權重衰退學習與明確正規化於 RBF 中之容錯訓練

羅文和 沈培輝 黃永發
Wun-He Lu John Sum Yung-Fa Huang
朝陽科技大學 國立中興大學 朝陽科技大學
s9530608@mail.cyut.edu.tw pfsum@nchu.edu.tw yfahuang@mail.cyut.edu.tw

摘要

本文乃探討幅狀基底函數(Radial Basis Function, RBF)網絡經由權重衰退學習(Weight Decay Learning, WDL)與明確正規化學習(Explicit Regularization Learning, ERL) 訓練後對加乘性權重雜訊(Multiplicative Weight Noise, MWN)容錯(fault tolerant)效果。從數學分析證明，在有足夠的訓練數據之下，WDL與ERL對MWN的容錯性，理論上有著同樣的效果。這結果提供了一個有力的證據，解釋了為何WDL可以被應用在容錯訓練。[1]

關鍵詞：幅狀基底函數、權重、容錯、權重衰退遞減學習法，明確正規化學習法、加乘性權重雜訊。

Abstract

Abstract — This paper studies the fault tolerant performance of a radial basis function (RBF) network being trained to against the multiplicative weight noise (MWN) use weight decay learning (WDL) and explicit regularization learning (ERL). Theoretically, it is shown that for enough training data, the fault tolerant abilities of using WDL and ERL against MWN, are provided to support the theorem.

Index Terms — radial basis function (RBF), multiplicative weight noise (MWN), weight decay learning (WDL), explicit regularization learning (ERL), fault tolerant.

1. 前言

人的智慧自於不斷的學習與成長，同樣的，若想讓類神經網路發揮其功能，一定要有良好的訓練學習機制。由獲取知識的途徑通常可利用已學習和已知的輸出或預測的輸出值之間的偏差(deviation)值，來調整類神經網路(Neural Network, NN)訓練的方式[1]。容錯技術和可靠性或穩健性(robustness)都是類神經網路中相關重要的議題[2,3,4,5,10,11]。然而，容錯式系統的主要精神就是當不管是軟體或硬體上的錯誤發生時，系統能繼續且正確地執行某特定的工作。現在的電腦整合到了商業、航管、軍事、工業控制等等，這些方面只要有一點點的錯誤，影響到的是大筆的資金、環境的安全，甚至是人類的生命。所以容錯系統之所以愈來愈重要只是因為電腦處理的問題愈來愈嚴厲了。因容忍錯誤影響到結構性因素是完全不同於容忍雜訊而影響輸入值。前者是容錯率的問題，後者是因穩健性的雜訊而干擾輸入。

利用 FPGA 合成類神經網路，一般上有兩種錯誤(fault)。分別為神經元死亡(node dead)[10,12,14]和權重值受雜訊干擾(weight noise)。而權重雜訊(weight noise)更可分為外加性(additive)和加乘性(multiplicative)兩種。在本文中著重於加乘性權重雜訊(Multiplicative Weight noise, MWN) [8,9,11,13,15]。

類神經網路能否適切地展現對現象的模擬與解析的功能，主要在於神經元模型的設計與各神經元間連結權重的妥適與否。學習演算法就是一套權重調整的演算法，藉由演算法逐步地調整神經元間連結的權重，使其達到最佳數的數值。在類神經網路中通常需要不同的權重值收斂而達到良好的歸納性。而調整所有這些衰退常數，以產生最佳估計歸納誤差往往需要大量的計算複雜度和儲存量[1]。因此經過權重衰退學習(Weight Decay Learning, WDL)能夠有效提高網路穩健性和歸納能力[6]。

幅狀基底函數(Radial Basis Function, RBF)的架構及相關的研究問題，將會描述在下一段落，第三段說明 RBF 針對 MWN 的解決方式，第四段介紹 WDL[6]與明確正規化(Explicit Regularization, ERL)[8]兩種學習演算法，而第五段則是提出方法使一維與二維向量能在學習演算法上能達到相同的容錯能力，而第六段是模擬數據的結果呈現，本文的結論在最後一段說明。

[1] 通訊作者:沈培輝

2. 研究問題

圖 1 呈現的為一組假設的測量數據，而這些數據是由一系統所製造出來的(如圖 2)。

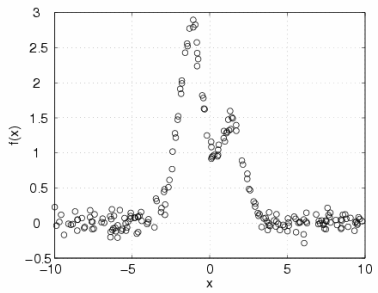

圖 1 測量數據

這系統由輸入值 x 經由非線性轉換 $g(x)$ 得出的結果加上外部的干擾因而產生的輸出值 y。一般而言，假設 $g(x)$ 為已知，只要求出外部干擾值，我們便可準確地得出這系統的行為，並因以預測。但在大部分的情況上，$g(x)$ 都是難以得知。研究者都會假定 $g(x)$ 為某一種模型，如(多項式、傅立葉級數、模糊系統、類神經網路)。然後利用測量數據去估計模型的參數。在本文，我們定義這 $g(x)$ 為幅狀基底函數網路模型[1]。

