

### 第三章 倒傳遞網路

人工神經網路或譯為類神經網路，是由人類心智和腦部活動所發展出來的一種模型。它的組成單元就像生物的神經元一樣，具備有接收、產生、傳遞和處理訊息的基本功能，這些單元組織成像腦神經一樣的構造，擁有類似腦神經的特性。就網路的架構而言，它是由許多簡單而且互相連結的處理單元所組成；而就網路的功能而言，它是一種由生物學得到靈感所啟發而產生出來的新型態資訊處理與計算方式[葉怡成 1997,1998]。

#### 3.1 類神經網路概論

類神經網路是一種平行計算系統，它包含軟體與硬體，且具有使用大量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，它從外界環境或其它人工神經元取得訊息來加以計算，並將其結果輸出到外界環境或其它人工神經元。換句話說，類神經網路可以利用一組或多組過去的範例(包含系統輸入與系統輸出資料)來建立模型，並將此系統模型用於預測、推估、決策或診斷上，因此類神經網路也可以看成是一種特別型式的統計技術。

神經網路的研究已有近四十年的歷史了，從早在 1957 年，第一種神經網路模式-感知機模式就被提出了，至今已有不下二十種網路模式。本研究則是以倒傳遞神經網路(Back Propagation Network, BPN)為應用模式。

### 3.2 BPN 理論

BPN 是目前類神經網路學習模式中最具代表性，應用最普遍的模式，已發表的應用案例甚多。BPN 基本原理是利用最陡坡降法（Gradient Steepest Descent Method）的觀念將誤差函數予以最小化。與其它模式的類神經網路相比較之下，有下列優點：

#### 1. 添加隱藏層之使用

使得網路可表現輸入處理單元間的交互影響。

#### 2. 改用平滑可微分的轉換函數

使得網路可應用最陡坡降法導出修正網路權重值的公式。

#### 3. 學習精度高

可處理複雜的樣本識別問題與高非線性的函數合成問題。

#### 4. 回想速度快

#### 5. 應用範圍廣

因為輸出值為連續值，適用於樣本識別、分類問題、函數合成、適應控制、雜訊過濾、資料壓縮、專家系統等應用，應用範圍最廣。

BPN 的基本架構包含處理單元、層、網路，處理單元為網路組成的基本單位，層為若干個具相同作用的處理單元集合而成，網路為若干個具不同作用的層集合而成。其中層又分為輸入層、隱藏層與輸出層，而各層中又包含數個處理單元，不同層中的單元又以權重值與閾值互相連結。輸入層為使用線性轉換函數表現網路的輸入變數，其處理單元數量依問題而定。隱藏層為表現輸入單元間的交互影響，其數量並無一定，通常測試各參數所產生的結果以挑選出網路最佳化來決定。輸出層為使用雙曲線轉換函數來處理網路的輸出變數，其處理單元數量亦依問題而定。

應用倒傳遞網路時，可能遭遇的問題包括：

- 1.局部最小值會影響誤差函數的最小化過程處理，使得收斂不完善。
- 2.學習過程中，處理單元（變數）達到飽和時，使得該處理單元無法發揮修正與其相連的連結加權值的效果。
- 3.減緩現象是因為加權值的改變量與誤差函數曲面的坡度成正比，在接近曲面最低點時，坡度可能逐漸減小，因而減小加權值的改變量，最後近乎停滯。反之，若誤差函數曲面存有峽谷，在接近峽谷時，坡度可能很大使加權值的改變量大大的跳到峽谷的另一邊，如此兩邊來回跳動，反而使誤差函數值加大，稱跳出現象。
- 4.暫時性的不穩定，此現象是如果訓練範例事先未加以隨機混合，而將各類分類的訓練範例依序集中載入網路，可能使網路學了後類分類規則時，卻遺忘了前類分類規則不完善。則，如此下去始終無法將各類分類規則均正確學習。解決方法為將單元順序隨機組合，以克服此暫時不穩定狀況。
- 5.執行時間較長：執行時間受學習範例尺度大小而異。
- 6.嘗試各種網路架構、網路動態皆無法收斂，訓練範例的輸入向量與輸出向量根本無充份關連可供學習，或訓練範例對整個問題領域而言，代表性不足。

