

積體電路良率管理之良率模式構建 -類神經網路之應用

Constructing Yield Prediction Model for IC Yield Management : Applying Artificial Neural Networks

¹謝昆霖 ²唐麗英

¹南華大學資訊管理學系

²交通大學工業工程與管理學系

摘 要

良率(yield)是半導體積體電路業者進行獲利評估的一個重要指標，為了能快速、有效地且正確地管理與提升製程的良率，許多有關於良率管理(yield management)的研究相繼提出。良率管理的領域非常廣泛，凡致力於提升良率的各種方法或是研究，都包含在良率管理之中；而良率管理中，對於提供預測功能的良率模式(yield model)之研究則是一值得投入研究的課題。

傳統良率模式因只考慮缺陷數目(defect count)與良率間的關係，所以所推導出來的良率模式其準確度並不好；近年來一些經過修正的良率模式持續被發表出來，但卻因過於複雜而導致使用上容易受到限制。本研究所提出的良率模式，在建構上的考量是將晶圓(wafer)上的缺陷來源(defect source)、缺陷數目以及缺陷群聚(defect cluster)等一併納入分析，並利用類神經網路對於非線性結構問題的良好處理能力，建立一適切的良率預測模型，以有效地解決傳統良率模式精確度較差的問題以及修正之良率模式在使用上過於複雜的問題。

本研究並實際收集新竹科學園區某 IC 製造廠的實際晶圓資料來構建適切的良率模式，從比較結果中可以發現本研究所構建的良率模式相較於其它良率模式，應用於半導體積體電路的良率預測時可以獲得較佳的成效。

關鍵詞：良率模式、良率管理、缺陷群聚、類神經網路

壹、 緒論

良率(yield)是積體電路(integrated circuit, IC)業者獲利的一個重要指標，所以在 IC 產品設計、開發以及生產的過程中，正確且嚴謹的良率管理(yield management)是非常重要的。Kang[4]認為良率管理包括了良率及生產資料的分析、良率模式的發展、缺陷監控(defect monitor)以及良率與成本的相關研究等。El-Karehp[3]更認為良率管理中，最為重要的就是良率模式。

自 1960 年以來，就不斷地有學者研究 IC 的良率模式(yield model)。依據良率模式推導的結果，業者可以利用良率模式來預估產品之良率、監控生產線、決定新產品推出的適當時機以及進行更高良率的電路設計。El-Karehp 認為良率模式是良率管理領域中最重要的研究項目，原因在於良率模式能預測在製品的最終良率，它幫助製造商在 IC 產品的生產過程中可以根據缺陷的情形預估產品最終的良率，如此便可以在生產過程中，採取一些改善或修正的行動來提升產品的良率。這些行動包括了晶片(chip)設計的變更、製程條件的改善或是調整修復電路(redundancy circuit)的數量等。除此之外，Albert 與 Ferris-Prabhu[1]更提出了利用良率模式進行新產品良率的預測，更提高了良率模式的能力。

良率模式的發展，除基本的卜瓦松(Poisson)模式外，尚有 Murphy 模式、Seeds 模式、Okabe 模式、Half-Gaussian 模式以及目前較為使用的負二項(Negative Binomial)模式[10]，然而傳統的良率模式僅考慮缺陷數與良率的關係，並沒有考慮其它可能影響

良率的因素，因此造成傳統的良率模式不是在預測上與實際的良率差異頗大，就是在使用上過於複雜。

為了使良率模式能夠更加地實用，本研究將利用類神經網路具有高度學習能力、記憶能力、歸納能力以及容錯能力等性質，將積體電路晶圓(wafer)上的缺陷以類神經網路中的模糊自調共鳴理論(Fuzzy Adaptive Resonance Theory, Fuzzy ART)的方式予以分類，並利用類神經網路中之倒傳遞(Back Propagation Neural Network, BPNN)演算法構建出一個較精準且使用簡便的良率模式。

本研究共分為五個章節，第一章為緒論，說明本研究之動機、目的，以及研究架構等；第二章為相關文獻的探討，說明良率的定義以及重要的 IC 良率模式之簡介，並探討可能造成良率損失的相關因素；良率模式的構建流程則於第三章中加以說明，敘述如何快速地構建一個合理且準確的良率模式；第四章為實例證明，將實際 IC 生產線上所收集的晶圓資料，利用本研究提出的方法來構建 IC 良率模式，並與其它的良率模式比較，以說明本研究所提出的良率模式建構方法之優越性；相關的結論則與以分析歸納整理於第五章。

貳、 相關文獻探討

本節將介紹一些與良率及良率模式相關的知識及文獻。

2.1 積體電路的良率

積體電路的良率一般定義為「通過最終規格測試的晶方(die)數相對於所有經過測試

晶方數之比率」，而 IC 的良率則可以大略的分為三個部分[3]，如下所示：

$$Y = Y_p \times Y_T \times Y_A$$

其中， Y_p 表示製程良率(process yield)；

Y_T 表示測試良率(test yield)；

Y_A 表示組裝良率(assembly yield)；

在良率模式的研究方面，皆假設製程已呈穩定狀態，不影響良率損失，所以將重點放在組裝良率與測試良率方面。

除此之外，在進行良率模式的研究當中，必須注意到良率受到隨機缺陷(random defect)或是系統缺陷(systematic defect)影響的情況。Mirza[2]說明良率可以分為系統良率(systematic yield, Y_s)以及隨機良率(random yield, Y_r)兩部分，即

$$Y = Y_s \times Y_r$$

其中， Y_s 通常是由缺陷群聚所造成的良率，而 Y_r 則是由隨機缺陷所造成的良率。

因為收集產品的最終良率非常不易，所以在本研究中，擬以估計良率(estimated yield)來代表產品最終良率；而所謂晶圓的估計良率，就是指晶圓上完全沒有缺陷的晶片佔全部晶片的百分比。

2.2 積體電路的良率模式

1960 年來，在良率模式的研究上，有許多種以不同分配為假設的模式被提出，目前最常見的為卜瓦松模式以及負二項良率模式，以下分別介紹這兩個良率模式。

1. 卜瓦松模式

根據缺陷數為卜瓦松機率分配函數的假設所發展出的良率模式，是早期用於預測 IC 良率的良率模式。在此模式中，假設缺陷發生在晶圓上的位置是呈現隨機分佈的，亦即

假設缺陷落在任何一片晶片上的機率是相同的，因此其缺陷密度(defect density)為一個固定的常數。在此假設之下，一個晶片上有 k 個缺陷的機率為：

$$P(k) = \frac{e^{-I_0} I_0^k}{k!} \quad k = 0, 1, 2, \dots$$

其中， I_0 表晶片之平均缺陷數， A 表示晶片面積， D_0 表示缺陷密度， $I_0 = AD_0$ 。

由此可導出卜瓦松良率模式為

$$Y = P(k = 0) = e^{-I_0} = e^{-AD_0}$$

當晶片面積小於 0.25cm^2 時，卜瓦松良率模式尚可以合理地預測良率值，但隨著晶片面積不斷地增大，卜瓦松模式則會出現低估良率的現象[9]。

2. 負二項(negative binomial)良率模式

由 Stapper[10]所提出的負二項良率模式是屬於複合卜瓦松良率模式(compound Poisson yield model)[6]的一種，其良率模式為：

$$Y = \frac{1}{(1 + D_0 A / a)^a}$$

負二項分配具有兩個參數 a 及 D_0 ，其中 a 參數使得此模式具有彈性，經由 a 參數的調整，負二項良率模式可以轉換成其它不同的良率模式。此 a 值被稱為群聚參數(cluster parameter)，它表示了缺陷在指定面積上的群聚程度，此值越小表示群聚的狀況越嚴重。