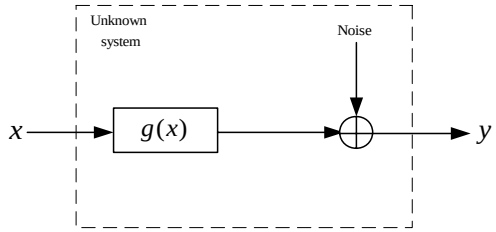

圖 2 模擬環境示意圖

而本文中所面臨到的問題是：當權重值受到不同幅度的擺動之下，我們必須將預測的誤差值要變得更小。在圖 3 中所呈現的是我們模擬的環境情況。首先在未知的系統裡找到合適的模式去訓練資料量，其中有兩種情況，第一：在未加任何特殊條件下輸入為 x，權重為 $\hat{w}$，得到的輸出為 $\hat{y}$；第二種情況在額外賦予其他的條件因素 $\tilde{w}$ 而使整個訓練過程有不同的效果呈現，此時輸出則以 $\tilde{y}$ 表示。

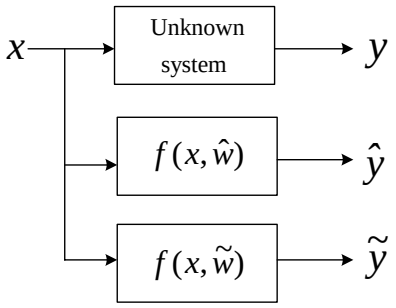

圖 3 模擬環境的問題分析

3. Multiplicative Weight Noise

幅狀基底函數網路是屬於基本的前饋式類神經網路(Feedforward Neural Network, FNN)，其架構包含輸入層、單一隱藏及輸出層。主要概念是建構許多幅狀基底函數，以函數逼近法(curve fitting)找出輸入/輸出間的映射關係，也就是在隱藏層中的各神經元建立相對應的幅狀基底函數 $\phi(\|x-c\|)$，其中所代表涵義如(1)式：

$$\phi(x-c)=\exp(-\frac{(x-c)^2}{\sigma}) \qquad (1)$$

式中 c 表示神經元的中心，σ 是指神經元的寬度。在圖 4 中包含 N 個維度的輸入層、M 個神經元的隱藏層與單一輸出的輸出層作為 RBF 架構的說明。

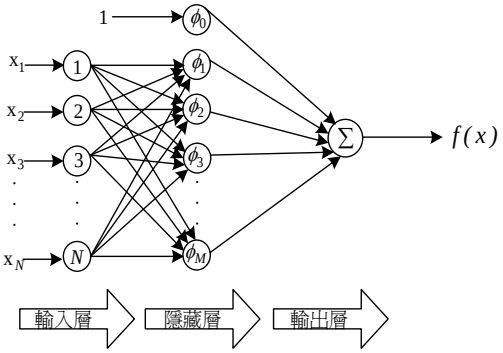

圖 4 幅狀基底函數網路架構

將各隱藏層中的輸出經加權後傳至輸出層，即可求得網路輸出如(2)式：

$$f(x)=\sum_{j=1}^{M} w_j \cdot \phi(x-c_j)+w_0 \qquad (2)$$

式中 $f(x)$ 為輸出層的輸出值，$\phi(x-c_j)$ 表幅狀基底函數，w_j 為隱藏層的第 j 個神經元至輸出層的權重值。在高維度空間的範例分類

(pattern-classification)問題，比低維度空間更符合線性分離趨勢，因此我們經常會建構高維度的隱藏空間的 RBF，即隱藏層神經元個數較多；另一方面，由於隱藏層神經元的個數，會直接影響網路建構輸入與輸出間映射關係的能力，越高維度的隱藏空間將可帶來越精確的近似估計值。