### 3.3 BPN 基本架構

圖 3.1 為 BPN 之基本架構，含處理單元(processing element)、層(layer)、網路(network)，相同的處理單元集成層，不同作用之層結合成網路。各對此三個基本架構說明如下：

#### 一、處理單元

處理單元為組成類神經網路的基本單元，其作用可用集成函數

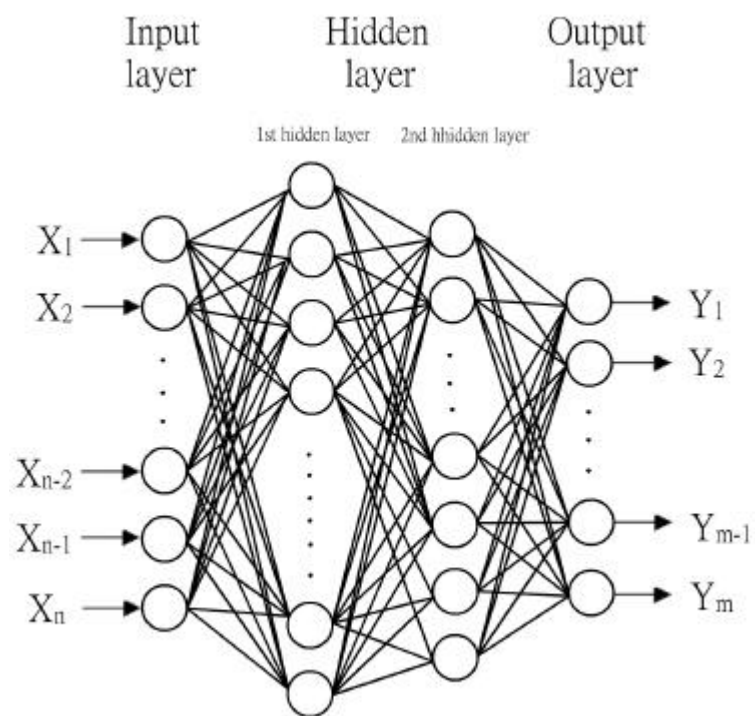


圖 3.1 BPN 基本架構

(summation function)、作用函數(activity function)與轉換函數(transfer function)來說明。

### 1.集成函數

集成函數是將其它處理單元之輸出，透過網路之傳送，將傳送來的資訊加以綜合。

### 2.作用函數

作用函數是將集成函數與目前處理單元的狀態加以結合。

### 3.轉換函數

轉換函數是將作用函數的輸出值轉換成處理單元的輸出。

## 二、層

層可分為輸入層、隱藏層與輸出層：

### 1.輸入層

網路的輸入變數，其處理單元目依問題而定，在倒傳遞類神經網路中使用線性轉換函數，即  $f(x) = x$ 。

### 2.隱藏層

網路輸入處理單元間的相互影響，其處理單元數目並無標準方法可以決定，必須以試驗方法決定其最佳數目數。

### 3.輸出層

網路的輸出變數，其處理單元數目依問題而定，使用非線性轉換函數。倒傳遞類神經網路最常用的非線性轉換函數為雙曲線函數，即

$f(x) = \frac{1}{1 + e^{-x}}$ 。此種函數在自變數趨近於正負無限大時，函數值趨近於常數，即函數值域在  $\{ 0,1 \}$  之間。