a 值除了以上的表示方法外，亦可以表示如下：

$$a = \frac{\bar{I}^2}{s^2 - \bar{I}} \dots \dots \dots (1)$$

其中 \bar{I} 表每個晶片的平均缺陷數， s^2 表晶片上缺陷之變異數。

然而 Cunningham[5]指出，利用(1)式求取 a 值時， a 值會顯得過於散亂，同時亦可能有負值的出現，這些都會造成分析時的不便，所以他建議以下列公式來計算 a 值：

$$a = (\bar{I}/s)_{avg}^2 \frac{1}{1 - \bar{I}_{avg}/s^2}$$

其中 \bar{I}_{avg} 表示一片晶圓上每片晶片之平均缺陷數。

Stapper[11]亦認為不同大小的缺陷群聚現象會對良率造成不同的影響，為了讓負二項良率模式可以更準確的預估 IC 良率，因此 Stapper 將缺陷群聚分為大於晶片面積的缺陷群聚及小於晶片面積的缺陷群聚，同時 Stapper[12]認為這兩種缺陷群聚現象對於良率所造成的影響是不同的，因此應考慮不同種類的缺陷群聚對良率的影響。

3. 利用迴歸分析所構建的良率模式

Collica[8]將晶片上的失效型態分為 11 種，並結合羅吉斯迴歸分析(logistic regression)的方法，建構出靜態隨機存取記憶體(static random access memory, SRAM)的良率模型。此模型不但可以有效地預估良率，同時可以找出何種的失效型態是導致良率下降的主因。

4. 修正良率模式(modified yield model)

趙豐昌 [14]則以模糊自調共鳴理論(Fuzzy adaptive resonance theory, Fuzzy ART)演算法修正了晶圓上缺陷群聚的現象，並將修正後的缺陷數代入卜瓦松良率模式進行良率預測，其建構程序如下：

步驟一：利用 KLA 晶圓偵測系統得到

每一片晶圓的缺陷總數以及缺陷座標。

步驟二：對晶圓上的缺陷作離群值分析，找出可能的離群值，並追查其發生的原因，以決定是否刪除這些離群值。

步驟三：判斷每片晶圓上的缺陷是否出現群聚現象。若缺陷分佈不為卜瓦松分佈，則進行步驟四；否則直接進行步驟五。

步驟四：利用 Fuzzy ART 進行缺陷分析。由晶圓上的缺陷座標中去學習缺陷的群聚規則，將所有的缺陷作一適當的分群。接著對缺陷數做一修正，將每一聚類當成一個新的座標點，而以聚類的中心當成新缺陷的座標。以此方式並配合修正警戒值，直到缺陷分佈符合卜瓦松分配才停止。

步驟五：計算缺陷密度，建立以卜瓦松分配為基礎的良率模式。

步驟六：利用步驟五所得之良率模式來預測良率，並進一步分析良率。

利用此方法可以得到比卜瓦松模式與負二項模式更佳的良率預測值，但是仍有一些缺點：

1. 應用此方法必須先行修正缺陷群聚現象，如此將導致缺陷資料失真，可能無法找出良率損失與缺陷數間較明確的關係。
2. 此方法必須反覆對每一片晶圓進行群聚分類以及測試的工作，建構流

程極為複雜。

3. 利用此方法僅能獲得 IC 產品最終的良率，但是卻無法藉由良率模式獲得其他相關的機率值，例如每片晶圓上， n 個缺陷落在單一晶片上的機率值。因此採用此方法將使得所構建出的良率模式功能受限。

2.3 造成良率損失的因素

除了缺陷的數量外，不同型態與來源的缺陷及缺陷在晶圓上群聚的程度，也是造成良率損失的重要原因，本小節分為兩個部分介紹如何對不同來源的缺陷進行分類，以及如何求取一個具代表性的指標來顯示缺陷群聚的程度。

1. 缺陷來源的分類

許志輝[13]認為晶圓上的缺陷會由許多的原因造成，例如刮傷、有雜質、晶崩、雜質度不高等等，因此他假設晶圓上缺陷群聚形狀，可以幫助我們判斷這些呈現相同群聚形狀的缺陷，是由相同原因造成的。他假設造成晶圓上產生缺陷的情況有四種(隨機、缺陷 1、缺陷 2、缺陷 3)，並利用模糊自我成圖(Fuzzy Self-Organizing Map, FSOM)類神經網路對晶圓上的缺陷群聚進行分類。此研究之方法提供我們利用類神經網路的方法辨識晶圓上缺陷來源的可行性。

2. 群聚嚴重程度之研究 - 方格法

Stapper[11]認為晶圓上的缺陷群聚程度是影響 IC 良率與品質的重要原因，故探討晶圓上的缺陷群聚現象是必須的。探討晶圓上缺陷是否存在群聚現象時，必須判定晶圓片缺陷數是符合何種分配。判定平面空間上點分佈的方法，在空間統計[7]中提到兩種 - 方

格法(Quadrat Method)與距離法(Distance Method)。由於距離法對某些特殊的缺陷分佈，無法偵測出其群聚現象，且在實用上困難性遠較方格法高，故本研究擬採用方格法來描述缺陷在晶圓上的分佈的情形。

方格法最早是應用在動物生態學上，其研究的方法是將所欲研究的區域分成許多大小相等的方塊，然後對每一個方塊內的點數作檢定。透過檢定可以得知在所研究的空間內，點的分佈是隨機或是具有群聚現象的。

由於卜瓦松分配的平均數等於變異數，故變異數與平均數的比值為 1，因此若將每一個方格內的點視為一隨機變數(random variable)，只需檢定其變異數與平均數的比值(Variance-Mean Ratio, V/M)和 1 之間是否有顯著差異，便可以推論點的分佈是否符合隨機現象。Greig-Smith 根據此特性建立了變異數平均數比值的 t 檢定量：

$$t = \frac{(V/M) - 1}{\sqrt{2/(n-1)}}$$

其中 V 是變異數， M 是平均數， n 是方格數。此 t 檢定量的自由度為 $n-1$ ，當 $t > t_{\alpha, n-1}$ 時，則可推論點的分佈有群聚現象。