當考慮到類神經網路的容錯能力時，我們必須知道雜訊對於執行任務之網路效能的影響程度。而權重值的大小與 MWN 是有關聯性的，因此，很直接的方式就是控制權重值達到收斂效果。為了評估錯誤中已存在的逼近誤差，我們必須了解錯誤的規律性及在錯誤模式下所考慮的工作參數變化，如圖 5 所示。當一個神經元故障時，

$$\tilde{w}_i = \hat{w}_i + s_i \tag{3}$$

$$s_i = b_i \hat{w}_i \tag{4}$$

式中的 $\hat{w}_i$ 是神經元原始的權重值，s_i 是加上容錯參數的錯誤值，則 b_i 是一個常態分布的隨機變數，且 $E[b_i]=0$ 和 $E[b_i^2]=S_b$，其中 S_b 是 MWN 中常態隨機變數 b_i 的變異數。

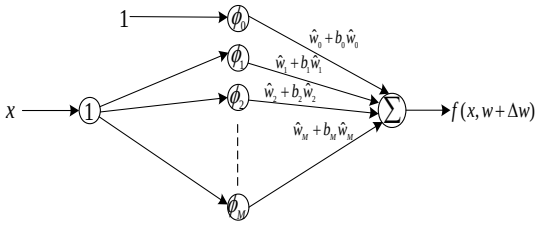

圖 5 加乘性權重雜訊之 RBF

因此在圖 6 中可看出當 $\hat{w}_2 > \hat{w}_1$ 時，錯誤值 s_2 的變異數相對 s_1 來的大。

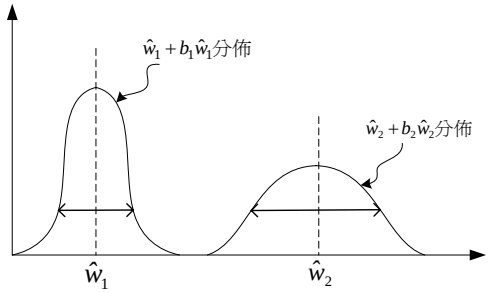

圖 6 加乘性權重的雜訊分布圖

4. 學習演算法

4.1 權重衰退遞減學習演算法[6]

在數值模擬中，權重衰退能改善 FNN 的歸納性效果。權重衰變對線性網路有兩種影響。首先，它能夠抑制對於權重向量不適合的組成元素，以解決學習問題。第二，如果權重大小選擇是正確的，權重衰退能抑制某些靜態噪聲指標的影響，便能提高歸納的能力[7]。權重衰退是一個正規化的方法，顧名思義，就是衰減大量的權重數值。權重衰退會讓微不足道的權重值收斂至零。這也造就一個網路架構上可減少一些自由參數的設置，理論上應該也會得到較佳的歸納法則。在本文的環境中假設有 N 個輸入 x 和輸出 y，

$$Data=\{(x_1,y_1),(x_2,y_2)\cdots\cdots(x_N,y_N)\} \tag{5}$$

目標函數(objective function) $c(w)$，

$$c(w)=\frac{1}{N}\sum_{k=1}^{N}(y_k - w^T\phi(x_k))^2 + \lambda w^T w \tag{6}$$

式中 y_k 為輸出值、$\phi(x_k)$ 為幅狀基底函數、λ 為學習速率。假設隱藏層中神經元為 ϕ_0、ϕ_1、ϕ_2 共三個，

$$\begin{aligned} c(w) &= \frac{1}{N}\sum_{k=1}^{N}(y_k - \sum_{i=0}^{2} w_i\phi_i(x_k))^2 + \lambda\sum_{i=0}^{2} w_i{}^2 \\ &= \frac{1}{N}\sum_{k=1}^{N}(y_k - \sum_{i=0}^{2} w_i\phi_i(x_k))^2 + \lambda(w_0{}^2 + w_1{}^2 + w_2{}^2) \end{aligned} \tag{7}$$

而經由推導後，權重向量則表示為

$$\begin{bmatrix} \frac{1}{N}\sum_{k=1}^{N}(y_k\phi_0(x_k)) \\ \frac{1}{N}\sum_{k=1}^{N}(y_k\phi_1(x_k)) \\ \frac{1}{N}\sum_{k=1}^{N}(y_k\phi_2(x_k)) \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum_{k=1}^{N}\phi_0(x_k)\phi_0(x_k)+\lambda & \frac{1}{N}\sum_{k=1}^{N}\phi_1(x_k)\phi_0(x_k) & \frac{1}{N}\sum_{k=1}^{N}\phi_2(x_k)\phi_0(x_k) \\ \frac{1}{N}\sum_{k=1}^{N}\phi_0(x_k)\phi_1(x_k) & \frac{1}{N}\sum_{k=1}^{N}\phi_1(x_k)\phi_1(x_k)+\lambda & \frac{1}{N}\sum_{k=1}^{N}\phi_2(x_k)\phi_1(x_k) \\ \frac{1}{N}\sum_{k=1}^{N}\phi_0(x_k)\phi_2(x_k) & \frac{1}{N}\sum_{k=1}^{N}\phi_1(x_k)\phi_2(x_k) & \frac{1}{N}\sum_{k=1}^{N}\phi_2(x_k)\phi_2(x_k)+\lambda \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}, \tag{8}$$

