

二.文獻探討

與本文理論方法相關之技術包括資料探勘、專家系統、案例式推論、類神經網路及全域最佳化等，以下將一一討論這些與本文相關的技術理論範疇。

在討論這些關鍵技術前，需先討論「知識」以貫穿各技術，如同五臟必以血脈相連；知識起源於智者的思維，是一種流動性質的綜合體，其中包括結構化的經驗、價值以及經過文字化的資訊。根據 Harris 描述，資料(Data)為最低層次，資料本身沒有意義，它必須被分類、聚集、分析和解釋，當資料經由處理轉變為資訊(Information)，且資訊有主旨與目的時，才具有其意義；當資訊與事件的過程經驗作結合，它才變成知識(Harris 1996)。Dretske 則認為資訊能產生知識，即代表能從中學習，知識可以定義為資訊所產生的觀念，且個人所接受的資訊會對其知識來源的可靠性認知有關(Dretske 1981)。Nightingale 則認為資訊是表現知識的基礎，資訊有必要說明其所代表事件的可能意義。在組織中，知識不僅存在文件與儲存系統中，也蘊涵在日常例行工作、過程、執行與規範當中(Nightingale 1998)。

表 1 知識探討列表

學者	重要論述
Scharmer(2001)	知識分為內隱知識、外顯知識以及自我超越的知識三種，外顯知識是容易傳授以及分享的知識；內隱知識是可以透過具體化傳遞與溝通，是行動的知識；在自我超越的知識方面試一種可能存在的感覺知識，但是無法具體化，無法看見的知識。
Dixon(2001)	知識是資料→資訊→行動→知識構成，格式化的資料處理，透過工具傳達資料成為資訊，將資訊和行動結合成為知識。
Davenport(1999) & Prusak(1999)	知識是由資料→資訊→知識而來，數據經過記錄後成資料，資料能啟發接收者成為資訊；資訊的流動綜合體即為知識。

一、專家系統

(一)專家系統之定義

專家是指在特定領域中具有專業知識的人，擁有不為多數人所知或所利用的專業技能，故能解決大多數人不能或無法有效地解決之問題(Giarratano & Riley 1998)；專家系統(Expert Systems)則是一種容納知識的電腦程式，目的在處理解決專業知識問題，亦可提出建議讓使用者作決定(Feigenbaum 1982)；專家系統的目的在找出解決問題的方案，而這個問題解決的方案，可以在眾多已存在的方案中分析選擇出來，也可以使用合成建構出來的方案。

(二)專家系統之內容

專家系統有相當多的好處，例如當使用者要做某一項決定時，它可用利用處理器快速的效能，提供以邏輯思考方法下取得的消息，作為使用者決策時的參考；此外，人類專家精神體能的持久性會隨時光的流逝而衰退，並且電腦系統也容易轉化或再生，而產生的結果也較為調和一致；專家系統可形成一種團體記憶(Institutional Memory)，不會因為人亡而政息，它可以協助新手克服青黃不接的困難；如此，專家系統即為專家的化身，它可以讓專業領域更完整，而非完全的取代專家。

專家系統既為專家的化身，而需要專家的工作意味著複雜性及不確定性；因此，專家系統應該具備邏輯推理及解決現實複雜問題的能力；此外，專家系統所提出的方案還須具有專家的水準及效能，它要能處理特定專業領域的問題，並還要有合理的推論及解釋提以供使用者參考。專家系統處理運作的基本概念如圖 4 所示：

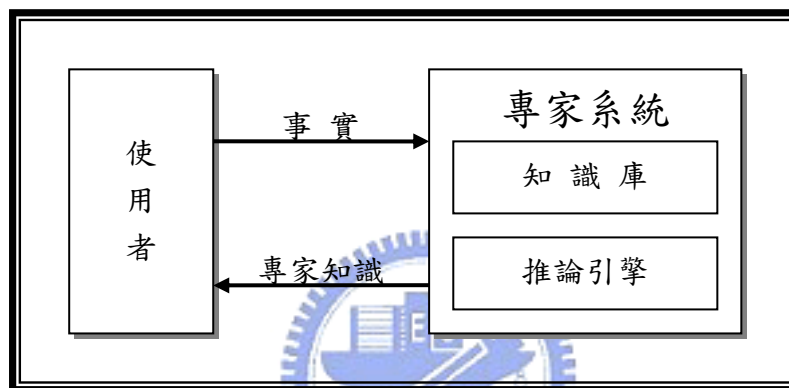


圖 4 專家系統運作的基本模式

專家系統具有符號表示、推論、搜尋及解釋等特性，甚至有從錯誤中學習的潛能，而最有用的特徵就是其具有領域專家的意見和技術。因此，專家系統須具備知識表示能力與推論的方法，並且以知識應用為重點，用有效的方法解決問題，而避免盲目搜尋；為了有效的解決問題，應設定問題領域的範圍與解決技術，並依照問題而採取特定的解決方法；還有，系統要能夠產生可提供解釋說明的結果給使用者參考；此外，專家系統不偏重數字或數學的運算，沒有固定的解題程序，且沒有過多的一般常識與主觀感覺；專家系統具有許多知識與經驗法則的應用(例如目標選定、情況選擇、問題發覺與對策提出等方面)，並能夠清楚地說明，採用的方法及考慮的條件。

因此，專家系統很多方面與傳統程式不一樣，這包括了資訊組成與表示方法，傳統的系統不易擴充和改良，更無必要去解說其他操作；而專家系統可以解決需要經驗判斷的問題，或是資料不全甚至模糊不清的複雜問題。此外專家系統還須有知識庫、推導器與解釋器；而專家系統的詳細構成內容有硬體、作業系統、程式語言、知識庫、推論程序、工作記憶體、人機介面，以及知識獲取和知識解釋等子系統。使用者透過人機介面就可以和專家系統進行溝通和交談，交談的方式可以為選擇式、填空式或自然語言；利用知識庫內的規則和資料庫的數據，再加上推論程式本身的歸納演譯法則，系統就可以依據其推論的結果，回答使用者所提出的問題，並