## 三、網路

不同作用的層集成網路，網路本身具備有學習(learning) 與回想(recalling)兩種作用。

## 1.學習過程

網路從問題領域中取得訓練範例與目標輸出值，並依學習演算法從中學習，以調整網路連結加權值之過程，其中範例含輸入變、輸出變數。

## 2.回想過程

網路從問題領域中取得測試範例或待推範例，並依回想演算法，以輸入變數來決定網路輸出變數的過程，其中待推範例只含輸入變數。

### 3.4 BPN 演算法

以一個具有單層隱藏層的層狀類神經網路為範例，說明倒傳遞演算法如何應用一個訓練範例的一組輸入值，與一組目標輸出值，修正網路連結加權值與門限值，而達到學習的目的。

在倒傳遞網路中，第  $n$  層的第  $j$  個單元輸出值為第  $n-1$  層單元輸出值的非線性函數：

$$A_j^n = f(net_j^n) \quad (3.1)$$

其中：

$$net_j^n = \text{集成函數} = \sum_i W_{ij} A_i^{n-1} - \theta_j$$

$W_{ij}$ =模仿第  $i$  個與第  $j$  個神經元之間的連結強度，即連結加權值。

$A_i^{n-1}$ =模仿第  $i$  個神經元傳來的輸入訊號。

$\theta_j$ =模仿第  $j$  個神經元的閾值。

$f$ =模仿生物神經元的模型的轉換函數或稱階梯函數。

因為監督式學習旨在降低網路輸出單元目標輸出值與推論輸出值之差距，為了瞭解學習的品質，所以一般以能量函數（或稱誤差函

數)  $E$  表示如公式(3.2)。

$$E = \left( \frac{1}{2} \right) \sum_j (T_j - A_j)^2 \quad (3.2)$$

其中  $T_j$ =輸出層目標輸出值

$A_j$ =輸出層推論輸出值

因此網路的學習過程變成使上述能量函數最小化的過程，通常以最陡坡降法來使能量函數最小化，即每當輸入一個訓練範例，網路即小幅調整加權值的大小，調整的幅度和誤差函數對該加權值的敏感程度成正比，即與誤差函數對加權值的偏微分值大小成正比：

$$\Delta W_{ij} = -\eta \cdot \frac{\partial E}{\partial W_{ij}} \quad (3.3)$$

$W_{ij}$ =介於第  $n-1$  層的第  $i$  個處理單元，與第  $n$  層的第  $j$  個處理單元間的連結加權值。

$\eta$ =學習速率 ( learning rate )，控制每次以最陡坡降法最小化誤差函數的步幅。

一般將  $\frac{\partial E}{\partial W_{ij}}$  寫成公式(3.4)，

$$\frac{\partial E}{\partial W_{ij}} = -d_j^n \cdot A_i^{n-1} \quad (3.4)$$

$A_i^{n-1}$ = $W_{ij}$  所連接之較低層處理單元的輸出值

$d_j^n$ = $W_{ij}$  所連接之較上層處理單元之差距量

如果  $W_{ij}$  所連接之較上層處理單元為輸出層，則  $d_j^n$  可表示為公式

(3.5)。

$$\ddot{a}_j^n = (\hat{O}_j - \hat{A}_j) \cdot f'(net_j^n) \quad (3.5)$$

否則  $\ddot{a}_j^n$  表成公式(3.6)。

$$\ddot{a}_j^n = \left( \sum_k \ddot{a}_k^{n+1} \cdot W_{jk} \right) \cdot f'(net_j^n) \quad (3.6)$$

若將公式(3.4)代入公式(3.3)得公式(3.7)