最後所求之權重值 w 為

$$w=\left[\frac{1}{N}\sum_{k=1}^{N}\phi(x_k)\phi^T(x_k)+\lambda \mathrm{I}_{\mathrm{M}\times\mathrm{M}}\right]^{-1}\left[\frac{1}{N}\sum_{k=1}^{N}y_k\phi(x_k)\right] \tag{9}$$

4.2 明確正規化學習演算法[8]

當學習演算法應用於多層感知器(Multilayer Perceptron, MLP)結構，也可用在不同解決方案的權重值，這包含了參數的適用規則或初始條件均發生變化。而在倒傳遞(Backpropagation, BP)演算法中最大限度的容錯率方案也曾被提出來。演算法裡增添一個新的項目在 BP 學習規範裡，包含均方誤差(Mean Square Error, MSE)遞減用於權重偏差問題而

其目標就是能夠達成最小化遞減。ERL 的目標函數[8,9]，

$$c(w)=\frac{1}{N}\sum_{k=1}^{N}(y_k-w^T\phi(x_k))^2+S_b w^T G w \qquad (10)$$

S_b 就是 MWN 中常態隨機變數 b_i 的變異數，而 **G** 就是新增的項目，

$$\mathbf{G}=\frac{1}{N}\sum_{k=1}^{N}\begin{bmatrix}\phi_0^2(x_k) & & \cdots & 0\\ & \phi_1^2(x_k) & & \\ \vdots & & \ddots & \vdots\\ 0 & & \cdots & \phi_M^2(x_k)\end{bmatrix} \qquad (11)$$

權重值 w 如(12)式：

$$w=\left(\frac{1}{N}\sum_{k=1}^{N}\phi^T(x_k)\phi(x_k)+S_b G\right)^{-1}\left(\frac{1}{N}\sum_{k=1}^{N}y_k\phi(x_k)\right) \qquad (12)$$

5. WDL=ERL(當 $w_o=0$)

要訓練一個 RBF 面對 MWN，我們可以利用 WDL 和 ERL。由於 WDL 和 ERL 為兩種截然不同的方法。WDL 是一種訓練 NN 去達到更好的歸納(regularization)演算法，而 ERL 則是針對 MWN。此章節主要說明一維與二維向量如何達成 WDL=ERL 的容錯效果。

5.1 一維問題

針對一維向量部分我們擬出以下研究問題：當權重值受到不同幅度的擺動，所預測的誤差值必須要變小。而針對 MWN 部分，我們在 WDL 與 ERL 得到之學習速率分別為

$$\begin{bmatrix}\lambda & \cdots & 0\\ & \lambda & \\ \vdots & \ddots & \vdots\\ 0 & \cdots & \lambda\end{bmatrix} \qquad (13)$$

與

$$\begin{bmatrix}S_b\frac{1}{N}\sum_{k=1}^{N}\phi_0^2(x_k) & & \cdots & 0\\ & S_b\frac{1}{N}\sum_{k=1}^{N}\phi_1^2(x_k) & & \\ \vdots & & \ddots & \vdots\\ 0 & & \cdots & S_b\frac{1}{N}\sum_{k=1}^{N}\phi_M^2(x_k)\end{bmatrix} \qquad (14)$$

假設當 $N\to\infty$ 時，在 $x\in[-L,L]$ 和 $p(x)=\frac{1}{2L}$ 的情況下，

$$\frac{1}{N}\sum_{k=1}^{N}\phi_i^2(x_k)\approx\int_{-L}^{L}\phi_i^2(x)p(x)dx \qquad (15)$$

$$\int_{-L}^{L}\phi_i^2(x)p(x)dx\approx\frac{1}{2L}\int_{-\infty}^{\infty}\phi_i^2(x)dx = \frac{1}{2L}\int exp(-\frac{2(x-c_i)^2}{\sigma})dx \qquad (16)$$

形成(17)式：

$$\int\phi_1^2(x)dx=\int\phi_2^2(x)dx=\cdots=\int\phi_M^2(x)dx \qquad (17)$$

而在這種情況下，我們假使把 w_0 取消也就是 $w_0=0$，此時所求出的(18)式，使用在 WDL 與 ERL 演算法上其容錯能力是相同的。

$$\lambda=\frac{S_b}{2L}\sqrt{\frac{\pi\sigma}{2}} \qquad (18)$$

5.2 二維問題

另外，對於二維的輸入向量部份，我們以符號函數(Signum function)轉換成輸出值 y。繼而從判斷式得知其規則。此外，在本文中我們將二維輸入 x_1 與 x_2 經過 Sign 函數轉換只判斷為 0 或 1。

$$f(x_1,x_2)=\begin{cases}1 & if\ \ x_1\cdot x_2\ge 0\\ 0 & if\ \ x_1\cdot x_2<0\end{cases} \qquad (19)$$

