**（chromosomes）中含有爲數眾多的**基因**（genes），每個基因具有某些特性。生物在繁殖下一代時，由父母的染色體，經過**交換**（crossover）、**突變**（mutation）、與**複製**（reproduction）而產生新的染色體，因此具有父母各一部份的基因，所以遺傳有父母的部分特質。

在基因演算法中，Holland 簡單以 0 與 1 表示基因，而染色體用以表示問題可能的答案。因此，將問題的答案表達成 0 與 1 的字串，稱爲**編碼**（encoding）。如圖 15-23(a) 所示爲兩個 4 位元字串的染色體。在基因演算法中，我們簡單以一個染色體代表一個**個體**（individual），而一個**群體**（population）是由 *N* 個個體所構成，也形成一個**世代**（generation）。在目前這個世代的群體中，有些個體比較優秀，依據優生學理論，優秀的個體繁衍的下一代變得更優秀的機率比較高。因而在一代一代的演進下，染色體會越來越優秀。在基因演算法中，我們訂定一個**適應函數**（fitness function），以評估每一個個體的適應情形。適應程度越高的個體代表越優秀，被選擇來繁衍下一代的機率也比較高。

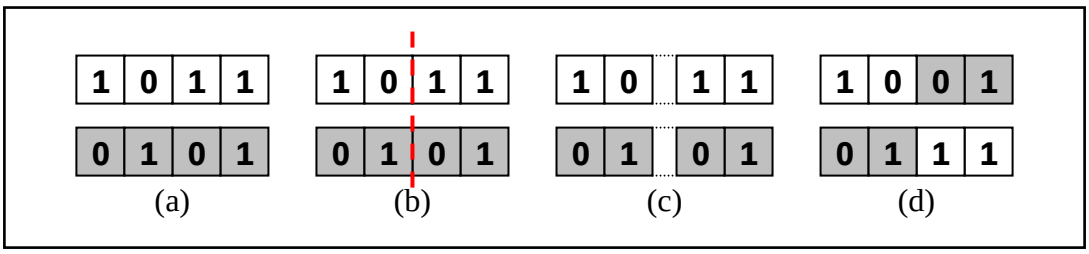


圖 15-23　一對染色體的交換運算

在進行交換的運算時，必須選擇一對染色體，決定**交換點**（crossover point），將兩個染色體在交換點的位置切成兩段，然後將後段互相對調，即構成一對新的染色體。以上例的一對染色體說明，假設交換點位置爲 2，意即在第二個基因與

第三個基因之間爲交換點所在位置，如圖 15-23(b) 所示。於是先切成四段如圖 15-23(c) 所示。然後，後段部分互相交換，可得到一對新的染色體，其結果如圖 15-23(d) 所示。至於突變運算，即將基因值改變，意即若原先基因爲 0，突變後爲 1，反之亦然。完整的基因演算法列於圖 15-24。

1. 首先將問題的可能答案編碼成固定長度的染色體，然後選定群體中染色體的數目 N，交換機率 p_c，突變機率 p_m。
2. 定義適應函數 f。
3. 隨機產生初始群體中的 N 個染色體：$x_1, x_2, \ldots, x_N$。
4. 計算每一個染色體的適應函數值：$f(x_1), f(x_2), \ldots, f(x_N)$。
5. 從目前的群體中選出一對染色體，以爲配對之用。染色體的選擇與其適應函數值有關。適應程度越高的染色體被選擇的機率就越高。
6. 利用基因運算子——交換、突變、和複製，以產生一對新的染色體。
7. 將新產生的染色體放入新的群體中。
8. 重複步驟 5，直到新產生的染色體數目達到 N 爲止。
9. 將新的群體取代舊的群體。
10. 跳到步驟 4，重複整個行程，直到終止條件滿足爲止。

圖 15-24　基因演算法

讓我們舉例說明基因演算法是如何運作的。假設尋找函數 $(15x - x^2)$ 的最大值（當 $x = 7.5$ 時，函數值爲 56.25 是最大值），其中參數 x 介於 0 到 15 之間。爲簡化起見，令 x 爲整數，以二進位表示需要四個位元，因此染色體可以由四個基因所組成，如表 15-4 所示。

表 15-4　染色體的基因表示法

整數	二進位碼	整數	二進位碼	整數	二進位碼
1	**0 0 0 1**	6	**0 1 1 0**	11	**1 0 1 1**
2	**0 0 1 0**	7	**0 1 1 1**	12	**1 1 0 0**
3	**0 0 1 1**	8	**1 0 0 0**	13	**1 1 0 1**
4	**0 1 0 0**	9	**1 0 0 1**	14	**1 1 1 0**
5	**0 1 0 1**	10	**1 0 1 0**	15	**1 1 1 1**

