行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※※※※※

利用基因演算法建構具有功能的中間節點之類神經網路

※※※※※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：個別型計畫

計畫編號：NSC－89－2218－E－009－019

執行期間：89 年 08 月 01 日 至 90 年 07 月 31 日

計畫主持人：李嘉晃

共同主持人：

本成果報告包括以下應繳交之附件：

□赴國外出差或研習心得報告一份

□赴大陸地區出差或研習心得報告一份

□出席國際學術會議心得報告及發表之計畫各一份

□國際合作研究計畫國外研究報告書一份

執行單位：國立交通大學資科所

中 華 民 國 年 月 日

行政院國家科學委員會專題研究計畫成果報告
國科會專題研究計畫成果報告撰寫格式說明
Preparation of NSC Project Reports

計畫編號：NSC 89－2218－E－009－019
執行期限：89 年 8 月 1 日至 90 年 7 月 31 日

主持人：李嘉晃
執行單位：國立交通大學資科所

一、中文摘要

在傳統的類神經網路（Artificial Neural Networks，ANNs）裡，所有的隱藏層節點（Hidden Node）皆被視為一致的個體，配合倒傳遞演算法（Back Propagation Algorithm）調整權值（Weight）以解決問題。然而倒傳遞演算法有訓練效率過低的缺點，並且在整合基因演算法（Genetic Algorithm，GA）演化類神經網路結構時，運用此法調整權值，也容易產生結構與權值無法配合的問題。本計劃提出賦予隱藏層節點特定的功能，節點據此調整權值，期望所得的輸出與指定功能的目標值一致。而整體網路的訓練便是以單層的網路訓練為基礎，由底而上漸次完成。並且利用基因演算法，選擇適當的節點功能，安排在適當的位置，讓整個系統得以因應不同的問題，自動產生適合的類神經網路結構。本方法具有隱藏層節點的定位確定，節點功能的再利用，網路訓練透明化與增益演化效率幾項特點。

關鍵詞：基因演算法、類神經網路、功能化隱藏結點

Abstract

In traditional Artificial Neural Networks, all hidden nodes are regarded identical, and the nodes adapt weights to solve problems with Back Propagation Algorithm. However, the speed of convergence of Back Propagation is slow, and when we incorporate Genetic Algorithms in evolving Artificial Neural Networks with this method tuning weights, the network architecture and weights will not match well easily. This project proposed giving hidden nodes specific functions. Then, the nodes change weights to make their outputs similar to the goals of the functions of them. The whole neural network training is based on the single neural networks training and then completes. In addition, we use Genetic Algorithm to select the functions for right nodes. The system will automatically generate a fit Artificial Neural Network for various applications respectively. The primary characteristics of this method include the niche of hidden nodes, the re-usefulness of node functions, the transparency of network training, and the improvement of evolution performance.

Keywords: Genetic Algorithms, Neural Networks, Functional Hidden Nodes

二、緣由與目的

近年來，人工智慧的研究領域愈來愈受到重視，許多過去難以解決的問題，都希望能藉由其範疇內的方法來解決。其中，類神經網路（Artificial Neural Networks，ANNs）與基因演算法（Genetic Algorithm，GA）是最備受青睞的兩種機制模型。在傳統的類神經網路訓練裡，倒傳遞演算法（Back-Propagation Algorithm）[1][2]是最早被提出，也是當時最有效率的方法。它將輸出端應得的正確值，減去實際的輸出，並將誤差的效應往輸入層傳遞，作為調整權值（Weight）與節點臨界值（Threshold）的憑藉。然而在利用倒傳遞法學習調整權值與臨界值時，每一條權值與每一個節點，完全無法知道自身在整體網路中的定位，因此無法明確且迅速地朝某個目標作調整。同時又會因為訓練不同的樣本（Pattern），而受到其他的節點與權值影響，導致原本已經調整好的數值又被迫改變，甚至與原來的數值呈反向的成長，這就是所謂的「獸群效應（Herd Effect）」[3]。而這也正是倒傳遞演算法最被詬病之處：匯集（Convergence）速度過慢。另一個在應用類神經網路時所遭遇的問題為：無法在解決問題之初，就明確的知道應該設計何種網路結構，才具有足夠的能力處理該問題。一般早期的解決之道，是研究人員憑藉過去的經驗猜測並且測試，然後再根據所得的實驗結果做調整。近期許多學者引進基因演算法，以類似生物進化的方式演化類神經網路，使其自動找到適合的網路架構[4][5]。可是在演化的過程裡，常會發生網路結構與權值不一致的情形。而當此問題發生時，研究人員並不清楚是網路結構不合適，或是權值還未調整至適當的值。