若以立體空間來表示，當輸入 x_1 與 x_2 經(19)式所定義之規則後可得 y，此時分成四個區塊，第一與第三象限為 0，而第二與第四象限則為 1。

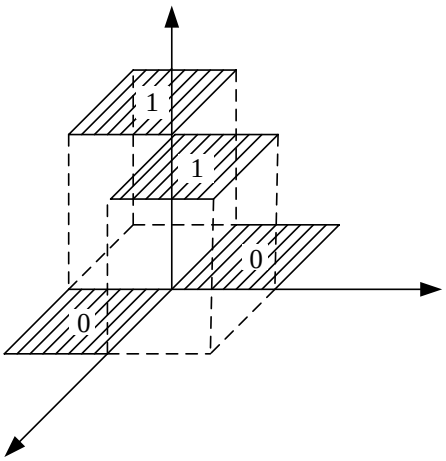

圖 7 二維向量經符號函數轉換之 3D 空間示意圖

當二維輸入向量由 x_1 與 x_2 所組成，且 $N \to \infty$ 時，假設情況如(20)式所示，這裡我們以 $\left(x_1^k, x_2^k\right)$ 代表第 k 個輸入樣本(input pattern).

$$\begin{aligned} &If,\ \frac{S_b}{N}\sum_{k=1}^{N}\phi_i^2\left(x_1^k,x_2^k\right)=\frac{S_b}{N}\sum_{k=1}^{N}\phi_j^2\left(x_1^k,x_2^k\right)=\rho, \forall i \neq j \\ &Then,\ \lambda=\rho \end{aligned} \tag{20}$$

此時，將 $\phi_i^2\left(x_1^k, x_2^k\right)$ 作積分，

$$\begin{aligned} &\frac{1}{N}\sum_{k=1}^{N}\phi_i^2\left(x_1^k,x_2^k\right) \\ &\approx \int_{-1}^{1}\int_{-1}^{1}\phi_i^2\left(x_1,x_2\right)p\left(x_1,x_2\right)dx_1dx_2 \\ &\approx \frac{1}{4}\int_{-1}^{1}\int_{-1}^{1}\phi_i^2\left(x_1,x_2\right)dx_1dx_2 \end{aligned} \tag{21}$$

$$\begin{aligned} &\because \int\phi_i^2(x)dx=\int exp\left(-\frac{2}{s}(x-c)^T(x-c)\right)dx \\ &\int exp\left(-\frac{1}{2}(x-\mu)^T\sum{}^{-1}(x-\mu)\right)dx \\ &=(2\pi)^{N_x/2}|\Sigma|^{1/2} \end{aligned} \tag{22}$$

假設維度為 2($N_x = 2$)和 s 為神經元的寬度時，

$$\begin{aligned} &2\Sigma=\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}\Big/2 \\ &\therefore|\Sigma|=\frac{1}{4}s^2 \end{aligned} \tag{23}$$

最後，由(24)式的結果用在二維向量的容錯部分與一維有相同的能力。

$$\lambda=\frac{\pi}{2}s^2s_b \tag{24}$$

當 $w_0 \neq 0$，則 WDL 與 ERL 便會有所差別。而在上述的假設情況下考慮到有兩種情形產生，當 $w_0 = 0$、$N \to \infty$ 或 $w_0 = \mathbf{1}$、$N \to \infty$ 時，在同時對 RBF 的容錯能力是否有所差別，因此我們將在下一段帶出我們所作的模擬結果。

6. 模擬結果(當 $w_0 = 1$)

為驗證 WDL 與 ERL 在這兩種學習演算法訓練過程中所得到的容錯能力是很相近的，我們應用 MATLAB 程式語言做電腦模擬，把模擬數據分為 training data 和 testing data。在本章節分別對一維及二維做出 RBF 對抗 MWN 在 WDL 與 ERL 兩種演算法上的容錯效果實驗。

6.1 Hermite Function

在本文的模擬實驗中，我們分成三個步驟來說明:首先在 training 部分使用 WDL 來訓練 RBF 模型之後會獲得 w 權重值；第二，我們做的結果是取 testing 的數據針對每個 λ 所對應的 S_b 值則會得到其錯誤值；第三，ERL 是針對 MWN，所以這部分是取 WDL 與 ERL 作出容錯效能的比較。在圖 8 中，所指的是在 RBF 訓練過程中所得到 w，但是考慮 MWN 的關係，我們對每個 S_b (0, 0.01, 0.02, …, 0.49, 0.5) 分別在 test 和 train 各做 100(run)次實驗。並且把各個 λ (0.0001, 0.0002, …, 0.0399, 0.04)相對於每條曲線 S_b 做出其錯誤值的比較。之後依上述的實驗(experiment)過程同樣重複做 200 次。