假設群體的染色體數 N 爲 6，交換機率 p_c 爲 0.7，突變機率 p_m 爲 0.001。在本例中，適應函數定義爲 $f(x) = (15x - x^2)$。基因演算法於是開始隨機產生初始群體的 6 個染色體，並計算其適應函數值，如表 15-5 所示。

表 15-5　初始隨機所產生的群體

染色體標籤	染色體字串	整數值	適應值	適應比例
*X*1	**1 1 0 0**	12	36	16.5%
*X*2	**0 1 0 0**	4	44	20.2%
*X*3	**0 0 0 1**	1	14	6.4%
*X*4	**1 1 1 0**	14	14	6.4%
*X*5	**0 1 1 1**	7	56	25.7%
*X*6	**1 0 0 1**	9	54	24.8%

染色體的選擇一般採用**輪盤方式選擇**（roulette wheel selection），如圖 15-25 所示。每一染色體在輪盤所佔的面積與其適應函數值成正比，亦即其適應比例（= 適應值 / 適應值總和）。因此，有較高的適應值，其在輪盤中所佔面積會比較大，相對的，被選擇的機會也會比較大。在圖 15-25 中，染色體 *X*1 的適應比例爲 16.5%，其所佔輪盤面積從 0%到 16.5%，染色體 *X*2 的適應比例爲 20.2%，其所佔輪盤面積從 16.5%到 36.7% (= 16.5% + 20.2%)，其餘類推。我們使用隨機亂數，其值落於某個染色體的範圍內，就選取該染色體。