$$\Delta W_{ij} = \eta \cdot \ddot{a}_j^n \cdot A_i^{n-1} \quad (3.7)$$

公式(3.7)即為倒傳遞網路最陡坡降法演算式之關鍵公式。

### 3.5 BPN 執行程序

BPN 演算法分為學習過程與回想過程兩部分：

#### 1. 學習過程

步驟(1).決定網路層數及各層間神經元數目

步驟(2).以均佈隨機亂數設定網路的初始加權值及初始偏權值。

步驟(3).輸入訓練範例的輸入變數值與目標輸出值。

步驟(4).計算推論輸出值。

步驟(5).計算輸出層與隱藏層的差距量。

步驟(6).計算各層間加權修正量與偏權修正量。

步驟(7).更新各層間的加權值及偏權值。

步驟(8).重覆步驟(3)至步驟(7)直到收斂(不再有明顯之誤差)或到



學習循環結束。

## 2. 回想過程

步驟(1). 設定網路參數及各層間神經元之數目。

步驟(2). 讀入學習過程中已訓練好的網路加權值與偏權值。

步驟(3). 輸入測試範例的輸入變數值。

步驟(4). 計算推論輸出值。

## 3.6 BPN 測試

BPN 學習過程通常以一次一個訓練範例的方式進行，直到學習完成所有的訓練範例稱為一個學習循環，一個網路可以將訓練範例反覆學習數個學習循環直到收斂為止。故當網路建構完成後，網路須加以測試，網路測試可分成收斂、驗證、確認、評估測試，四個階段。

### 一、收斂測試

在網路學習過程中監督網路的誤差函數或能量函數是否合理收斂需測試。在監督式學習類神經網路學習過程中，收斂過程通常以均方根來衡量，表示如公式(3.8)。

$$\text{均方根} = \sqrt{\frac{\sum_p \sum_j (\hat{O}_j^p - \tilde{O}_j^p)^2}{M \cdot N}} \quad (3.8)$$

其中  $\hat{O}_j^p$  = 第 p 個範例的第 j 個輸出單元之目標輸出值。

$\tilde{O}_j^p$  = 第 p 個範例的第 j 個輸出單元之推論輸出值。

M = 範例數目

N = 輸出層處理單元數目

均方根的值域在 0.0~1.0, 但實際上在迭代之初始值應為 0.5 左右。

觀察收斂的過程必須同時考量訓練範例以及測試範例的收斂過程，一般收斂過程可分為四類：

第一類收斂：訓練範例的誤差收斂良好，且測試範例的誤差收斂良好。如圖 3.2 (a)

第二類收斂：訓練範例的誤差收斂良好，但測試範例的誤差收斂不好。如圖 3.2 (b)

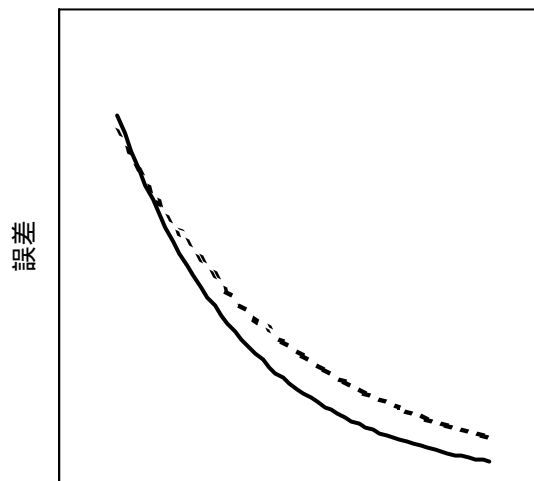
第三類收斂：訓練範例的誤差收斂不好，但測試範例的誤差收斂良好。如圖 3.2 (c)

第四類收斂：訓練範例的誤差收斂不好，且測試範例的誤差收斂不好。如圖 3.2 (d)