說明其推論過程及知識解釋。

專家系統亦可稱之為知識工程，雖然有其適用性也有其限制；在專家與系統之間建構者稱為知識工程師；知識工程師從人類專家對問題解決的程序、策略和規則中抽出相關知識，並注入所建立之專家系統。程式解決難題的能力來自於擁有的知識，要使程式具有智慧，必須提供特定問題範圍內大量高水準和特殊的知識；知識的累積和編纂是專家系統中重要的做法，而專家知識的來源有事實、指導原則及觀察解決問題的方法等。專家系統是以符號集合來代替問題的觀念，在 AI 的術語中，一個符號是一串文字，代表一些真實世界的觀念(例如：產品)；符號可陳述彼此間的關係，當這些關係是在人工智慧程式表示時，被稱為符號結構(例如：有缺點的產品)；因此，符號使用的選擇、組成和編譯非常重要。

發展專家系統時，必須謹慎的考量問題的性質、複雜程度和解決問題的範疇是否適合發展專家系統；另外，專家系統發展必須有足夠的資源，招來具有專業實力的知識，且須共同贊成領域內問題的選擇，以及問題解答的精確性，並使專家能夠清晰地表達它們用於解題的方法。專家系統須有發展的正當理由，例如工作解決花費甚鉅，或是此領域專家甚為罕見還有可能失傳；此外人類專家會隨人事變動而工作環境卻無法遽然接受此情況，還有在惡劣的工作環境中需要人類專家等情況。專家系統工具有輔助系統建立、設定框架、推論邏輯、程序設計與規則基底語言等形式，其中以規則基底語言佔最多數；其功能如診斷偵測、數據分析編譯、分析評估、設計、策劃、經驗學習、監督、知識蒐集、管理顧問、訊號編輯方面等(Waterman 1985)。

(三)專家系統之應用

專家系統發展應用以醫藥為最多，其次為軍事，第三為電子；專家系統的應用例如在生物工程的 DNA 結構分析合成，化學工業的解釋決定分子結構及原子成分及有機物合成，電腦系統的診斷失誤，計算用的協助軟體建構除錯，協助教育教學，工程用的協助設計及確認最佳解及核能反應或積體電路製作問題。在通用工具方面有商用知識表現、知識獲取系統、推論系統、知識表示與解決問題控制、知識提示等；在法律應用的有產物在物索賠調解人的決策過程及稅法法則，管理學的應用有整合資料能力、風險評估、成本估計等；在製造業方面有模仿監督擬定進度、工作安排等；醫療應用如因果性網路，醫學應用如診斷、監視並指出療法；軍事用途有計劃航空、追蹤聲納、空中搜索、陣地通訊分析等；資源探勘方面如油井探測、解決水資源、評估礦儲等。

而專家系統在決策支援方面的應用包括：金融市場預估、銷售分析、市場分析、管理分析、訓練、計劃及製造等。由於電腦能快速依人類意志執行，所以有專家系統的輔助，將會使專家的眼光放的更遠(Ljubica 2002)。

二、案例式推論

案例式推論 (Case-Based Reasoning; CBR) 是由 Schank 與 Abelson 所提出，為人工智慧領域中一個重要的範疇，乃是一個方法論，用於建構電腦系統模擬人類思考推論之方法論(Schank & Abelson 1977)。在 CBR 系統方法之下，最主要的精神有二：

1. 儲存既有經驗於電腦記憶體中。
2. 將這些經驗用於解決某些新遭遇的問題：
 - (1) 自記憶體中重新取得近似的經驗或者相近狀態。
 - (2) 重複使用這些經驗中的解決方式於新遭遇的問題中。
 - (3) 並再次將新案例儲存於記憶體中。

這些經驗的儲存是一組包裹好的知識或者經過編輯與整理的問題與解答，以這些包裹好的特徵 (Feature) 集合表現出案例整體的特性與蘊含的經驗知識，並依此為依據搜尋相似之經驗。

(一) 案例式推論之理論

CBR 的基本精神就是愈是相似的問題，其解決方法愈類似，這是案例式推理所堅信的，也是自然界中的常理；其最大的特色是具有學習的功能，也是屬於機器學習的一種，因為案例式推理能靠著解決問題而不斷地持續學習，當問題成功地被解決後，這個經驗將被儲存並應用在未來發生的相似問題上，若系統遇到失敗的例子，這些失敗的教訓也將被描述且記憶，以避免將來再犯同樣的錯誤。

Watson 曾指出案例式學習並非一項特別的新技術，它是一種解決問題的方法，CBR 可以整合運用好幾種不同的技術，如類神經網路、模糊理論、遺傳演算法、歸納法等，不同技術會有不同程度的效益，針對不同需要的問題，使用合適技術才有最大的效益(Watson 1999)。Aamodt 與 Plaza 認為 CBR 是由以下四種循環過程所組成，利用過去的經驗來解決類似的問題，並把試行的結果做為經驗的累積，增加解決問題的能力，且這種類比式推理是人的天性，而其中循環流程分述如下 (Aamodt & Plaza 1994)：

1. 檢索(Retrieve)：從案例庫中搜尋相似的案例，提供新案例建議解決方案。
2. 復用(Reuse)：依相同或相似案例的解決方法去解決問題。
3. 修正(Revise)：藉由修改法則來矯正處理方式，使結果更符合當下的問題。
4. 回存(Retain)：資料可不斷累積，儲存在案例庫中，以供新案例使用。

一個新問題被案例庫推理解決，是靠著檢索一個或多個過去的相似案例，再利用這些案例來提出解決問題的方案，然後依據實行結果修正解決方案，再利用這些案例來提出解決問題的方案，然後根據實行結果修正解決方案，萃取出有用的新經驗存入案例庫 (case base) 中；案例式推論的循環模式如圖 5 所示：