在本計畫裡，我們提議賦予類神經網路的隱藏層節點 (Hidden Layer Nodes) 一些功能。使其在訓練的過程中，與整合基因演算法演化網路結構時，能更有效率。這些功能的選擇可以是隨機的，也可以是來自於對應用的部分瞭解所產生的。為了解決上節所提到的問題，本計劃提出一個方法 設計類神經網路內的隱藏層節點具有特定的功能，並且利用基因演算法演化網路架構。我們賦予每一個隱藏層節點指定的功能，當此節點接受輸入後，根據所屬的功能，會有相對應的輸出值。而此節點只要根據這些相對應的輸出，訓練所屬的權值與臨界值，使得其輸出與目標值儘量一致，然後再將其輸

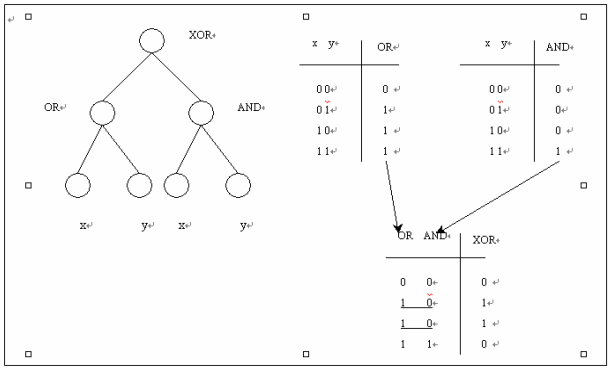
出值傳遞給上一層的節點即可。與傳統利用倒傳遞演算法訓練類神經網路不同的，節點的臨界值與所屬權值的訓練，完全不會受到頂層輸出端所得誤差值的影響。當然正在訓練的權值與臨界值，也不會受到其他節點與權值的影響，故將不會發生獸群效應。而整個類神經網路的訓練，就是以個別單層網路的訓練為基礎，由底而上漸次完成。而除了改變訓練類神經網路的方式，與新增加功能外，整個網路隱藏層節點的數量，所配置的位置和各具有的功能，以及輸出、輸入和隱藏層節點間的連結關係，都是交由基因演算法因應不同的應用自動演化。由於每一個節點都有自身的定位，故在演化的過程裡，所屬的權值將會朝一特定的方向作調整，而不需擔心網路結構與權值無法配合的問題。我們希望本方法能提供一個更佳的解答機制，因應不同的問題自動產生適合的類神經網路。而另一個更長遠的目標，就是將節點視為一個軟體物件，連結的權值視為物件間的輸出與輸入介面，利用自然競爭與演化的方式，將它們放置到適合的位置上，彼此「合作」，則整個的網路架構可以視為另一個因應不同的應用，自動產生的整合型軟體。這不但符合軟體元件再利用的精神，同時具有類似程式自動化的意味，而這也正是我們最後希望達成的目標。

三、方法

所謂指定功能的隱藏層節點，就是此節點具有一個特定的功能，它能根據不同的輸入而有一定相對應的輸出值。舉例言之，如果一個節點具有布林代數 AND 的功能，而且其輸入只有「正值」(Positive Value) 與「負值」 (Negative Value) 兩種型態，那麼除了全部的輸入皆為正值外，其他的輸入組合皆會得到結果為負值的輸出。這個定義在多層類神經網路架構的訓練中指明了一件事，即任何具有指定功能的隱藏層節點，都只需負責自身所屬的輸入，而不會受到其他權值改變的影響。同時這也表示整個類神經網路的訓練過程，是由各個單層的網路訓練所堆砌出來的。圖一說明如何利用本方法，訓練類神經網路達到 XOR 的功能。圖一設計一個能夠學習 XOR 的三層類神經網路。底層的 x 和 y 為二元輸入，中間標示 OR 和 AND 的節點，為各具有布林代數 OR 與 AND 功能的隱藏層節點，而標示 XOR 的節點為輸出端，表示此網路最後應習得的功能。x 與 y 分別同時輸入到標示具有布林代數 OR 與 AND 的節點做訓練，由於 OR 和 AND 為線性可以區隔的問題 (Linear Separable Problem)，故利用簡單的差別法 (Delta Rule) 即可在至多幾十個訓練週期 (Epochs) 完成各自對應的正確輸出的訓練。然後這兩個隱藏層節點再將所對應的輸出，傳給最上層的單層網路作為輸入，由真值表可知，最後單層網路所需解決的問題，也是簡單線性可區隔的問題，故此單層網路也可以輕易的解決問題，而最後整個類神經網路便學習到 XOR 的功能，於是整個網路的訓練便結束。