本研究將藉助變異數平均數比值 (V/M) 這一項指標來代表缺陷分佈的群聚程度，並以晶圓上的晶方數代表分割方格的數目。

參、類神經網路介紹

類神經網路是一種模仿生物神經網路的資訊處理系統，它使用大量簡單的相連處理單元(process element, PE)來模仿生物神經網路系統的能力。PE 從外界環境或是其它 PE 取得資訊後，經過簡單的運算，再將結果輸出到外界環境或是其它 PE。一般而言，一個

類神經網路的優劣與否，取決於訓練樣本或測試樣本經過處理後，產生的誤差均方根 (root mean square error, RMSE) 值之大小，通常 RMSE 較小者為優。本研究中採行了兩種的類神經網路演算法，一為自調共鳴理論演算法，另一則為倒傳遞演算法。本研究使用類神經網路電腦軟體 Neural Professional Plus/II 中的 Fuzzy ART 以及 BPNN 演算法，以下將以 Neural Professional Plus/II 中所提供的這兩個演算法為基礎來簡介這兩種類神經網路演算法。

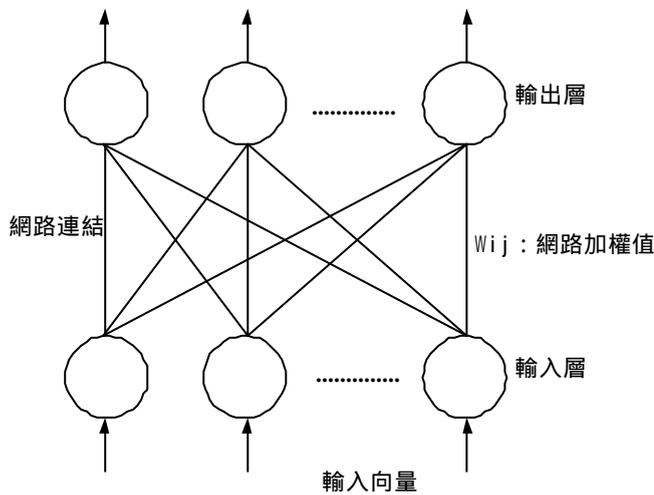
3.1 自調共鳴理論網路

Fuzzy ART 演算法將模糊集合(Fuzzy Set)的觀念帶入傳統的 ART-1 演算法中，如此便可以快速且穩定地處理輸入值為類比或是二元值。

Fuzzy ART 的網路架構分為兩層，即輸入與輸出層。在輸入及輸出層各個 PE 間

都有網路相連。其中輸入層的第 I 個 PE 與輸出層的第 j 個 PE 之連結網路上有一個網路加權值 W_{ij} 。輸入層的 PE 個數代表著輸入向量的維度(dimension)，而輸出層的 PE 個數代表最大可能的聚類個數。Fuzzy ART 的網路架構圖如圖一所示。

利用 Fuzzy ART 時，必須先對輸入向量進行處理，因為 Moore [14] 發現用自調共鳴理論網路來處理類比值時，可能造成類別激增 (category proliferation) 的情形。所以在讀入輸入向量後，必須先對輸入向量進行補數編碼 (complement coding) 以避免類別激增而造成不適當的歸類。所謂的補數編碼是把原先輸入量的維度加倍，而多出的維度內容，則以原先輸入向量各 PE 值的補數代之，例如輸入向量 $I = \{I_1, I_2, \dots, I_M\}$ 經過補碼後的輸入向量則為 $I = \{I_1, I_2, \dots, I_M, 1 - I_1, 1 - I_2, \dots, 1 - I_M\}$ 。



圖一 Fuzzy ART 架構圖

在應用 Fuzzy ART 網路時必須設定三個參數，亦即選擇參數(a)，學習參數(b)，以及警戒參數(r)。其中以警戒參數 r 最為重要； r 值越大，越不容易通過警戒測試，所得的聚類個數就越多，反之亦然。選擇參數 a 的作用是為避免選擇函數值為 1，在設定時，應注意 a 值不宜太大或太小。至於學習速率 b 會直接影響連結網路之加權值的調整， b 值越大，加權調整值的幅度就會越大。

由於 Fuzzy ART 演算法可用來進行聚類分辨的工作，因此在本研究中，將用來作為分析造成缺陷原因之工具。

3.2 倒傳遞演算法

倒傳遞類神經網路架構如圖二所示，包含輸入層、隱藏層與輸出層。輸入層與輸出層的 PE 數目依問題而定，而隱藏層的層數與其 PE 個數則沒有一定的法則可循，通常以試誤法(try and error)決定適當的數目；一般而言，一至二層隱藏層已足夠解決大部分的問題。連接至隱藏層的連結使用的是非線性轉換函數，在倒傳遞類神經網路中最常用的非線性轉換函數為雙彎曲函數(sigmoid function)，其公式為：

$$f(x) = 1/(1 + e^{-x})$$

倒傳遞類神經網路的計算過程分為向前(forward)與向後(backward)兩階段。首先，輸入層 PE 接收到輸入樣本資料，然後直接將資料傳送給隱藏層，隱藏層中的神經元會將輸入資料加權過後之總和，利用轉換函數轉換成激發值(activation value)輸出至輸出層。輸出層的 PE 再以隱藏層輸出之激發值，同樣的計算其加權過後之總和，並利用轉換函數轉換成激發值輸出，最後比較網路輸出值