圖 5 案例式推論的循環模式

(二) 案例式推論的流程

由Montazemi與Gupta所提出的案例式推論之推論流程(參考圖6),是個相當完整的案例式推理系統,許多學者提出各式各樣的案例式推理系統,但大都相去不遠;在其推論流程中首先輸入新問題的描述,接著自案例庫中擷取過去相似的案例,分析擷取的案例是否需要改編,若需要則改編之,以適用於新問題,然後評估案例的可行性與效果,若評估結果是好的則將新案例存入案例庫(Montazemi & Gupta 1996)。

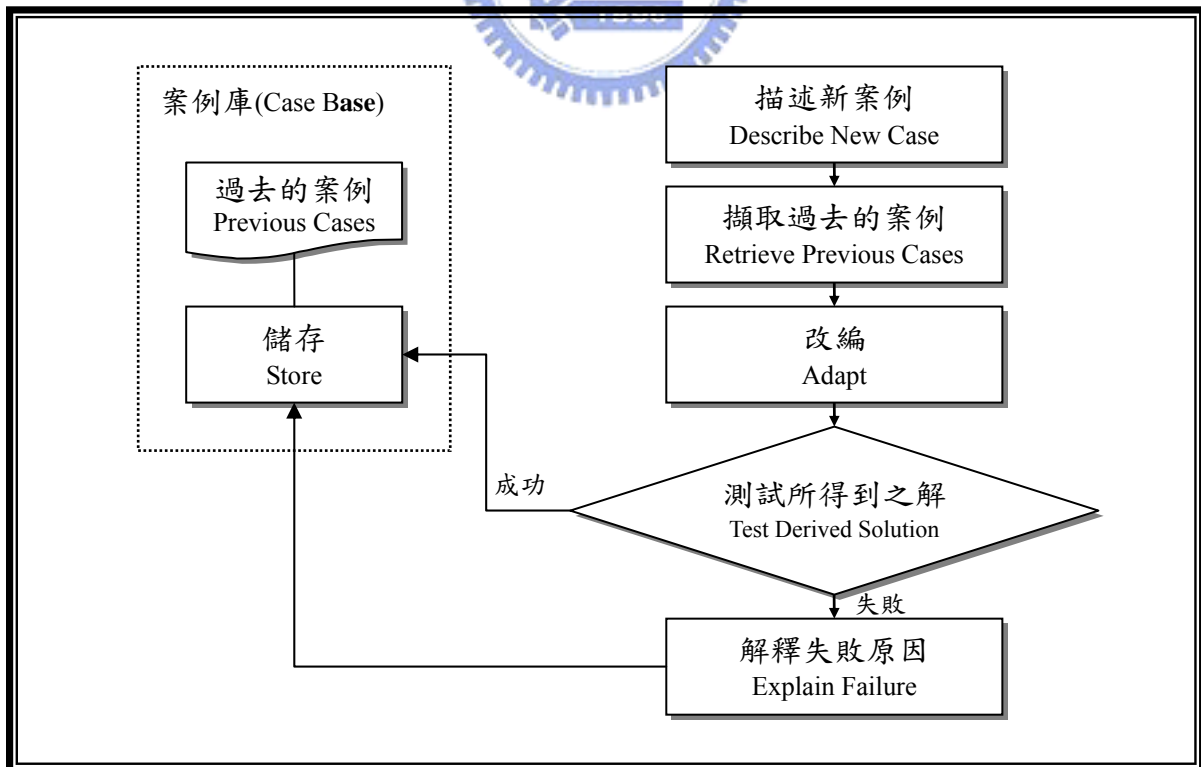


圖6 CBR系統推論流程

以下針對CBR系統推論流程之主要程序加以概略說明：

1. 案例擷取(Case Retrieval)：

此步驟通常包含兩部分：擷取過去相似的案例與選擇最佳的案例；擷取過去相似案例的目的在於取得好的案例，亦即取得對新案例能夠做出相關預測的案例，而擷取的過程是利用新案例的特徵為案例庫的案例索引(index)。在選擇最佳案例方面，此目的在於自眾多相似案例中取得最貼近或最具代表性而且值得列入考量的候選案例；在此案例擷取步驟中，案例間相似性如何判別則是關鍵所在。

2. 案例改編(Case Adaptation)：

前一步驟所選擇出的案例，可能對於新問題並不是完全適用，就如同人類過去的經驗與現在所逢的遭遇並不完全相同一樣，可能必須納入其他因素或是需要加以修正；此步驟包含兩部分：分析哪些項目必須改編以及執行改編；可以訂定一些改編策略，或是使用一些啟發式解法予以改編。

3. 案例評估(Evaluation)：

此部分主要在於測試此推論的結果輸出是否正確，包含實際運用之前以及實際應用之後的模擬評估。許多系統是由使用者自行評估，因為案例式推理系統扮演的是輔助的角色並不能真正擬出正確的解決之道，最後的決定仍在於使用者本身，然而在本論文將配合事件專家系統，使CBR系統部分能自行評估案例之正確程度；運用之前評估的方式，可以使用比較與對比的方式，將所得到的解決之道與其他相似的方案比較分析；而實際應用後的結果可以幫助系統的學習功能，透過錯誤學習，形成新的經驗，或是正確學習，將新案例存入案例庫。

4. 案例庫(Case Base)：

案例式推論因為具有案例庫而得以運作與學習。案例庫裡存放過去的案例與解決方案，如同資料庫一般分門別類，運用案例索引搜尋與儲存，可隨時存取案例，案例庫愈大，包含的案例就愈多，則所得到的結果也會愈佳。而當案例庫愈大，則可使用樹狀架構、物件導向或是階層式管理等方式，來強化案例庫的記憶體管理，使其在存取案例時，不會因案例庫龐大而降低效率。