相對於一般利用倒傳遞演算法訓練多層類神經網路的方式，我們提出的方法具有以下的兩個主要優點：

1. 每個隱藏層節點皆有明確的目標，故能有效率地調整所屬的權值，務使其輸出與目標值一致，同時在學習過程中不會發生所謂的「獸群效應」。
2. 整個網路的訓練是各個單層網路訓練的總和結果。各單層網路一經訓練完畢，便凍結其權值不再改變，故可以視之為一個具有特定功能的基本單位。當其他上層的節點需要利用到已經完成訓練的節點時，此節點僅需將它已經完成訓練的輸出，傳遞給上層網路即可，而無須再行訓練，此方式將可大幅增益整體網路訓練的成效。



圖一

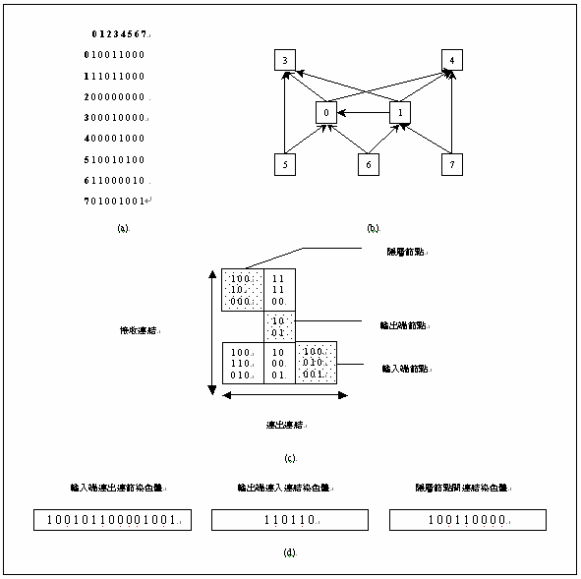
誠如在開頭已經提過，在設計類神經網路架構解決問題之前，必須先猜測何種的架構才足以應付所需。通常皆是研究人員憑藉著過去的經驗猜測，然後試驗，根據所得的結果再行調整。可是此方法不僅浪費人力，沒有效率，而且通常找出的類神經網路架構也是僅為堪用，並不是最佳化，或次佳化。倘若是再換成解決相類似的問題，也許還是得從新再設計一次網路，至少也必須浪費人力在調整部份網路結構。有鑑於此，利用基因演算法演化類神經網路的方法被引用進來，希望能藉由其具有在龐大資料中搜尋的能力，自動產生最佳化的網路結構解決問題。從過去許多成功的實驗結果中，發現到利用此法應用在演化類神經網路方面應該是個正確的決定。

目前一般的應用方法，主要分為下列三種：

1. 直接將整個網路的結構（節點數，連接方式，有些也包括權值）全部編碼放置入染色體中，整體進行演化。
2. 不直接演化整個類神經網路，而是演化產生網路架構的規則 [4]。
3. 將整體網路架構以節點為單位，分割成多個染色體。然後群體競爭，最後再將表現優異的染色體群整合，以成一完整網路架構。

我們在基本精神上是採用第一種方式，也就是直接將整個網路的結構（不包括權值）編碼置入染色體。之所以採用此方法，是因為在研究初期比較容易控制整個網路的節點數，隱藏層節點的功能與配置的位置，以及節點間的連結方式，而且在目前的實驗中皆能得到不錯的成果。然而不可否認的，若要繼續解決更實際複雜的問題，採用第二種演化產生網路結構規則的方式，應該會有更好的成效表現。由於我們提出具有特定功能的隱藏層節點，故演化網路架構的主要問題為：隱藏層節點的配置方