與標準值的差異，視情形進行連接權重的調整。輸出層的第 k 個 PE 其權重調整公式如下：

$$\Delta W_{ij} = hd_k O_j$$

其中， ΔW_{ij} =神經元 j 與神經元 k 間權重改變量； h =學習速率；

d_k =神經元 k 的差異量； O_j =隱藏層第 j 個神經元之輸出。

隱藏層的第 j 個神經元其權重調整公式如下：

$$\Delta W_{ji} = hd_j O_i$$

其中， ΔW_{ji} =神經元 i 與神經元 j 間權重改變量； h =學習速率；

d_j =神經元 j 的差異量； O_i =隱藏層第 i 個神經元之輸出。

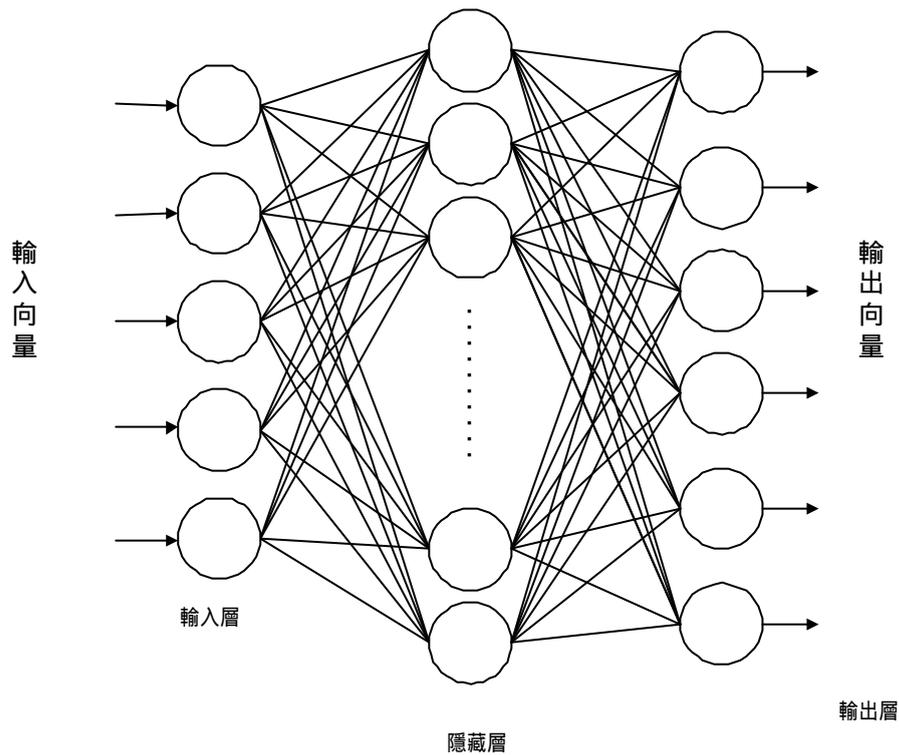
倒傳遞類神經網路學習精確度高，回想速度快，可處理複雜的樣本辨識問題與高度非線性的函數適配(fitting)，利用此特點，可以幫助我們完成 IC 良率預測模式的構建工作。

肆、良率模式的構建流程

本研究首先利用晶圓偵測系統(wafer inspection system，如 KLA2110 系統)來收集實際 IC 生產線上的資料，接著使用類神經網路中 Fuzzy ART 演算法將缺陷群聚予以分類，並計算出各種群聚現象的缺陷數量，同時利用方格法，求得每片晶圓上缺陷群聚的程度。另外，輔以其它整理出的資料作為輸入、輸出值，便可以利用類神經網路中的 BPNN 演算法來構建良率模式，以預測晶片上發生缺陷的機率。這個藉助類神經網路的

作法不但可以快速地構建出預估良率模型，且又容易應用，因此可以有效地解決傳統良率模式構建困難以及使用不易的問題。

本研究所構建之良率模式的流程共分成資料收集與整理、良率模式構建以及模式驗證等步驟，分別以下列幾個小節來說明。



圖二 倒傳遞類神經網路架構圖

4.1 資料收集

本研究將利用 KLA 2110 晶圓偵測系統所提供之晶圓座標圖(wafer map)，求得晶圓上每個缺陷的位置、大小以及型態之後，進行缺陷數目、缺陷分類以及缺陷群聚程度的計算，並計算出估計良率來代表實際良率。

4.2 資料整理

在取得 KLA 機台收集的晶圓資料後需要加以整理，下面三個步驟中，將說明如何將這些資料整理成我們所需要的形式。

1. 缺陷來源的分類

此階段的目的是，要分辨出晶圓上不同缺陷來源之個別缺陷數目。首先決定需要進行訓練的晶圓片數，將前幾片晶圓上的缺陷座標，以 Fuzzy ART 演算法做訓練分群，並調整網路的警戒值參數，在網路穩定後，將其餘的晶圓資料輸入已構建完成之 ART 類神經網路，進行缺陷分類的工作。

本研究在缺陷來源分類的過程中，採用 Fuzzy ART 演算法代替 Fuzzy SOM 演算法 [12] 的原因，在於 ART 演算法可以自動地根據警戒值進行分群的工作，而不需要給定分群的數目，因此可以提供一個較為客觀的分群數，同時能夠較 Fuzzy SOM 更為快速地獲得分群的結果。

最後計算每一片晶圓上不同缺陷來源所包含的缺陷數，用以當成下一步驟倒傳遞演算法的輸入值。

2. 群聚程度的判斷

在 2.5 節中提到，缺陷嚴重的程度可以利用方格法來判斷。假定方格的大小為晶圓上一個晶方的大小，並利用 V/M 的值代表每片晶圓的群聚程度，當 V/M 值越大，可以判定晶圓上的群聚現象越嚴重，因此， V/M 值可作為一個類神經網路倒傳遞演算法的輸入值。

3. 良率與相關缺陷機率值的求取

本研究以不同來源之缺陷數與群聚嚴重的程度作為類神經網路倒傳遞演算法的輸入值，而本節將介紹如何求取倒傳遞類神經網路的輸出值。原始的晶圓資料中，經過整理可以獲得以下數種機率值（以單片晶圓為單位，此機率值即為估計良率值）：

1. 完全沒有缺陷的晶片佔全部晶片的

百分比，以 $P(D=0)$ 代表。

2. 包含一個缺陷點的晶片佔全部晶片的百分比，以 $P(D=1)$ 代表。

3. 包含兩個缺陷點的晶片佔全部晶片的百分比，以 $P(D=2)$ 代表。

4. 包含三個缺陷點的晶片佔全部晶片的百分比，以 $P(D=3)$ 代表。

5. 包含四個缺陷點的晶片佔全部晶片的百分比，以 $P(D=4)$ 代表。

⋮

6. 包含超過 k 個缺陷以上的晶片佔全部晶片的百分比，以 $P(D \geq k)$ 代表。

這 k 項機率值，將在下一階段中輸入倒傳遞類神經網路當作其輸出值。一般的良率模式只預測 IC 的良率，即 $P(D=0)$ ，對於其它的晶片缺陷機率並不加以考慮，但唯有在預測良率之外，同時能夠預測晶片的缺陷機率，才能發揮良率模式除了預測良率之外的其它功能。

4.3 利用類神經網路的倒傳遞演算法構建良率模式

由上述的步驟可獲得晶圓的缺陷群聚程度指標、不同種類缺陷的數目以及缺陷出現的機率值，現分別以這些數值作為輸入及輸出值，利用類神經網路中的倒傳遞演算法進行訓練。圖三表示輸入、輸出值與倒傳遞類神經網路的情形。