(三) 案例式推論的應用

案例式推論的應用在初期發展時目標在改善專家系統，故而被應用於診斷類的系統，利用舊有的案例來判斷新案例的解決方式，運用的領域包括醫療、機器故障維修、保養及顧客售後服務等；而近來則多被應用在研發設計等相關領域，其中大部份又用於製造業的前端作業，也就是應用於研發設計的流程上；藉由提供過去消費者需求的規格、設計的資料、裝配的功能和零件的特徵，使研發人員易於開發新產品；近期 CBR 之實際應用如汽車製造業的設計、發展電力設施原型系統之設施規劃、電腦輔助夾具設計系統、機械裝配裝配、機械設計等。

三、類神經網路

類神經網路的發明與現階段電腦的發明時間相距不遠，然而兩者的發展速度及應用程度卻相差甚巨，電腦在近半個世紀中高速發展，然而人腦在一秒之內所執行的複雜工作(例如辨識一張臉)，如讓現在的電腦來處理，卻顯得非常沒有效率與效能，於是以高度平行處理的類神經網路又獲得世人的青睞(Vasant & Merrick 1994)。

(一)類神經網路概述

類神經網路是一種模仿生物神經網路的資訊處理計算系統，它用大量簡單的相連人工神經元來模仿生物神經網路的能力，人工神經元是生物神經元的簡單模擬，它從外界或從其他神經元取得資訊並加以非常簡單的運算，而後以扇狀輸出結果給其他人工神經元，以此機制來處理資訊，並對外部的輸入以網路動態來回應(Stephen 1993)；依本論文修正案例庫的需要，比較各神經網路之內容功能如表 2：

表 2 類神經網路內容功能列表

網路模式	內容說明	特性功能
倒傳遞網路 Back-Propagation Network(BPN) 1974,P.Werbos	基本原理是以平滑可微分的轉換函數，應用最陡坡降法(the gradient steepest descent method)的觀念，將誤差函數予以最小化導出網路加權值；此網路模型具有隱藏層，通常依問題的複雜度而採用層數，並常以試驗方式決定最佳處理單元數目	學習速度慢但回想速度快，且學習精度高，為目前應用最普遍的類神經網路模型
雙向聯想記憶 Bi-directional Associative Memory(BAM)	此法之歸納能力是發揮在相關樣本的聯想，由輸入樣本回想起另一個配對樣本，再產生樣本輸出；此法似人類依循線索回憶的方式；1024 個神經元的 BAM 只能取用 24 對左右。	學習訓練時間較短，但迴響速度較慢
波茲曼機器 Boltzmann machine(BM) 1984,Hinton	應用統計物理學的模擬退火觀念，從霍普菲爾網路衍生出的網路模式，其神經元狀態為 0 與 1 二元值，而神經元則分可見與隱藏(如倒傳遞之隱藏層)，可見神經元可與所有神經元連結，而隱藏神經元只可與可見神經元連結；網路型態依輸出入方式而分自聯想與異聯想型態。	以統計方法提高學習品質，並解決霍普菲爾網路局部最小值的問題；然而執行成本大且速度較慢

(二)倒傳遞網路概述

倒傳遞網路是一個非常重要的多層感知(Multilayer Perceptron)網路，典型的倒傳遞網路有一層接收訊號的感知單元，有一層到數層(一般應用很少有超過三層的)不等的隱藏層運算單元，及一層輸出運算單元；多層感知網路被成功的應用在解決一些多元複雜的問題，而依據監督(supervised)學習的方法訓練網路具備解決問題的能力。

倒傳遞網路使用回傳誤差(error back-propagation)的演算法達到訓練的修正效果，並以修正誤差學習率作為此演算法的基礎；回傳誤差學習是由前傳(forward pass)及後傳(backward pass)兩種不同的網路傳遞方式所建構。前傳網路是經由調節網路節點的輸入向量，從輸入層往輸出層方向傳遞，真實地反應網路的表現；前傳網路到輸出層時，網路將經由輸出與目標的差量得到網路誤差修正的基礎，此誤差訊號將調整網路回傳時的修正結果，而回傳時所修正的權重值即如同生物神經之突觸重(Synaptic weight)的學習反應；網路在前傳時的權重值是固定的，而在後傳時的權重值，將會接受誤差修正的調適；一個多層感知網路有三種主要特徵(Patrick 1990)：

1. 在多層感知的網路模型中，除輸入層神經元外，每一個神經元都包括一個非線性的活化函數；活化函數的重點就在於它是用非線性的平滑曲線，有別於 Rosenblatt 的感知機網路所用的硬限函數，這種非線性活化函數是採用數學邏輯的 S 型非線性函數：

$$y_j = \frac{1}{1 + \exp(-v_j)} \quad (\text{附註 1})$$

2. 多層感知網路除了輸入層和輸出層外，還包括一層或多層的隱藏神經元，由於隱藏層的加入，使網路得以從事更複雜的工作，並能接受更多特徵及更具涵義的輸入。
3. 此網路經由神經鍵而使整體神經元高度的連結，其結構也十分的強韌，所以其修正與調整是整體性的，處理問題的能力也較為細緻。