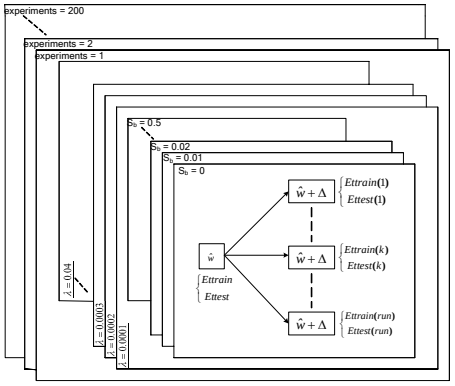

圖 8 模擬的實驗步驟

在整個模擬結果中，為了取樣更多的樣本數以達到更準確的數據，因此在圖 9 中 λ 設定 0.001, 0.002, …, 0.009, 0.01，S_b 設定 0, 0.01, 0.02, …, 0.49, 0.5。而圖中每一條曲線代表的是 S_b，橫軸是 λ，縱軸是錯誤值，這裡所呈現是在不同 λ 值下，每一條 S_b 的錯誤值也相對不一樣。還有在同一條的 S_b 曲線中，其錯誤值會逐漸下降，然後 λ 到一定的數值之後其錯誤值會慢慢的提高。

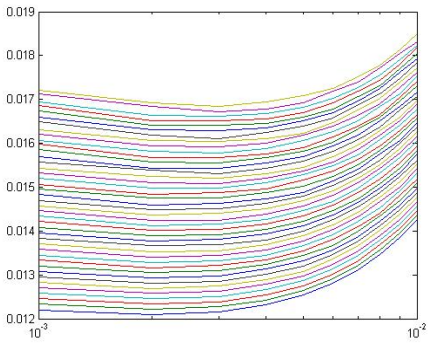

圖 9 測試資料的錯誤值曲線圖

本文對於一維向量的實驗過程所使用的是 WDL，而我們假設當 $\lambda = \frac{S_b}{2L}\sqrt{\frac{\pi\sigma}{2}}$ 的情況下，則會呈現出 WDL 與 ERL 是相同的演算法，如圖 10 所示其容錯效果是很貼近的。圖中Δ代表的是 WDL，□代表 ERL，X 與 Y 座標分別為 S_b 和錯誤值。

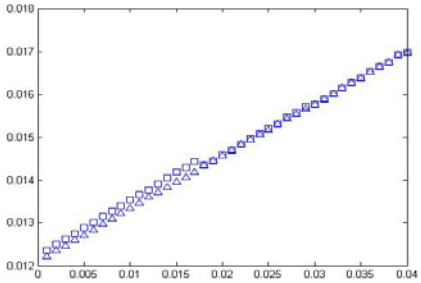

圖 10 演算法在容錯率上的比較圖

6.2 Generalized XOR

在二維的實驗結果，我們依據符號函數的規則判斷輸入向量。此方法的概念經由圖 11 中可得知，E1 與 E2 的組合會直接影響到 E3 的輸出值。

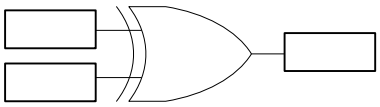

圖 11 Generalization XOR gate

$$F(x1,x2) = 1 \quad \text{if } \text{sign}(x1*x2) \geq 0 \qquad (25)$$

$$F(x1,x2) = 0 \quad \text{if } \text{sign}(x1*x2) < 0 \qquad (26)$$

在二維部分為了取得更詳細的實驗數據，分別對神經元數(*M*)、實驗次數(experiment)、回合數(run)、神經元寬度(*s*)設定不同的參數值，以比較彼此之間的關連性。

表 1 實驗參數設定值

M=100	*M*=150
experiment=20	experiment=20
run=100	run=100
s=0.5、0.6、0.7、0.8、0.9	*s*=0.5、0.6、0.7、0.8、0.9

從二維部份的研究實驗數據顯示，將輸入向量經由符號函數轉換後其容錯能力與一維有同等效果。由於 RBF 類神經網路是以函數逼近的方式來建構網路，因此，訓練整個網路通常包含大量的連結權重需要調整，尤其在面對大量神經元數或是大量訓練資料。另外在訓練過程中，目標函數有時會落入誤差函數的局部的最小值(local minimum)。微積分概念中的微分，其中基於多項式不斷微分概念的泰勒定理(Taylor's theorem)，更成為函數逼近論的基礎。而藉由表 1 實驗參數設定的不同，將 WDL 與 ERL 的容錯效果分別作出圖 12 至圖 21。