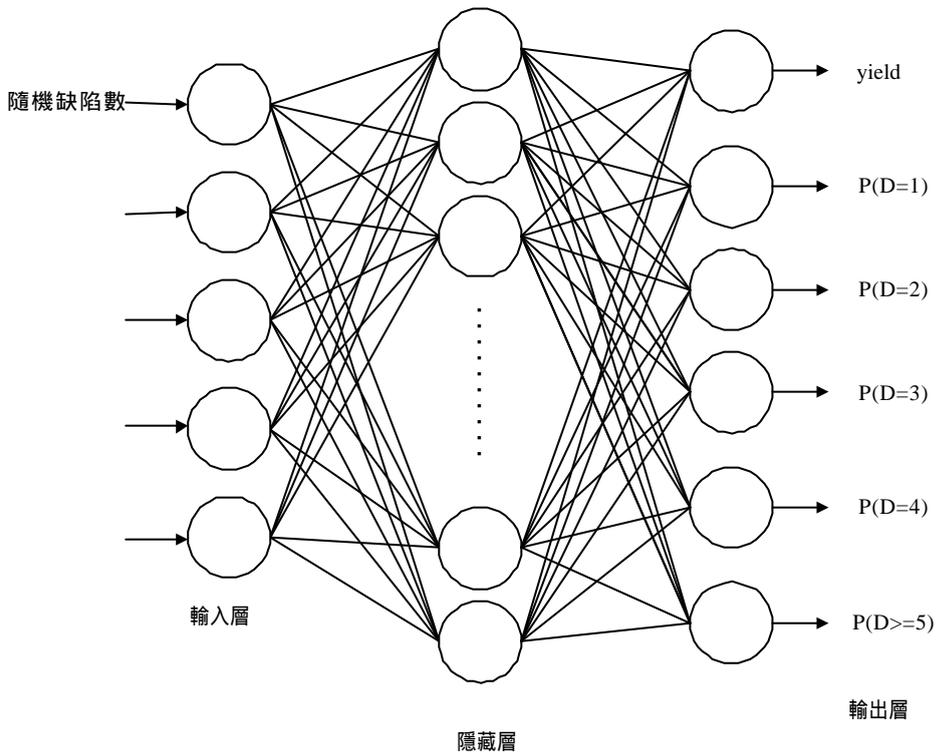
在第參節中，已說明了倒傳遞演算法除了輸出、輸入值的決定外，尚須決定隱藏層的數目、每個隱藏層中的 PE 數目、學習速率與激發函數，本研究將僅選擇一層隱藏層，並利用試誤法進行隱藏層的 PE 數目之判斷。在決定學習速率方面，一開始可以先自訂定一個值，之後再利用試誤法進行調整

並進行訓練，以期求出更佳的结果。至於激發函數，則可以選則雙彎曲函數作為此類神經網路的基礎。

4.4 模式驗證

接著進行良率模式正確性的驗證工作。

將倒傳遞演算法所求出的晶圓良率值及相關機率值與實際的數據進行比較，求出 $P(D=0)$ 、 $P(D=1)$ 、 $P(D=2)$ 、 $P(D=3)$ 、 $P(D=4)$ 至 $P(D \geq k)$ 的 RMSE，以判斷是否接受這個良率模式。



圖三 倒傳遞類神經網路與輸入、輸出值

若此模式的 RMSE 值過大，則必須依序由以下三個準則來判斷是否需要進行調整的工作，並在調整後重新進行模式構建：

1. 在倒傳遞演算法中，所輸入資料的型態、隱藏層的數目、隱藏層 PE 數目以及網路的學習速率是否恰當。
2. 在進行缺陷來源分類時，Fuzzy ART 演算法中警戒值的選取是否恰當。
3. 以 KLA 系統所收集的資料是否具有代表性？是否需要重新收集資料或修正收集的方式？

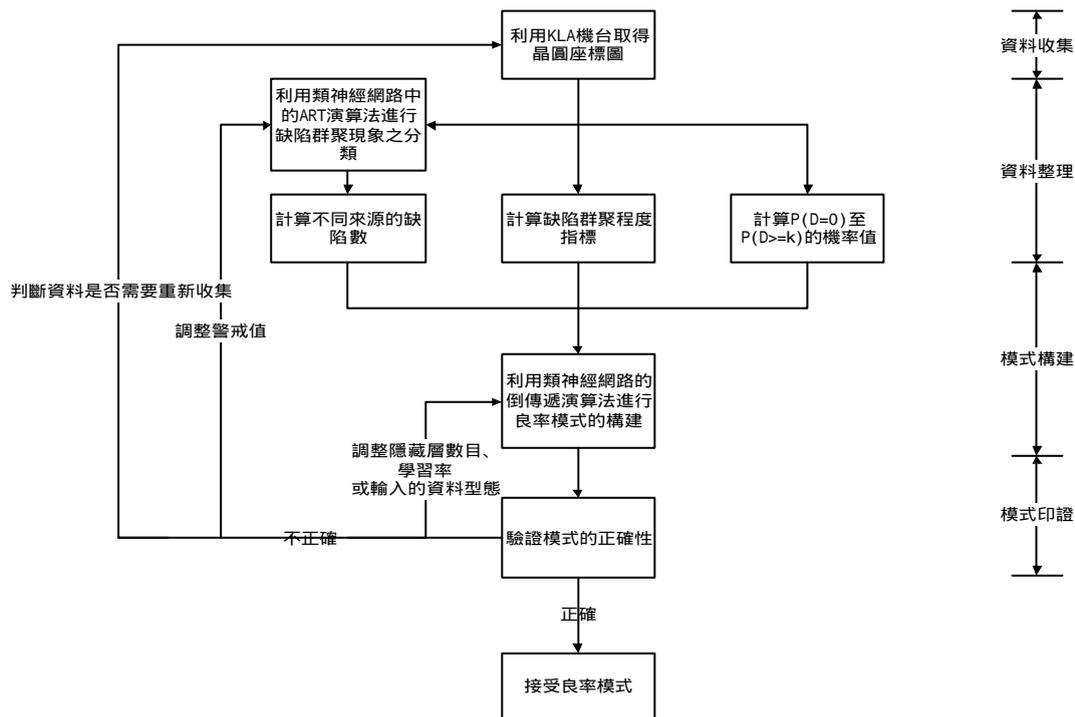
4.5 積體電路良率模式之構建

現將本研究所發展之積體電路良率模式構建流程彙整成以下幾個步驟：

- 步驟一：利用 KLA2110 晶圓偵測系統得到每一片晶圓之缺陷總數及其缺陷座標。
- 步驟二：利用類神經網路 Fuzzy ART 演算法對缺陷予以分類，求出每一類缺陷的數量。
- 步驟三：求取每一片晶圓上的 V/M 值以代表晶圓上的缺陷群聚程度。
- 步驟四：整理 KLA 系統獲得的晶圓資料，求出 $P(D=0)$ 、 $P(D=1)$ 、 $P(D=2)$ 、 $P(D=3)$ 、 $P(D=4)$ 至 $P(D \geq k)$ 。

步驟五：將步驟二及三中求得之晶圓上每一類缺陷數量，及缺陷群聚程度作為輸入值，步驟四中求得的六個機率值作為輸出值，代入倒傳遞演算法中進行訓練，以構建積體電路的良率模式。

步驟六：印證求出良率模式的正確性。利用步驟五的良率模式所預測出之良率及相關機率值與實際值做比較，並計算出 RMSE；若 RMSE 值過高，則需回到建構流程中，對某些項目進行調整，並重新構建良率模式。反之，則接受此良率模式。以上流程可以圖四表示。



圖四 研究方法流程圖

伍、 實例分析

本研究以新竹科學園區某家晶圓廠的六吋晶圓產品經 KLA 晶圓偵測系統所得之晶圓的座標圖作為實例分析的資料。其中，每一片晶圓被切割成 396 個晶方。本研究將這些晶圓資料代入所提的方法進行良率模式的構建及預測，以說明如何應用本研究所提之方法。此外，本研究並將所提之良率模式與其它的良率模式做比較，以證明本研究方法所構建之良率模式的準確性。