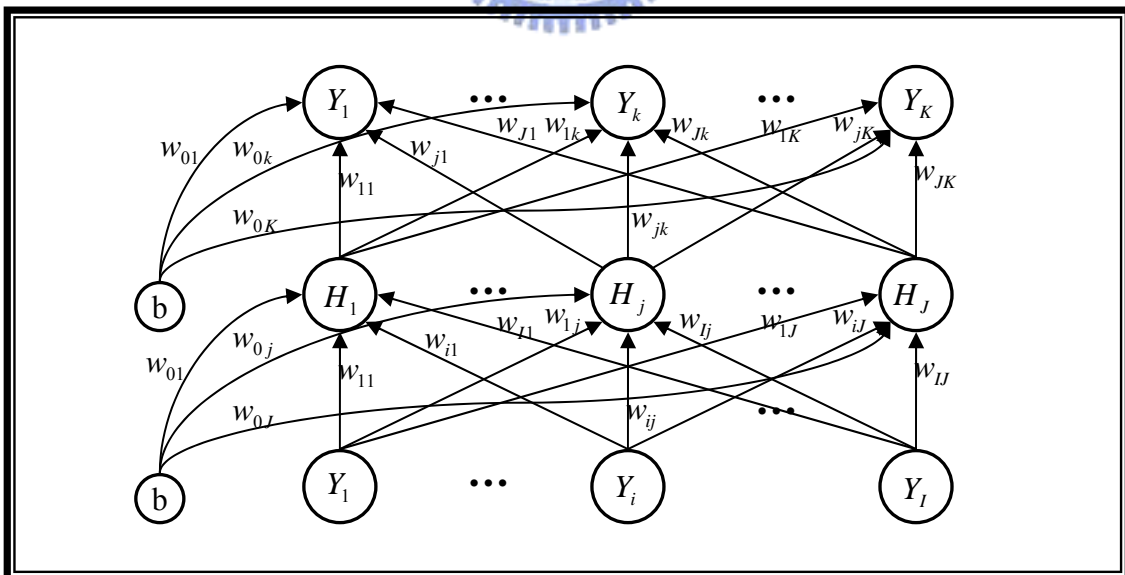


圖 7 具一層隱藏層的倒傳遞網路模型

附註 1：公式用於前傳網路的過程，其中 v_j 為 j 神經連接前一層各神經元之輸入與加權的乘積和； y_j 為神經元輸出值。

(三)倒傳遞網路原理

倒傳遞網路是使用回傳誤差(error back-propagation)的演算方法來達到訓練的修正效果；而輸出層神經元的誤差信號(error signal)來自於輸出值與目標值(Target)的不同，我們可以用 Y 代表輸出值並以 T 代表目標值，將目標值減去輸出值即得誤差值，得列式如下：

$$e_k(n) = t_k(n) - y_k(n) \quad (\text{公式 2.1})(\text{附註 2})$$

倒傳遞網路輸出層從隱藏層獲得信號後，直到取得誤差值之間的過程，如圖 8：

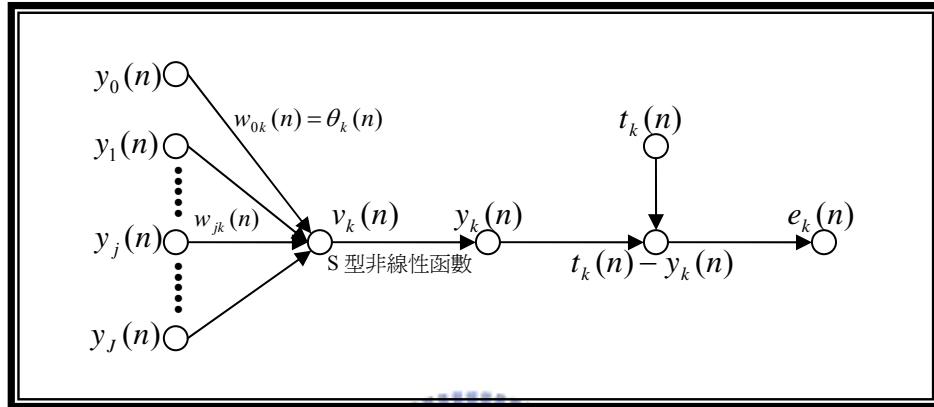


圖 8 誤差值取得之模型

在上圖中 $v_k(n)$ 為集乘函數，是輸入值與對應權重值的乘積和，可表示如下：

$$v_k(n) = \sum_{j=0}^J w_{jk}(n) y_j(n) \quad (\text{公式 2.2})(\text{附註 3})$$

經過 S 型非線性函數轉換後，即可得到輸出元的輸出值：

$$y_k(n) = f(v_k(n)) \quad (\text{公式 2.3})(\text{附註 4})$$

通常我們以即時加總輸出元的平方差來衡量網路學習的即時成效，其公式如下：

$$E(n) = \frac{1}{2} \sum_{k=0}^O e_k^2(n) \quad (\text{公式 2.4})$$

進一步我們可將公式(2.2)所求得的即時差量再加總平均，以作為衡量網路學習成效的具體依據：

$$E = \frac{1}{N} \sum_{n=1}^N (E(n)) \quad (\text{公式 2.5})(\text{附註 5})$$

附註 2：k 為輸出層神經元；n 為訓練樣本

附註 3：J 為 k 神經元的輸入單位總數。

附註 4：定義 O 含括所有的輸出層神經元

附註 5：N 代表訓練設定的樣本總數。

網路學習過程的目標就是要調適差量平均值(E)這個自由參數，以使其最小化。在輸出層取得誤差量，網路將隨之回傳並調整權重值以達到學習效果，接著我們來繼續思考網路如何回傳？如何使權重值得到修正，並使修正後達到差量平均值最小化的效果。

倒傳遞網路的權重值是採用即時斜率(instantaneous gradient)作為修正基礎，而此斜率是即時差量(E(n))微分與權重值($w_{ij}(n)$)微分的比率，採用微積分連鎖率，我們可以推得如下公式：

$$\frac{\partial E(n)}{\partial w_{jk}(n)} = \frac{\partial E(n)}{\partial e_k(n)} \times \frac{\partial e_k(n)}{\partial y_k(n)} \times \frac{\partial y_k(n)}{\partial v_k(n)} \times \frac{\partial v_k(n)}{\partial w_{jk}(n)} \quad (\text{公式 2.6})$$