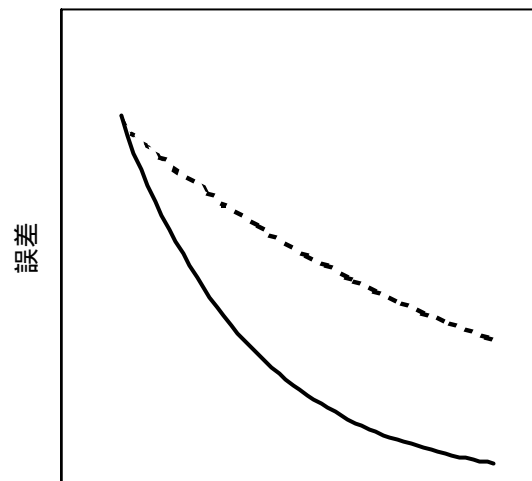
## 二、驗證測試

測試在網路學習完成後分析網路的誤差的特徵，以作為改進網路學習效果的依據。於 BPN，其輸出層處理單元的推論輸出值與目標輸出值均為實數，因此可用散佈圖來作驗證測試。散佈圖乃以橫軸為輸出層處理單元的目標輸出值，縱軸為輸出層處理單元的推論輸出值，所繪之關係圖如圖 3.3 所示。散佈圖上的點分佈在圖的對角線上附近則結果良好。此外也可用相關係數來衡量散佈圖上的點集中在對角線上的程度。其公式 R 為：

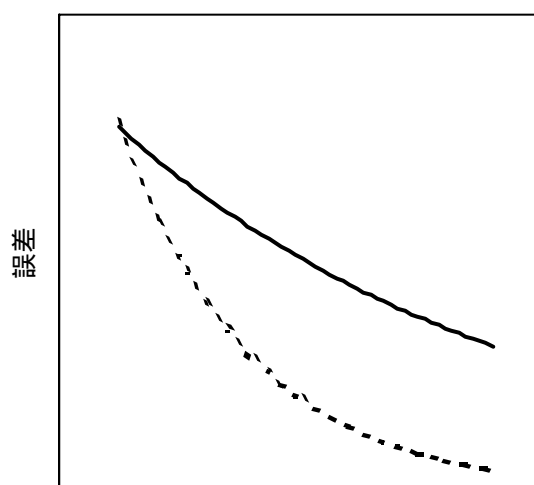
$$R = \frac{(\sum T_i Y_i) - n \bar{m}_i \bar{m}_y}{(n-1) s_i s_y} \quad (3.9)$$



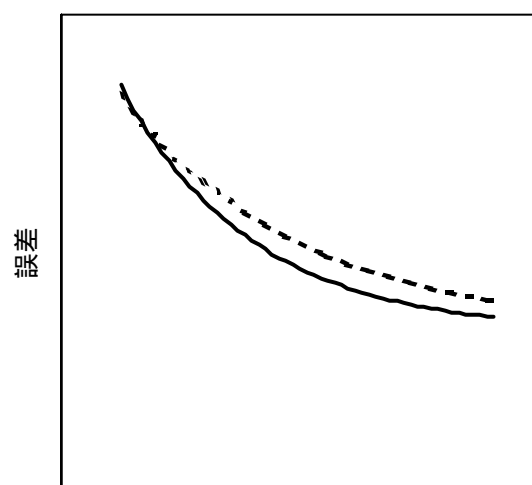
(a) 第一類收斂圖



(b) 第二類收斂圖



(c) 第三類收斂圖



(d) 第四類收斂圖

圖 3.2 BPN 收斂種類

其中  $\sigma_t$  = 目標輸出值標準差 =  $\sqrt{\frac{\sum_i (t_i - m)^2}{n-1}}$

$\sigma_y$  = 推論輸出值標準差 =  $\sqrt{\frac{\sum_i (y_i - m)^2}{n-1}}$

$\mu_t$  = 目標輸出平均值。

$\mu_y$  = 推論輸出平均值。

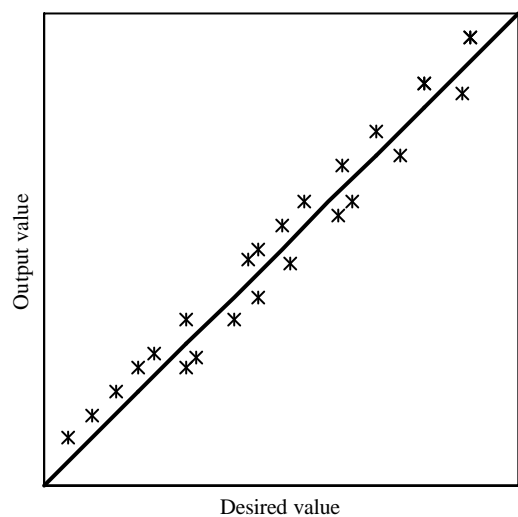
相關係數的值域在[-1.0, 1.0]，實際上在最差的狀況也應在 0.0 左右，如在 0.8 以上可視為結果良好。