表 2 *M*=100 之參數設定

	exp.	run	*s*
圖 12	20	10	0.5
圖 13	20	10	0.6
圖 14	20	10	0.7
圖 15	20	10	0.8
圖 16	20	10	0.9

表 3 *M*=150 之參數設定

	exp.	run	*s*
圖 17	20	10	0.5
圖 18	20	10	0.6
圖 19	20	10	0.7
圖 20	20	10	0.8
圖 21	20	10	0.9

神經元寬度 *s* 的設定大小亦關係著 WDL 與 ERL 的容錯能力好壞。圖中Δ代表的是 WDL，□代表 ERL，X 與 Y 座標分別為 S_b 和錯誤值。

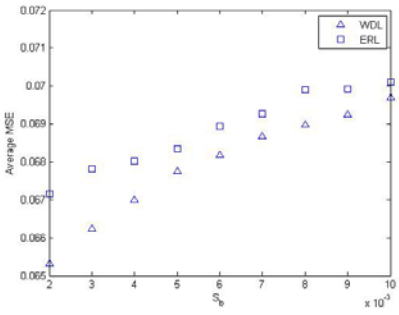

圖 12 WDL&ERL 容錯效能比較圖

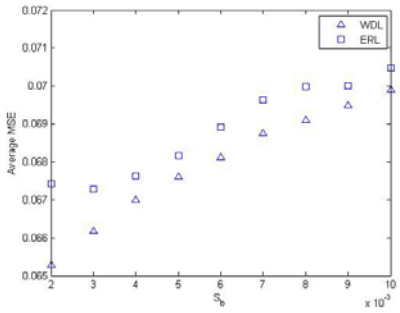

圖 13 WDL&ERL 容錯效能比較圖

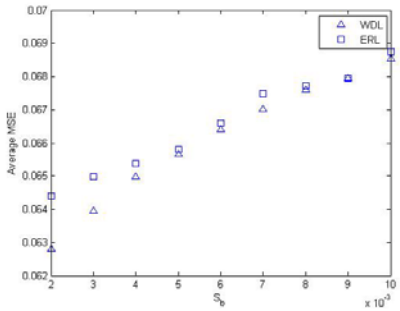

圖 14 WDL&ERL 容錯效能比較圖

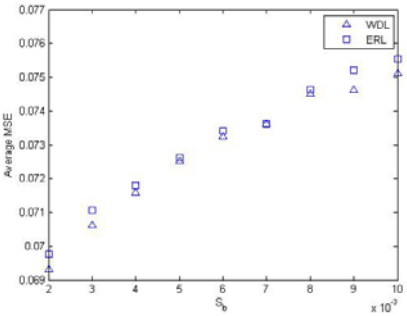

圖 15 WDL&ERL 容錯效能比較圖

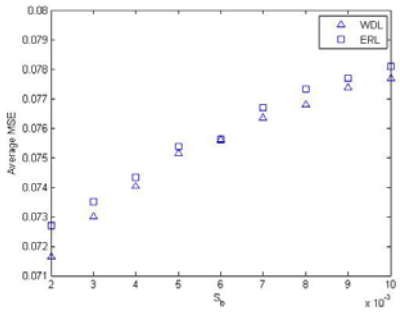

圖 16 WDL&ERL 容錯效能比較圖

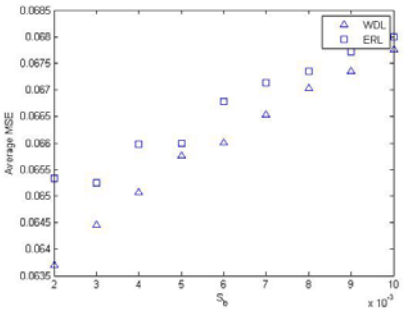

圖 17 WDL&ERL 容錯效能比較圖

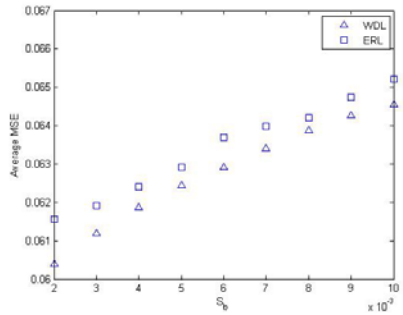

圖 18 WDL&ERL 容錯效能比較圖

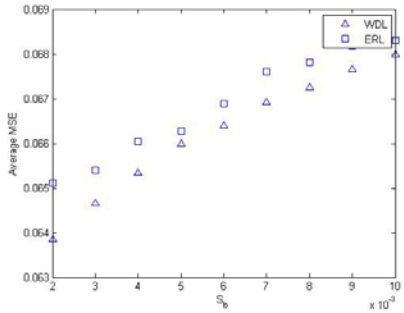

圖 19 WDL&ERL 容錯效能比較圖

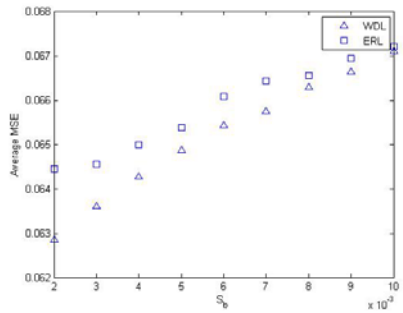

圖 20 WDL&ERL 容錯效能比較圖

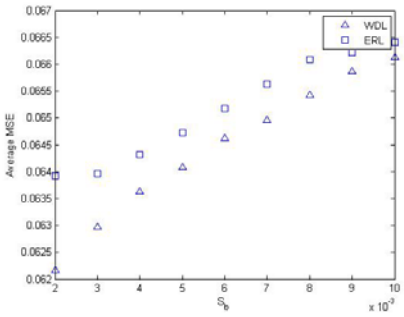

圖 21 WDL&ERL 容錯效能比較圖

由表 2 及表 3 的參數設定於 M=100 且 s=0.7、0.8、0.9 時其容錯效果是很相近的；另外 M=150、s=0.6 與 0.7 的容錯效果也不錯。

7. 結論

由本篇所做的實驗結果顯示在一維與二維向量的某特定條件下，訓練 RBF 對抗 MWN 使用 WDL 與 ERL 兩種學習演算法會有同樣的容錯能力。此外，得知在二維部份的神經元寬度 s 參數設定攸關容錯效能的好壞程度。未來會以不同類神經網路模型來訓練容錯率方面的問題，作為彼此之間的效果分析與研究。

參考文獻

[1] S. Haykin, *Neural networks*, "A Comprehensive foundation," *Macmillan College Publishing Company Inc*., 1994.

[2] A.F. Murray, and P.J. Edwards, "Enhanced MLP performance and fault tolerance resulting from synaptic weight noise during training," *IEEE Transactions on Neural Networks*, vol. 5, no. 5, pp. 792-802, 1994.

[3] O. Fontenla-Romero, *et al.*, "A measure of fault tolerance for functional networks," *Neurocomputing*, vol. 62, pp. 327-347, 2004.

[4] J. Sum, "On a multiple nodes fault tolerant training for RBF: Objective function, sensitivity analysis and relation to generalization," *Proceedings of TAAI'05*, Tainan, ROC, 2005.