在上式中 $\partial E(n)/\partial e_k(n)$ 根據公式(2.4)可以推得：

$$\frac{\partial E(n)}{\partial e_k(n)} = e_k(n) \quad (\text{公式 2.7})$$

而 $\partial e_k(n)/\partial y_k(n)$ 部份，根據公式(2.1)可得其結果為：

$$\frac{\partial e_k(n)}{\partial y_k(n)} = -1 \quad (\text{公式 2.8})$$

其次 $\partial y_k(n)/\partial v_k(n)$ 部份，根據公式(2.3)可推其式為：

$$\frac{\partial y_k(n)}{\partial v_k(n)} = f'(v_k(n)) \quad (\text{公式 2.9})$$

最後 $\partial v_k(n)/\partial w_{jk}(n)$ 的部份以公式(2.2)推得：

$$\frac{\partial v_k(n)}{\partial w_{jk}(n)} = y_j(n) \quad (\text{公式 2.10})$$

將公式(2.7)到(2.10)帶入公式(2.6)中，可以推得即時斜率為：

$$\frac{\partial E(n)}{\partial w_{jk}(n)} = -e_k(n) f'(v_k(n)) y_j(n) \quad (\text{公式 2.11})$$

求得即時斜率後，將之乘上一特定學習率(learning rate)後即為權重值的修正量：

$$\Delta w_{kj}(n) = -\eta \frac{\partial E(n)}{\partial w_{jk}(n)} \quad (\text{公式 2.12})(\text{附註 6})$$

我們可將輸入值部份自即時斜率中分離出來，並將其餘部份用一局部斜率(local gradient)含括，以便於網路後傳使用，根據(2.11)及(2.12)兩公式，可得：

$$\Delta w_{kj}(n) = \eta \times [e_k(n) f'(v_k(n))] \times y_j(n) = \eta \delta_k(n) y_j(n) \quad (\text{公式 2.13})$$

附註 6： $\Delta w_{kj}(n)$ 代表輸出層到隱藏層間之權重值修正量； η 代表學習率；用負號是為使即時斜率下降。

由上式可以得知局部斜率($\delta_k(n)$)為調適權重的關鍵係數，並且包含輸出元的誤差信號及輸出值的微分。在網路中 k 神經元是局部的輸出層神經元，它的輸出值與目標值是單一而直接「點」的影響關係，但在隱藏層中的神經元所接受的誤差信號，是由輸出層或上一層隱藏層所傳遞的，也就是說它必須整合處理上一層每個神經元的反應，這就牽涉到上一層「面」的影響關係，因此必要將之區分。

由圖 9 可知隱藏層神經元的局部斜率，並非個別從目標值獲取誤差值，因此我們須從新定義隱藏層的局部斜率；我們可以引用公式(2.7)及(2.8)並消去其中誤差值得部份，即可從新定義隱藏元 j 的局部斜率：

$$\delta_j(n) = e_j(n) f'(v_j(n)) = -\frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} f'(v_j(n)) = -\frac{\partial E(n)}{\partial y_j(n)} f'(v_j(n)) \quad (2.14)$$

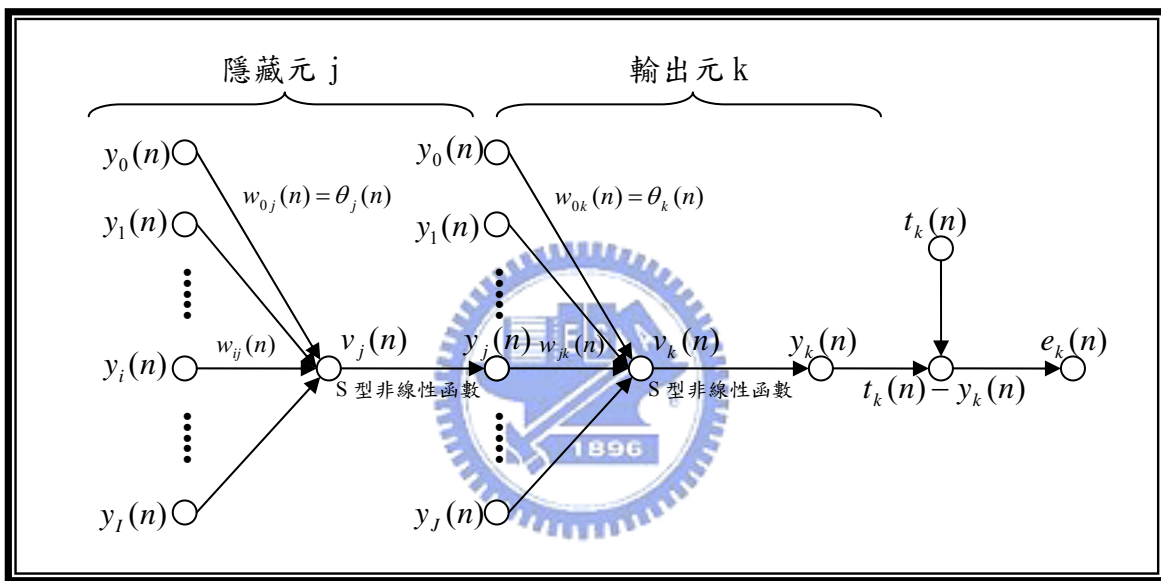


圖 9 由隱藏元 j 至誤差取得之模型

將公式(2.4)與相關聯的信號 $y_j(n)$ 匯入，以分析 $\partial E(n)/\partial y_j(n)$ ，可列式如下：

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial y_k(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \quad (公式 2.15)$$

繼續我們對公式(2.1)進行演繹，來解決 $\partial e_k(n)/\partial v_k(n)$ 的問題：

$$e_k(n) = t_k(n) - y_k(n) = t_k(n) - f'(v_k(n))$$

同時對兩邊微分：

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -f'(v_k(n)) \quad (公式 2.16)$$

由公式(2.2)可以求得 $\partial v_k(n)/\partial y_j(n)$ 如下：

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{jk}(n) \quad (公式 2.17)$$

我們將公式(2.16)及(2.17)帶入公式(2.15)中，即可獲得下式：

$$\begin{aligned}\frac{\partial E(n)}{\partial y_j(n)} &= -\sum_k e_k(n) f'(v_k(n)) w_{jk}(n) \\ &= -\sum_k \delta_k(n) w_{jk}(n)\end{aligned}\quad (2.18)$$