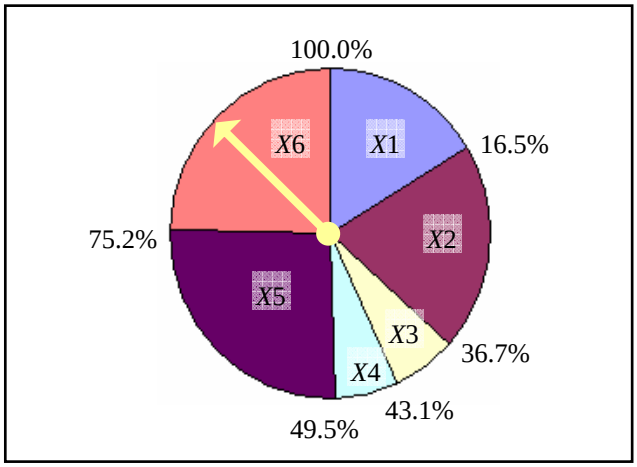


圖 15-25　輪盤方式選擇

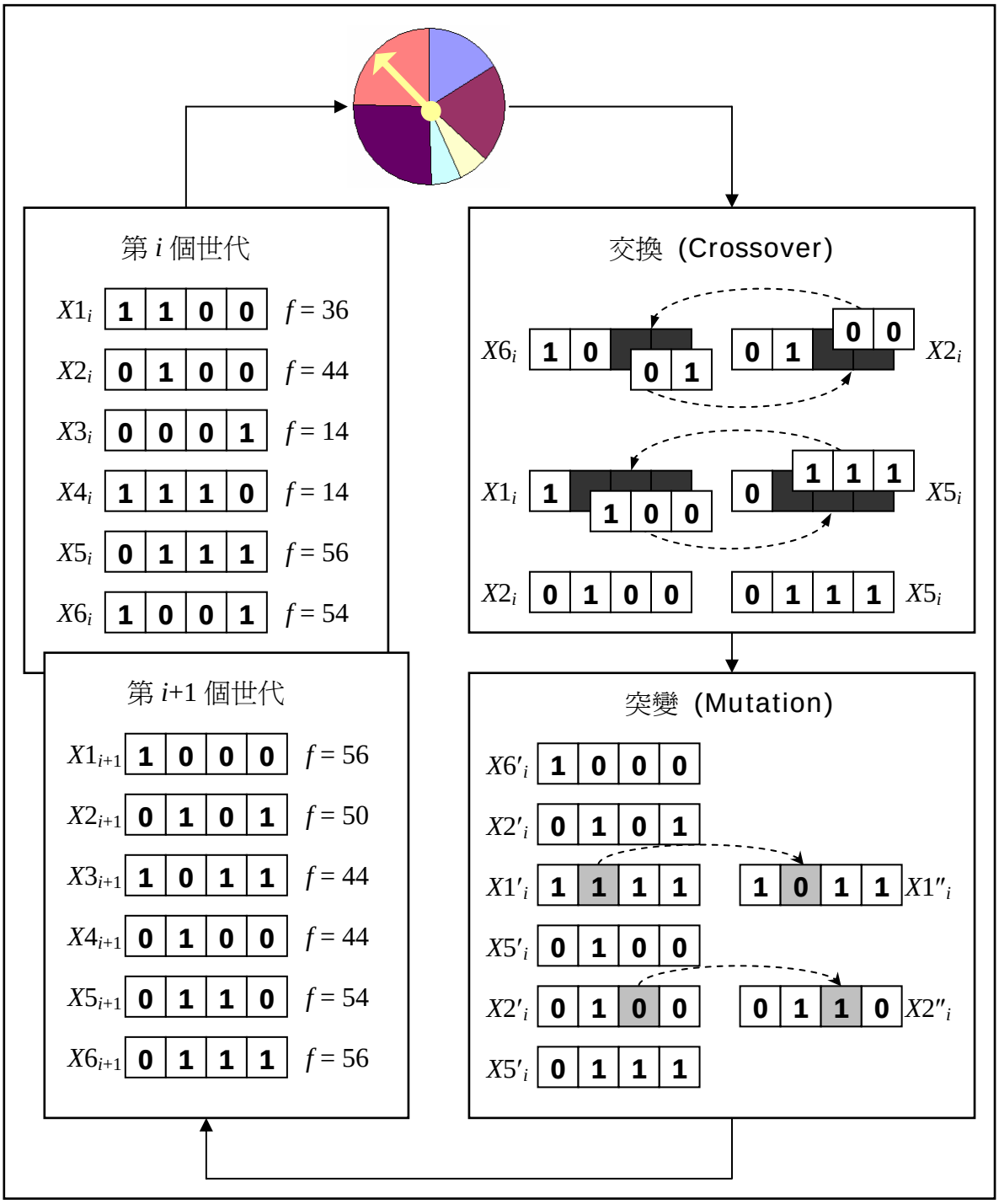


圖 15-26　基因演算法的週期

假設染色體 $X6$ 與 $X2$ 首先被選取，交換點隨機定爲 2，並且產生一隨機亂數值大於交換機率 p_c，所以此兩個染色體會進行交換，產生兩個新的染色體 $X6'$ 與 $X2'$，如圖 15-26 所示。另一對染色體 $X1$ 與 $X5$ 於交換點爲 1 的地方也進行交換，最後選取的染色體 $X2$ 與 $X5$ 因所產生的隨機亂數值小於交換機率，故而沒有交

換，直接複製成新的染色體 $X2'$ 與 $X5'$。在突變運算中，針對每個新染色體中的每個基因，隨機產生一數值，若其值小於突變機率 p_m，則將該基因改變；若原爲 1，則改爲 0，若原爲 0，則改爲 1。在此例中，有 $X1'$ 的第二個基因與 $X2'$ 的第三個基因產生突變，因而產生兩個突變的新染色體 $X1''$ 與 $X2''$。至此所產生的 6 個新染色體成爲下一個世代的群體，形成一個基因演算法的週期。如是繼續下一個週期，直到滿足終止條件爲止。而終止條件一般設定爲經過 M 個世代的演進，或是最佳適應值已收斂（上一世代與下一世代的最佳適應值的差距小於某個預定值）。從上例中可以看到，新一代的染色體，其適應值普遍提高，其平均值爲 50.67，比前一代的平均適應值 36.33 高出 14.34。

１５－８　結語

在本章中，我們介紹了人工智慧領域中，一些傳統研究的基本原理，包含：困難問題的解決、電腦遊戲、專家系統、自然語言處理、機械學習、類神經網路、與基因演算法等。介紹完其基本原理後，並輔以有趣的範例，詳細說明其運作的過程，使讀者能更深入了解人工智慧是如何利用基本原理，並透過電腦演算法來解決複雜與困難的問題。本章所介紹的部分，在整個人工智慧領域中，只是冰山的一角。人工智慧的研究乃在利用電腦來模擬人的智慧，故而有關人類智慧的探討、人類智慧的運作、與設計智慧型系統的研究，均包含在人工智慧的領域中。因此與人工智慧相關的研究領域甚廣，除資訊科學外，尚包含：心理學、語言學、神經學、邏輯學、哲學、認知科學、生物學……等等。在現代的電腦中，處處可以看到人工智慧的應用，讓電腦更有智慧，例如智慧型大樓內的種種設備。希望透過本章的介紹，能引發讀者對人工智慧的興趣，進一步將人工智慧的研究應用於利益人類社會。

１５－９ 參考文獻

[DeepBlue2005] Deep Blue URL http://www.research.ibm.com/deepblue/, 存取日期 2005/03/12.

[Giarratano2004] Joseph Giarratano and Gary Riley, *Expert Systems: Principles and Programming*, Fourth Edition, 2004.

[Hopcroft1979] John E. Hopcroft and Jeffrey D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.

[Negnevitsky2002] Michael Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems*, Addison Wesley, 2002.