5.1 積體電路良率模式之構建

現以上述實例說明如何利用本研究提出的方法來構建積體電路的良率模式：

步驟一：取得晶圓座標圖

從晶圓座標圖得到每片晶圓之缺陷座標及每片晶圓上之總缺陷數。在本研究中，共有 111 片晶圓的相關資料。

步驟二：對於缺陷進行分類

利用 Fuzzy ART 演算法對 111 片晶圓進行缺陷分類的工作。首先先取前 40 片晶圓上缺陷的座標當作訓練值，並選取適當的警戒值(本研究設定之最初警戒值為 0.95)，待進行 100,000 次的訓練後，判斷此 Fuzzy ART 類神經網路已達穩定狀態。再將所有的晶圓資料輸入此訓練完畢之 Fuzzy ART 類神經網路進行分類，獲得 4 種缺陷分類結果，其中

一種為隨機分佈的缺陷，另三種則為有不同群聚現象的缺陷分佈。

步驟三：計算缺陷群聚程度指標

以 V/M 值來代表缺陷群聚程度的指標，各晶圓之 V/M 指標值列於表一中。

步驟四：計算 $P(D=0)$ 、 $P(D=1)$ 、 $P(D=2)$ 、 $P(D=3)$ 、 $P(D=4)$ 至 $P(D \geq k)$ 的機率值

利用 KLA 所獲得的晶圓座標圖，我們可以求出晶片上存在的缺陷數，將此缺陷數除以 396 便可以得到 $P(D=0)$ 、 $P(D=1)$ 、 $P(D=2)$ 、 $P(D=3)$ 、 $P(D=4)$ 及 $P(D \geq k)$ 的機率值。經判斷認為所分析的 111 片晶圓中，晶片上缺陷數超過 5 的機率值非常的小，故將晶片上缺陷數大於或等於 5 的機率予以合併，以 $P(D \geq 5)$ 表之。由 111 片晶圓所獲得的缺陷數機率值列於表二中。

步驟五：利用倒傳遞演算法建立良率模式

將步驟二及三求得的五個值作為倒傳遞演算法的輸入值，而以步驟五求得的六個機率值作為倒傳遞演算法的輸出值。一開始我們選擇隱藏層中 PE 的數目為 5、10、25、50、100，固定學習速率為 0.5，其重複訓練 150,000 次後的 RMSE 記錄於表三。由表三中，我們可知當 PE 的數目為 25 個時，網路可以獲得較佳的結果。

表一 各晶圓的缺陷群聚程度

晶圓編號	V/M 值										
1	1.202532	20	1.044846	39	1.271248	58	1.019443	77	14.02311	96	65.92789
2	1.044846	21	1.118657	40	2.21385	59	1.179747	78	1.544304	97	4.640506
3	2.193065	22	6.874561	41	0.98481	60	0.979747	79	5.130567	98	1.97796
4	1.193924	23	0.962025	42	1.853671	61	6.803617	80	1.7804	99	10.09837
5	1.635525	24	1.736709	43	1.918582	62	1.177722	81	1.975784	100	7.690166
6	1.328942	25	1.340253	44	1.493671	63	0.926582	82	1.127089	101	3.162025
7	1.331284	26	1.327595	45	2.864979	64	2.90346	83	1.610127	102	16.72426
8	4.558853	27	5.153381	46	0.979747	65	11.66186	84	1.476534	103	2.568608
9	1.092388	28	1.785005	47	3.167089	66	27.69096	85	5.663436	104	1.900253
10	1.357165	29	9.777994	48	0.972152	67	2.002532	86	10.7762	105	1.152405
11	1.200506	30	1.482079	49	1.123856	68	25.30395	87	2.278481	106	1.587608
12	5.338724	31	4.901266	50	1.202532	69	1.232911	88	11.82684	107	3.888608
13	17.29989	32	6.653096	51	1.422785	70	7.899903	89	1.620515	108	1.337975
14	0.972152	33	1.068354	52	1.028499	71	6.425316	90	27.66751	109	7.477653
15	11.88771	34	1.087342	53	16.87975	72	2.41519	91	6.961181	110	10.70549
16	1.860759	35	1.453165	54	1.321519	73	10.49456	92	2.396962	111	32.08608
17	1.432327	36	3.04557	55	4.16984	74	31.67007	93	1.271019		
18	0.95443	37	1.313328	56	1.177722	75	1.958684	94	2.450196		
19	1.16549	38	2.164777	57	1.278119	76	15.37097	95	6.241664		

表二 各晶圓上缺陷數的機率值

晶圓編號	Pr(D=0)	Pr(D=1)	Pr(D=2)	Pr(D=3)	Pr(D=4)	Pr(D>=5)	晶圓編號	Pr(D=0)	Pr(D=1)	Pr(D=2)	Pr(D=3)	Pr(D=4)	Pr(D>=5)
1	0.979798	0.005051	0.005051	0.005051	0.003037	0.002474	57	0.911616	0	0.025253	0.05303	0.003228	0.002302
2	0.949495	0.005051	0.015152	0.007576	0.002639	0.002398	58	0.939394	0	0.015152	0.032828	0.002574	0.002372
3	0.707071	0.10101	0.146465	0.060606	0.005538	0.001786	59	0.959596	0	0.010101	0.027778	0.002979	0.002423
4	0.931818	0.022727	0.020202	0.020202	0.003015	0.002353	60	0.977273	0	0.005051	0.015152	0.002474	0.002468
5	0.939394	0.015152	0.025253	0.025253	0.00413	0.002372	61	0.974747	0	0.002525	0.065657	0.017181	0.002461
6	0.876263	0.045455	0.025253	0.040404	0.003356	0.002213	62	0.977273	0	0.010101	0.007576	0.002974	0.002468
7	0.954545	0.012626	0.010101	0.010101	0.003362	0.00241	63	0.924242	0	0	0.075758	0.00234	0.002334
8	0.883838	0.042929	0.022727	0.032828	0.011512	0.002232	64	0.808081	0	0.207071	0.10101	0.007332	0.002041
9	0.835859	0.045455	0.037879	0.050505	0.002759	0.002111	65	0.898899	0	0.116162	0.05303	0.029449	0.00227
10	0.896465	0.025253	0.040404	0.030303	0.003427	0.002264	66	0.95202	0	0.005051	0.227273	0.069927	0.002404
11	0.901515	0.017677	0.030303	0.030303	0.003032	0.002277	67	0.919192	0	0.020202	0.068182	0.005057	0.002321
12	0.931818	0.045455	0.015152	0.04798	0.013482	0.002353	68	0.94697	0	0.017677	0.126263	0.063899	0.002391
13	0.95202	0.007576	0.012626	0.065657	0.043687	0.002404	69	0.982323	0	0.007576	0.002525	0.003113	0.002481
14	0.969697	0.005051	0.010101	0.010101	0.002455	0.002449	70	0.929293	0	0.025253	0.14899	0.019949	0.002347
15	0.929293	0.012626	0.040404	0.106061	0.030019	0.002347	71	0.969697	0	0.058081	0.020202	0.016226	0.002449
16	0.959596	0.012626	0.020202	0.022727	0.004699	0.002423	72	0.888889	0	0.010101	0.126263	0.006099	0.002245
17	0.972222	0.012626	0.002525	0.015152	0.003617	0.002455	73	0.858586	0	0.055556	0.179293	0.026501	0.002168
18	0.95202	0.007576	0.015152	0.015152	0.00241	0.002404	74	0.873737	0	0.030303	0.209596	0.079975	0.002206