將公式(2.18)帶入公式(2.14)中，得到隱藏元重整後的局部斜率如下式：

$$\delta_j(n) = f'(v_j(n)) \sum_k \delta_k(n) w_{jk}(n) \quad (2.19)$$

最後將公式(2.19)帶入公式(2.13)中，即可獲得隱藏層部份的權重值修正量：

$$\Delta w_{ij}(n) = \eta \delta_j(n) y_i(n) \quad (2.20)$$



附註 7: $\Delta w_{ji}(n)$ 為隱藏層到輸入層間之權重值修正量； $y_i(n)$ 為隱藏元 j 的輸入信號。

四、資料採礦

(一)資料採礦之概述

資料採礦(Data Mining)是利用統計原理和機器學習技術來建立自動預測角色行為的模型，從一大堆的資料中找出相關的模式(Relevant Patterns)，自動萃取出可預測的資訊，並利用這一些資訊輔助決策，依據使用者需求自資料中選擇適合資料，加以處理、轉換、採礦至評估的一連串過程，期望能找出真實世界運行時隱含於其內的運作現象以輔助解決問題之用(Fayyad 1996)。資料採礦(Data Mining)是知識發現(Knowledge Discovery in Database; KDD)過程中的一個步驟階段(Adriaans & Zantinge 1996)，由作為資料採礦的演算法則與機器學習所組成，從歷史資料中自動化或半自動化地採礦出未知或隱藏的資訊、樣板、結構或關聯；資料採礦之標的可分為描述(Descriptive)性與預測(Predictive)性兩種類型，描述性資料採礦是從資料庫中描述資料的屬性，預測性資料採礦會由目前的資料預先產生對未來的推論(Han & Kamber 2001)。資料採礦的功能是從資料中取出特定知識，藉由資料採礦各種不同的功能來實現知識發現(KDD)之精確度與良好的預測性，(Chen 2001)其功能可概略劃分如下：

1. 法則與樹歸納(Rule and Tree Induction)：法則歸納用於產生關於商業用途的樣板模式(Patterns)，而歸納樹是最被廣泛使用與目標導向發現(Goal Driven Discovery)的資料採礦技術。
2. 特性描述(Characterization)：類似於估算(Estimation)，適用於處理連續性數值，目標在於一般化及概括化資料的特性。
3. 分類(Classification)：分析定義新事件的所有特質，經訓練後指派至一現有群集中。
4. 回歸(Regression)：與分類最大的不同在於它被用來預測連續性的屬性。
5. 關聯(Association)：主要功能是發覺哪些事物總是同時發生，經常用於購物藍分析(Market Basket Analysis)與縱橫銷售(Cross Selling)。
6. 集群(Clustering)：將一群一直的群體區隔為同性質較高的群集或子集群，其不同於分類之處在於它並沒有依靠事先明確定義的類別進行分類；群集通常也可作為其他資料採礦建置模型的前置作業(Berry & Linoff 1997)。

(二)資料採礦的架構

資料採礦的整體架構一般包括使用者溝通介面、資料庫、應用領域知識、採礦出之知識及資料採礦之方式等，其運作流程之架構如圖 10 所示：

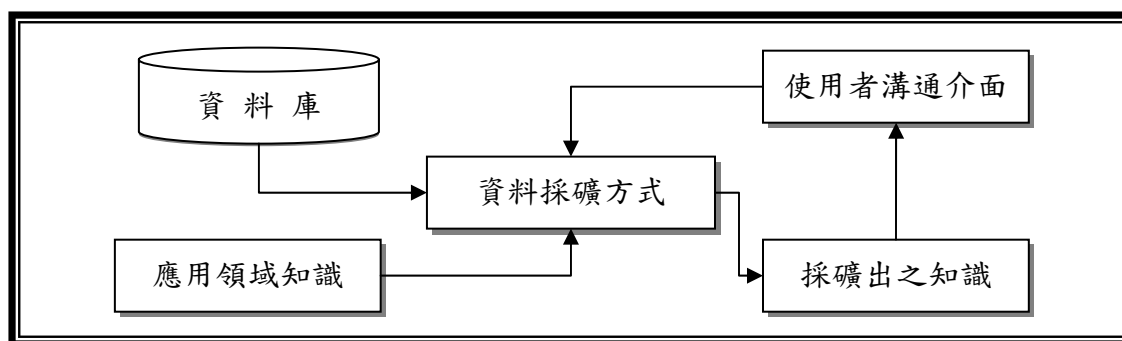


圖 10 資料採礦之架構

五、全域最佳化

(一)最佳化問題概述

最佳化是管理科學的一項主要領域，強調發展數學模型來描述現實問題的狀態，並利用人類的智慧去判定解釋其產生的結果(Reid & Sanders 2005)。一個標準的最佳化問題，有使目標最大化或最小化的需求，還有限制式使目標符合現實的運作，決策變數符合所須控制的特殊量。

最佳化問題關鍵要點說明：

1. 目標(Objective)：使量最大化或最小化。
2. 條件(Constraints)：必須滿足的限制或需求。
3. 決策變數(Decision variable)：作決策者的控制量。
4. 強迫最佳化問題(Constrained optimization problem)：當條件被滿足或限制時，以數學模式試者去使目標量最大或最小化。

例如一個學習績效的問題，我們希望最大化平均等級，限制式為總學習時間的限制，決策變數則是各課程時間的分派，根據最佳化問題模式，可描述問題如下：

1. 最大化目標：平均成績等級。
2. 限制條件：學習在可用的學習時間範圍內，每一課程的成績等級都超過最低等級C。
3. 決策變數：花再每一課程上的總時間。

最佳化問題被三個基礎要素所組合(Miettinen 2004)：

1. 目標方程：最小或最大 (例如：製造過程，盡力使利潤最大、總成本最小)
2. 未知變數設定：對目標方程造成影響的值 (例如：製造問題，在各項活動的條件中，盡量包括不同的使用來源或次數總額變數。
3. 限制式設定：允許未知變數在確定值的範圍內 (例如製造問題，任何製造活動所花費的時間總值不可能為負，如此需強迫所有的時間變數為非負變數。