[Nilsson1998] Nils J. Nilsson, *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann, 1998.

[Rich1991] E. Rich and K. Knight, *Artificial Intelligence*, Second Edition, McGraw-Hill, 1991.

[Weizenbaum1966] Joseph Weizenbaum, “ELIZA—A Computer Program For the Study of Natural Language Communication Between Man and Machine,” *Communications of the ACM*, Volume 9, Number 1 (January 1966): 36-45. http://i5.nyu.edu/~mm64/x52.9265/january1966.html

[Winston 1992] Patrick H. Winston, *Artificial Intelligence*, Third Edition, Addison-Wesley, 1992.

[孫家麟 1985] 孫家麟，*微電腦的人工智慧實驗*，松崗電腦圖書資料有限公司，1985 年。

[葉怡成 1993] 葉怡成，*類神經網路模式應用與實作*，儒林圖書有限公司，1993 年 8 月二版。

15－10 習作

1. 有三個傳教士與三個野蠻人要渡河，岸邊剛好有一艘能乘載兩人的竹筏。任何時候，在河岸兩邊如果傳教士的人數比野蠻人少，就會被野蠻人吃掉。請問：要如何才能將此六人安全地渡河到對岸去？
2. 假設在井字遊戲中，目前的盤局如下。*MAX* 下一步要如何走才可以得到最大的勝算？請畫出如圖 15-4 的樹狀結構圖來分析，以得出最好的答案。

3. 假設機器人羅碧看到一隻動物，猜想牠是企鵝。若羅碧利用後向鍊結法推論，那它必須要辨識到哪些特徵才能確定是企鵝？
4. 請畫出英文句子 “I want the fast printer” 的剖析樹。
5. 在機械學習的範例中，假設 Dana 的體重改爲重，且會曬傷。請建構出一識別樹，並得出其識別規則。
6. 在感知機的範例中，經過幾個學習循環會收斂？請畫出每一次學習後修正的邊界線，了解邊界線修正的過程。
7. 在基因演算法的範例中，假設選取、交換、與突變的方式都與圖 15-26 產生第 $i + 1$ 個世代一樣，求得第 $i + 2$ 個世代的 6 個新染色體。

15－11 索引

二劃	人工智慧	artificial intelligence
四劃	井字遊戲	Tic-Tac-Toe
	文法	grammar
五劃	由上而下剖析	top-down parsing
	由下而上剖析	bottom-up parsing
	加權值	weight
	世代	generation

六劃	自然語言	natural languages
	交換	crossover
	交換點	crossover point
七劃	抑制神經突觸	inhibitory synapse
八劃	知識庫	knowledge base
	知識表達法	knowledge representation
	狀態轉變圖	state transition diagram
	事實	facts
	門檻值	threshold
九劃	前提	antecedents
	前向鍊結	forward chaining
	後向鍊結	backward chaining
	突觸	synapse
	突變	mutation
	染色體	chromosomes
十劃	特徵	features
	剖析樹	parse tree
	訓練樣本	training samples
	個體	individual
十一劃	規則	rules
	專家系統	expert systems
	專家知識	expertise
	推論機制	inference engine
	符號式學習	symbolic learning
	連接式學習	connectionist learning
	細胞體	soma
	基因	genes
	基因演算法	genetic algorithms
十二劃	極大極小法	minimaxing method
	評估函數	evaluation function
	結論	consequents
	軸突	axon
	測試樣本	test samples
	最佳化	optimization
	階梯函數	step function
十三劃	電腦遊戲	computer game
	電腦程式語言	computer programming languages
	感知機	perceptron

	群體	population
	複製	reproduction
十四劃	語彙分析	lexical analysis
	語法分析	syntactical analysis
	語意分析	semantic analysis
十五劃	熵	entropy
	輪盤方式選擇	roulette wheel selection
	適應函數	fitness function
	編碼	encoding
十六劃	機械學習	machine learning
	樹狀突	dendrites
	興奮神經突觸	excitatory synapse
	遺傳演算法	genetic algorithm
	學習循環	learning cycle
十八劃	轉換函數	transfer function
十九劃	識別樹	identification tree
	類神經網路	artificial neural networks

１５－１２　致謝

衷心感謝陳俊文、彭建文、和陳志遠等三位博士生的細心校稿，讓錯誤降到最低；尤其感謝陳俊文所提出的諸多建議，使得本章內容更爲充實、可讀性更爲提升，在此致上十二萬分的謝意。最後要感謝系主任王英宏教授給我這個機會，將自己心中長久以來的心願:『以簡明易懂的文字介紹人工智慧的基本原理給初學者』得以如願以償，並感謝主任在這段期間耐心等待本章的完稿。

洪文斌 敬上

中華民國九十四年四月十三日

於淡江大學 資訊工程學系