### 三、確認測試

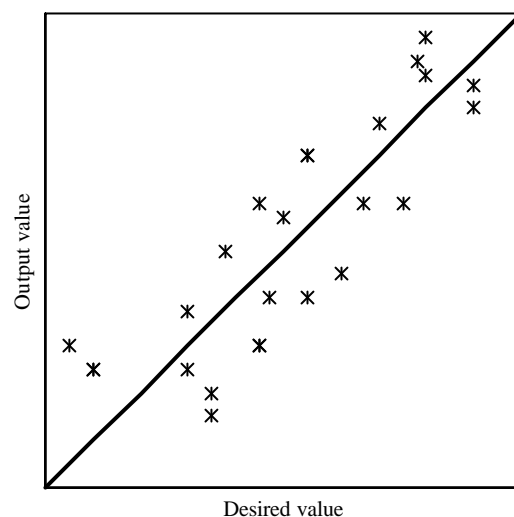
確認在網路學習完成後分析網路的結果與其它網路模式或其它技術的所得結果相較之下優劣。

### 四、評估測試

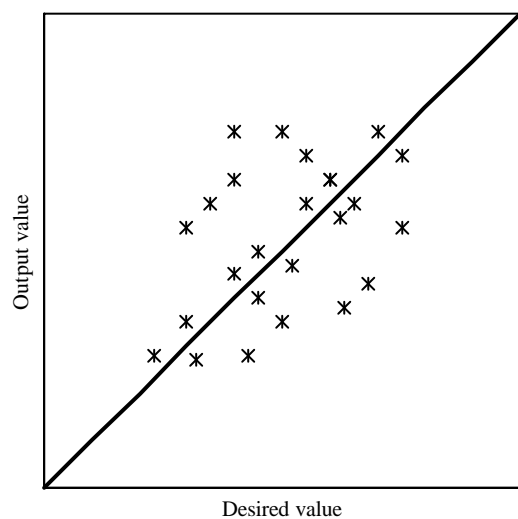
評估在系統完成後分析系統的結果是否滿足專案目標的需求。本研究評估測試方面為與試驗數據比較且對非訓練測試範例做模擬，最後進一步將 BPN 殖入 FEM 並與雙曲線模式比較。



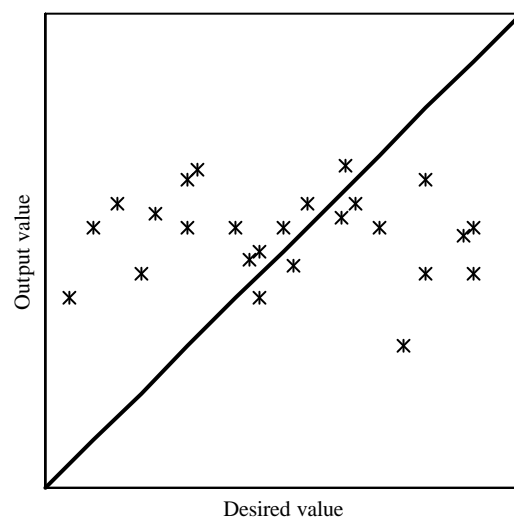
(a) 良好的散佈圖



(b) 略差的散佈圖



(c) 不良的散佈圖



(d) 不良的散佈圖

圖 3.3 BPN 散佈圖示意圖