19	0.957071	0.010101	0.017677	0.010101	0.002943	0.002417	75	0.901515	0	0.032828	0.035354	0.004946	0.002277
20	0.949495	0.002525	0.015152	0.020202	0.002639	0.002398	76	0.959596	0	0.166667	0.030303	0.038816	0.002423
21	0.94697	0.027778	0.007576	0.015152	0.002825	0.002391	77	0.957071	0	0.090909	0.020202	0.035412	0.002417
22	0.959596	0.027778	0.022727	0.020202	0.01736	0.002423	78	0.848485	0.085859	0.032828	0.022727	0.0039	0.002143
23	0.959596	0.005051	0.015152	0.007576	0.002429	0.002423	79	0.775253	0.073232	0.075758	0.073232	0.012956	0.001958
24	0.964646	0.015152	0.010101	0.005051	0.004386	0.002436	80	0.830808	0.050505	0.04798	0.063131	0.004496	0.002098
25	0.94697	0.022727	0.015152	0.017677	0.003384	0.002391	81	0.911616	0.037879	0.025253	0.010101	0.004989	0.002302
26	0.936869	0.005051	0.042929	0.015152	0.003353	0.002366	82	0.929293	0.015152	0.017677	0.032828	0.002846	0.002347
27	0.949495	0.030303	0.022727	0.030303	0.013014	0.002398	83	0.95202	0.025253	0.010101	0.015152	0.004066	0.002404
28	0.94697	0.010101	0.010101	0.012626	0.004508	0.002391	84	0.944444	0.027778	0.015152	0.010101	0.003729	0.002385
29	0.954545	0.030303	0.017677	0.010101	0.024692	0.00241	85	0.911616	0.179293	0.027778	0.017677	0.014302	0.002302
30	0.962121	0.020202	0.002525	0.007576	0.003743	0.00243	86	0.977273	0.007576	0.085859	0.007576	0.027213	0.002468
31	0.941919	0.015152	0.020202	0.032828	0.012377	0.002379	87	0.954545	0.027778	0.010101	0.012626	0.005754	0.00241
32	0.94697	0.010101	0.012626	0.017677	0.016801	0.002391	88	0.901515	0.103535	0.032828	0.143939	0.029866	0.002277
33	0.957071	0.002525	0.017677	0.015152	0.002698	0.002417	89	0.94697	0.022727	0.022727	0.020202	0.004092	0.002391
34	0.962121	0.017677	0.015152	0	0.002746	0.00243	90	0.944444	0.083333	0.020202	0.143939	0.069867	0.002385
35	0.959596	0.010101	0.012626	0.015152	0.00367	0.002423	91	0.929293	0.088384	0.027778	0.04798	0.017579	0.002347
36	0.949495	0.045455	0.005051	0.017677	0.007691	0.002398	92	0.941919	0.027778	0.022727	0.012626	0.006053	0.002379
37	0.962121	0.005051	0.010101	0.010101	0.003316	0.00243	93	0.959596	0.010101	0.010101	0.020202	0.00321	0.002423
38	0.967172	0.027778	0.015152	0.010101	0.005467	0.002442	94	0.95202	0.022727	0.017677	0.017677	0.006187	0.002404
39	0.984848	0.007576	0.002525	0.007576	0.00321	0.002487	95	0.95202	0.022727	0.042929	0.015152	0.015762	0.002404
40	0.94697	0.035354	0.015152	0.010101	0.005591	0.002391	96	0.941919	0.007576	0.007576	0.325758	0.166485	0.002379
41	0.982323	0	0.002525	0.010101	0.002487	0.002481	97	0.939394	0.030303	0.022727	0.055556	0.011718	0.002372
42	0.929293	0	0.030303	0.05303	0.004681	0.002347	98	0.941919	0.012626	0.012626	0.022727	0.004995	0.002379
43	0.909091	0	0.042929	0.045455	0.004845	0.002296	99	0.974747	0.032828	0	0.073232	0.025501	0.002461
44	0.957071	0	0.010101	0.037879	0.003772	0.002417	100	0.967172	0.022727	0.010101	0.058081	0.01942	0.002442
45	0.967172	0	0.007576	0.040404	0.007235	0.002442	101	0.957071	0.027778	0.007576	0.022727	0.007985	0.002417
46	0.977273	0	0.002525	0.020202	0.002474	0.002468	102	0.969697	0.050505	0.002525	0.080808	0.042233	0.002449
47	0.949495	0	0.015152	0.055556	0.007998	0.002398	103	0.977273	0.025253	0.005051	0.007576	0.006486	0.002468
48	0.969697	0	0.007576	0.020202	0.002455	0.002449	104	0.974747	0.005051	0.007576	0.025253	0.004799	0.002461
49	0.969697	0	0.015152	0.017677	0.002838	0.002449	105	0.954545	0.010101	0.015152	0.005051	0.00291	0.00241
50	0.979798	0	0.002525	0.012626	0.003037	0.002474	106	0.959596	0.002525	0.005051	0.015152	0.004009	0.002423
51	0.931818	0	0.020202	0.045455	0.003593	0.002353	107	0.941919	0.020202	0.012626	0.068182	0.00982	0.002379
52	0.916667	0	0.017677	0.05303	0.002597	0.002315	108	0.967172	0.012626	0.007576	0.010101	0.003379	0.002442
53	0.959596	0	0.017677	0.020202	0.042626	0.002423	109	0.914141	0.098485	0.030303	0.133838	0.018883	0.002308
54	0.987374	0	0.007576	0.007576	0.003337	0.002493	110	0.939394	0.032828	0.025253	0.065657	0.027034	0.002372
55	0.964646	0	0.010101	0.020202	0.01053	0.002436	111	0.94697	0.085859	0.027778	0.012626	0.081025	0.002391
56	0.977273	0	0.010101	0.012626	0.002974	0.002468							

表 4-三 PE 數目與 RMSE 的關係

PE 數目	5	10	25	50	100
RMSE	0.1141	0.0627	0.0582	0.0657	0.0664

在決定隱藏層 PE 的數目後，進一步進行類神經網路訓練，當訓練次數約達 250,000 後，網路呈現穩定狀態，即完成建構良率模式的過程。

步驟六：驗證網路的正確性

在完成類神經網路的訓練程序後，接下來便須測試網路的正確性。將 111 筆數據輸入已構建完成的良率模式中，可以得到預測的 IC 良率值及其他相關的機率預測值，再利用所得的結果與估計良率計算 RMSE，可得結果列於表四中：