[5] C.S. Leung and J. Sum, "A fault tolerant regularizer for RBF networks," accepted for publication in *IEEE Transactions on Neural Networks*.

[6] J.E. Moody, "Note on generalization, regularization, and architecture selection in nonlinear learning systems," *First IEEE-SP Workshop on Neural Networks for Signal Processing*, 1991.

[7] C. Bishop, "*Neural Networks for Pattern Recognition*," *Oxford University Press*, 1995.

[8] J.L. Bernier *et al.*, "Obtaining fault tolerant multilayer perceptrons using an explicit regularization," *Neural Processing Letters*, vol. 12, pp. 107-113, 2000.

[9] J. Sum, *et al.*, "On objective function, regularizer and prediction error of a learning algorithm for dealing with multiplicative weight noise," in submission to *IEEE Transactions on Neural Networks.*

[10] J. Sum, *et al.*, "On prediction error of a fault tolerant neural network," *Neural Information Processing*, Springer LNCS no. 4232, pp. 521-528, 2006. Spring-Verlag Berlin Heidelberg.

[11] C.S. Leung and P.f. Sum, "Analysis on Bidirectional Associative Memories with multiplicative weight noise," *Neural Information Processing*, Springer LNCS, 2007. Spring-Verlag Berlin Heidelberg.

[12] J. Sum, "Towards an objective function based framework for fault tolerant learning," in *Proceedings of TAAI'2007*.

[13] W.H. Luo, J. Sum, Y.F. Huang, "Empirical study of a fault tolerant RBF," in *Proceedings of TAAI'2007*.

[14] J. Sum, C.S. Leung, L.P. Hsu, Y.F. Huang, "An objective function for single node fault RBF learning," in *Proceedings of TAAI'2007*.

[15] J. Sum, C.S. Leung and L. Hsu, "Fault tolerant learning using Kullback-Leibler Divergence," in *Proceedings of TENCON'2007* Taipei.