(二)全域最佳化概述

全域最佳化是利用數學公式找出有資格達到所求的目標條件的絕對最佳值，當最佳化問題為非線性時，利用數學模式去求區域最佳解雖較為容易，但是求全域最佳解則相當困難。然而利用啟發技術則能使求解全域最佳化問題便的較為容易，譬如透過禁忌搜尋(tabu search)、接受門檻(threshold accepting)等(Horst & Pardalos 1995)。全域最佳化是有關於求解全域最佳特性描述的非線性方程，應用數學模型、演算法和計算方法，幫助實際應用於解決世界上多樣化的問題；全域最佳化提供策略和數值程序去分析解答非線性最佳化問題。全域最佳化這個領域是用數學的方法找出非線性條件和全域最佳化問題的完整解答，許多科學研究與工程應用都需要使用者去解決非線性限制式或最佳化非線性方程目標的問題(Horst et. al. 2000)。

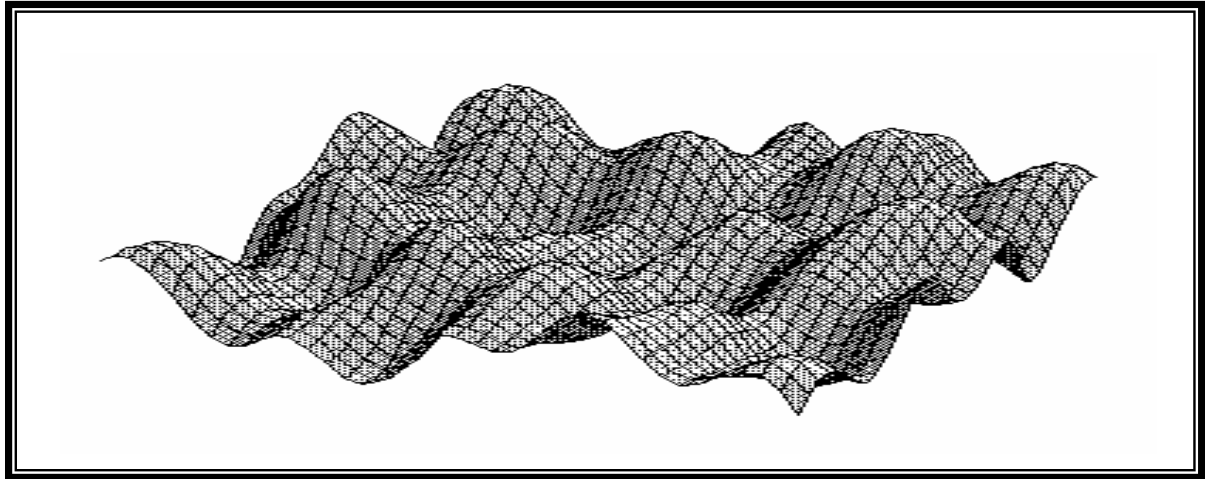


圖 11 一非線性問題之 3D 圖

(三)線性最佳化技術

累進線性技術(piecewise linearization techniques)(Li et. al. 2002)採用線性逼近最佳解模式，是一種非常有用且強而有力的概念，其模型類於泰勒展開式，但卻更進一步將之設計成，能於解決全域最佳化問題的模式。

一個對數方程式問題，能夠表示如圖12，其中 $0 < a_1 \leq x \leq a_m$ ，以累進線性(piecewise linearization)模式可得近似值如下：

$$\ln x \doteq \ln \hat{x} = \ln a_1 + s_1(x - a_1) + \sum_{j=2}^{m-1} \frac{s_j - s_{j-1}}{2} (|x - a_j| + x - a_j) \quad (\text{公式2.21})$$

註： $a_j, j=1,2,\dots,m$ ，為 $f(x)$ 的截斷點，且 $a_j < a_{j+1}$ ； s_j 為介於 a_j 與 a_{j+1} 的斜線部分，可表示算式如下：

$$s_j = \frac{f(a_{j+1}) - f(a_j)}{a_{j+1} - a_j} \quad j=1,2,\dots,m-1 \quad (\text{公式2.22})$$

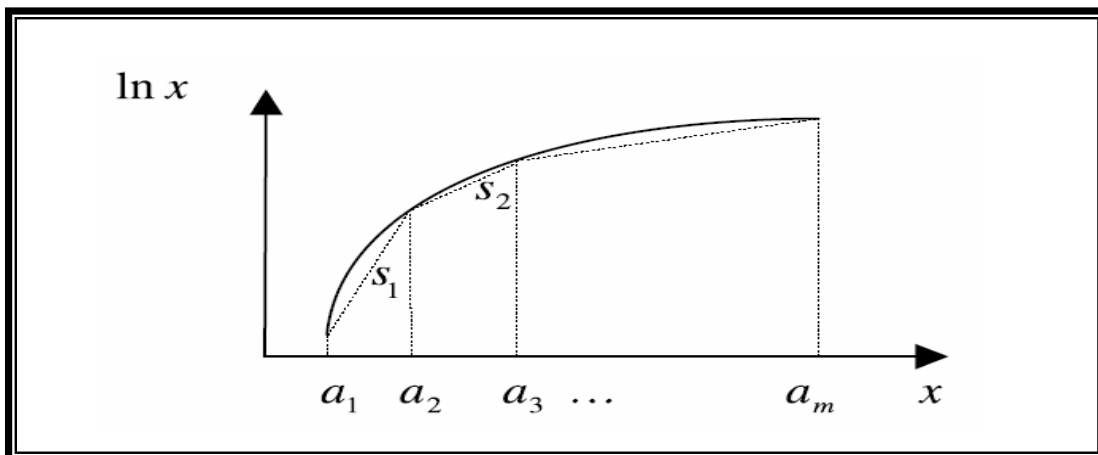


圖 12 $f(x)$ 的線性累近圖