表四 預測機率與實際值的 RMSE

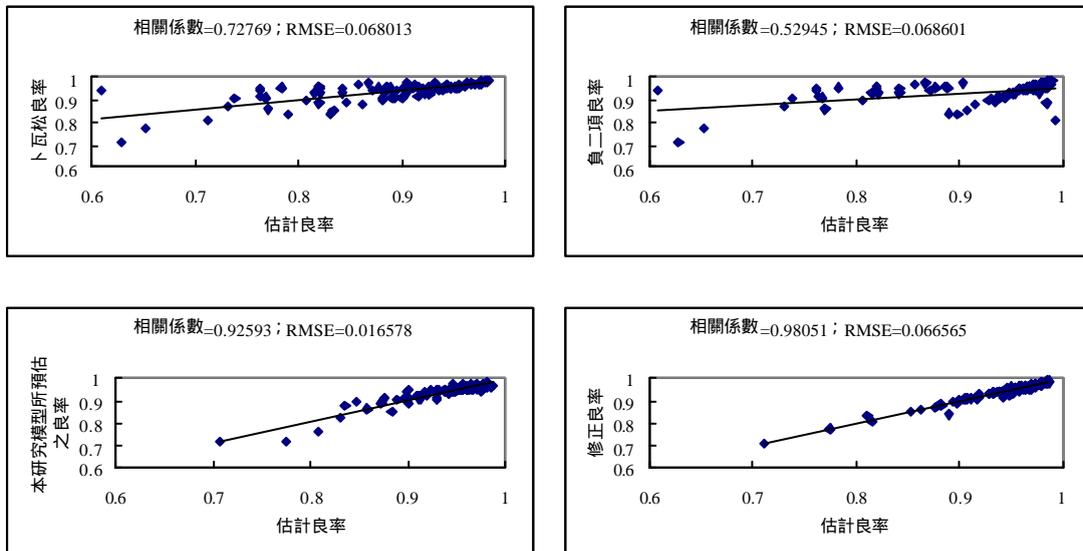
預測機率	Yield	Pr(D=1)	Pr(D=2)	Pr(D=3)	Pr(D=4)	Pr(D>=5)
RMSE	0.016578	0.006046	0.008169	0.013333	0.011352	0.000042

由表四中可知，六個預測結果的 RMSE 值皆非常低，表示此良率模式具有高正確性。

5.2 與其他積體電路良率模式的比較

將本研究所構建的良率模式和卜瓦松良率模式、負二項良率模式及修正良率模式進行比較，以瞭解本研究所構建之良率模式的

預測能力。就各模式所預測的良率與估計良率進行比較，發現本研究所提之良率模式可以獲得較低的 RMSE 值以及相當高的相關係數($r=0.92593$)，如圖五所示，這個結果說明了本研究提供的良率模式具有較佳的預估能力。



圖五 各種良率模式之比較

陸、 結論

傳統的良率模式由於僅考慮到缺陷數目與良率的關係，因此導致這些良率模式在預測良率時發生嚴重的偏差，而修正後的良率模式又太過複雜，實際應用價值不高。本研究因此提出以類神經網路的方法來建構良率模式，並且將不同來源的缺陷以及缺陷群聚的現象同時納入考慮。採用的實例證明本研究所提出的良率模式構建方法，確實比其它的良率模式簡單且預測良率之能力較佳。本研究利用類神經網路所構建之良率模式，包括以下之優點：

1. 利用本研究的方法所構建之 IC 良率模式，在預測良率的準確度上，明顯的優於卜瓦松模式、負二項模式以及修正良率模式。
2. 本研究利用類神經網路構建良率模式，不但使用簡便，同時能夠快速地構建出良率模式。
3. 本研究所提供的良率模式構建方法，可以寫成程式與 KLA 機台整合，並應用於 IC 生產線，對於 IC 生產線上良率的提升應有所助益。
4. 實際 IC 產品的生產需考慮多層加工 (multilayer) 的良率預測問題。本研究因採用類神經網路的技術，故可以整合每一層的資料以預測良率。
5. 可以瞭解不同來源的缺陷造成良率損失的嚴重程度，如此可以在生產的過程中，對於影響良率較大的缺陷來源進行監控或是缺陷分析，以有效地提升良率。
6. 可作為工程師訂立動態隨機存取記憶體 (dynamic random access memory, DRAM) 產品的修復電路配置策略時之依據。

參考文獻

1. Albert, V. Ferris-Prabhu, *Introduction to Semiconductor Device Yield Modeling*, 1992.
2. Mirza, Aghan I., Donoghue, Geoffrey O', Drake, Alvin W., and Graves, Stephen C., "Spatial Yield Modeling for Semiconductor Wafers," 1995 IEEE/SEMI Advanced Semiconductor Manufacturing Conference, pp 276-281.
3. El-Karehn, B., Ghatalia, A., Satya, A. V. S., "Yield Management in Microelectronic Manufacturing," 1995 IEEE/SEMI Advanced Semiconductor Manufacturing Conference, pp 58-63
4. Kang, Boo-Sik, Lee, Janng-Hee, Shin, Chung-Kwan, Yu, Song-Jin, Park, Sannng-Chan, "Hybrid machine learning system for integrated yield management in semiconductor manufacturing," *Expert Systems with Applications*, Vol. 15, 1998, pp. 123-132
5. Cunningham, J. A., "The Use and Evaluation of Yield Models in Integrated Circuit Manufacturing," *IEEE Trans. on Semiconductor Manufacturing*, Vol. 3, No.2, May 1990, pp.60-71.
6. Murphy, B. T., "Cost-size Optimum of Monolithic Integrated Circuits," *Proc. IEEE*, Vol. 52, pp1537-1545, December 1964.
7. Rogers A., *Statistical Analysis of Spatial Dispersion: the Quadrat Method*, Pion, 1974.
8. Collica, Randall S., "A Logistic Regression Yield Model for SRAM Bit Fail Patterns," *IEEE*, 1993, pp127-135.
9. Seeds, R. B., "Yield and Cost Analysis of Bipolar LSI," presented at IEEE

- International Electron Meeting,
Washington, D.C., Oct. 1967.
10. Stapper, C. H., "Defect Density Distribution for LSI Yield Calculations," IEEE Transactions on Electron Devices (Correspondence), Vol. ED-20, July 1973, pp. 655-657.
 11. Stapper, C. H., "Large-Area Fault Clusters and Fault Tolerance in VLSI Circuits: A Review," IBM J.RES. DEVELOP., Vol. 33, No. 2, March 1989, pp.162-173
 12. Stapper, C. H., "Small-Area Fault Clusters and Fault Tolerance in VLSI Circuits," IBM J. RES. DEVELOP., Vol.33, No.2, March 1989, pp.174-177.
 13. 許文輝, 鍾雲恭, 薛如珊, "模糊自我成圖神經網路在晶圓缺陷分類的應用," 中華民國品質學會第三十四屆年會暨第四屆全國品質管理研討會論文集, pp.571-580
 14. 趙豐昌, "利用類神經網路構建之積體電路量率預估模式", 國立交通大學工業工程與管理學系碩士論文, 民國八十六年