式與具有的功能，和所有節點間連結的方式，這與第三種將網路以節點為單位切割成不同的染色體，群體競爭找出表現優異的個體，然後再行組合產生網路的方法所著重之處不同。所以基於以上的原因，我們最後決定採第一種方法演化類神經網路架構。從許多的研究裡，我們發現到許多演化類神經網路的方法，都是將輸入、輸出和隱藏層節點一起演化 [4]。可是在解決問題之前，我們已經明確的知道輸入與輸出端的個數，所不曉得的僅為隱藏層節點的數量，和所有節點間連結的關係。故我們將輸入與輸出端獨立抽取出來，各成為一條染色體，希望能更有效率的找出最佳的網路架構。圖二顯示以一個矩陣表示網路結構，與其染色體和外在的型態。圖二 (a) 是將整個網路的架構以矩陣陣列的方式表示，位於對角線的位元 (Bits)，表示此位置的節點存在與否，”1” 表示存在，”0” 表示不存在。而對於每一個節點而言，其同列 (Row) 的位置表示從此節點發出到對應的節點的鏈結，”1” 表示存在，”0” 表示不存在；其同行 (Column) 的位置表示此節點從所對應的節點接收的鏈結，相同地，”1” 表示存在，”0” 表示不存在。根據圖二 (a) 的矩陣圖，圖二 (b) 為其外在型態的類神經網路架構。然而在此要特別說明，在從個體的染色體基因轉換為網路結構時，必須先除去一些不合適的部分，這些部份分別為： (1) 所有的鏈結不能連到不存在的節點，或是從不存在的節點連出。(2) 由於我們所討論的僅是非遞迴 (Non-Recurrent) 的網路，故所有遞迴的鏈結都必須去除。所謂的去除僅是在轉換過程中做調整，以產生一個正確且適合的網路，而非對染色體內的基因做改變。圖二 (c) 將圖二 (a) 更細部的劃分成三個部份，分別是輸入端，輸出端與隱藏層節點。誠如前述所言，輸入和輸出的節點數在問題確定的同時已經被決定了，所以僅剩連結的問題；可是隱藏層節點除了連結的問題外，還有數量與位置的問題。數量決定網路結構的大小，而位置決定此節點具有的功能。在此對於位置決定功能作更詳細的說明：每一個座落在隱藏層節點矩陣裡，對角線上的位元，皆表示隱藏層節點可能出現的位置，而每一個位置預先都已經被指定具有特殊的功能，此功能可以是系統隨機產生，抑或已經被指定的；換言之，只要隱藏層節點出現在此位置，就具備此位置設定的功能。以圖二 (c) 的隱藏層區塊為例，對角線上的第一與第二個位元為 "1"，表示這兩個隱藏層節點存在，而假若系統預先已經設定對角線上的三個位元，分別具有 AND，OR 與 NAND 的功能，則第一個隱藏層節點便具有 AND 的功能，而第二個則具有 OR 的功能。由於位於對角線上的第三個位元為 "0"，故依此矩陣產生的類神經網路僅有兩個隱藏層節點，且分別具有 AND 與 OR 的功能。從圖二 (c) 中可以很明顯的看到，由於輸入端的數量與位置已經被確定，故其僅需負責與輸出端和隱藏層節點的連結關係，於是輸入端只是將此連結關係編碼置入染色體內；相同的，輸出端也是僅需負責與隱藏層節點間的連結關係，並且也僅將此關係編碼置入染色體中；而隱藏層節點部份不但要負責與自身群體間的連結關係，還要決定數量與位置的問題，是故需要將這些資訊皆編碼置入染色體內。圖二 (d) 示出分屬於輸入端，輸出端與隱層節點的染色體。圖二 (d) 輸入端的染色體基因，是圖二 (c) 左下角兩個白色區塊內的位元，以列為單位上至下的放置；而圖二 (d) 輸出端的染色體基因，是圖二 (c) 中上白色區塊內的位元，以行為單位左至右放置；圖二 (d) 隱藏層的染色體基因，是圖二 (c) 左上網狀區塊（隱藏層區塊）內的位元，以列為單位由上而下的放置。



圖二

四、結果

參照先前的相關文獻，我們發現 XOR 與 N-Parity 函數為最常被用來測試類神經網路學習成效，同時也是最簡單的標竿測試資料 (Benchmark)。XOR 為布林代數中的一個函數，屬於線性不可區隔的問題。它具有兩個輸入與一個輸出，當兩個輸入值相同時（同為 0 或同為 1），則輸出值為 0；反之，當兩個輸入值不一致時，則輸出為 1，故共有 4 個訓練樣本。而 N-Parity 的函數則是 XOR 的延伸，”N” 表示此函數有 N 個輸入，同樣地也只有一個輸出。當 N 個輸入中有奇數個 1 時，輸出為 1；反之，則為 0，所以此函數總共有 2^n 個學習樣本。

執行次	演化代數(generations)	隱層節點數
1	1	4
2	1	2
3	1	2
4	1	3
5	1	1
6	1	3
7	1	3
8	1	3
9	1	2
10	1	2

表格一 隨機挑選功能學習 XOR 的結果

執行次	演化代數(generations)	隱層節點數
1	10	3
2	1	1
3	2	3
4	1	2
5	1	4
6	1	3
7	1	2
8	1	1
9	1	2
10	1	5

表格二 隨機挑選功能學習 3-Parity 函數的結果

實驗各應用一些基本參數至 XOR 與 Parity 函數的實驗中，並分別獨立執行 10 次 (Runs)，所得的結果以表一與表二示之。由表一及表二中可以看出，利用本方法解決 XOR 和 3-Parity 的問題是相當有效率的，同時所建構出的網路架構也相當精簡。但是由於這兩個問題過於簡單，因此我們又以另外一個較複雜的例子，驗證本方法確實有效。

鳶尾花 (Iris) 的分類

鳶尾花的分類問題，是三個分別具有 50 筆，總共 150 筆的資料，描述花朵特性與種類，讓系統根據所學習的資料，找到三個種類的特性的分類問題。本實驗最大隱藏層節點數設為 8，節點功能由系統隨機產生。表三顯示執行 10 次的結果。從過去的文獻中 [6][7]，知道鳶尾花分類問題的錯誤，約在 0 到 5 個之間。從表三中可知，所得的結果並不亞於一般的分類方法。

執行次	演化代數	隱層節點數	訓練正確率	測試正確率
1	1	6	72/75	71/75
2	3	3	72/75	71/75
3	1	4	72/75	71/75
4	1	3	72/75	69/75
5	2	3	72/75	71/75
6	1	3	72/75	72/75
7	1	3	72/75	71/75
8	1	3	72/75	70/75
9	2	5	72/75	70/75
10	1	2	72/75	71/75
平均值	1.4	3.5	96%	94.27%

表格 三 隨機挑選功能學習鳶尾花分類的結果

五、結論

本計畫提出一個新的觀念：設定類神經網路的隱藏層節點具有特定的功能，並利用基因演算法自動搜尋適合的網路架構，以因應不同的應用。這就好比一家公司，有許多人分配至不同的職位上。這些在位者，必須調整自己以適應新的職務；同時就整體而言，公司又必須視總體表現，決定工作的分配，部門的規畫，與職務的調動。從所得的實驗結果中，可以驗證應用此想法在解決不同的問題上，確實具有不錯的成效。底下歸結本方法的特點。

本模型的特點

1．隱藏層節點的定位確定

各個隱藏層節點都有特定的功能，它們僅需配合自身的輸入，調整所屬的權值，盡量達到設定的功能即可，而不會被其他各層的隱藏層節點與權值的變化所影響。如此將使得各隱藏層節點有效率地找到自身的定位，而不會發生獸群效應。

2．量身定做的節點功能

各個節點的功能，可以依應用的不同而有不同的設計。此外，相較於其他一般訓練網路與演化結構的模型，本方法還具有另外一項特點，那就是可以加入「提示 (Hint)」，使得所演化的類神經網路架構更趨最佳化，同時演化與訓練的效率更高。所謂的「提示」，即在解決問題之前，對該問題有一定的瞭解，曉得哪些功能的加入，將可以增益訓練的效率，於是設定節點具備該功能。

3．節點功能的再利用

由於各個隱藏層節點一經訓練完畢，其所屬的權值將不再變動。當其他未完成訓練的上層節點，需要利用到已經完成訓練的節點時，僅需直接取得相對應的輸出，而無須再行訓練，以達到功能節點再被利用之效，增進系統效能。

4. 網路訓練透明化

過去通常視類神經網路的訓練為一個黑盒子，訓練成功與否很難去分析與解釋，而追蹤與剖析錯誤更是困難重重。可是採用本方法可以明確的知道，各個節點應該有哪些相對應的輸出，這在偵測錯誤，分析與解釋訓練結果上有極大的助益。

5．增益演化效率

1．由於本系統採用基因演算法演化類神經網路，故當系統從其他的實驗結果中，得到一個不錯的網路結構時，可以以此網路結構作為基礎，再進行演化以獲得更佳的網路結構，而此舉可以節省演化的時間。

2．避免在演化類神經網路的過程裡，發生網路架構與權值不配合的情形。

3．將輸入、輸出與隱藏層節點分開演化，減少搜尋的空間，避免無謂的搜尋。

五、參考文獻

[1] Parker, D. B., *Learning-Logic*. Technical Report TR-47, Massachusetts Institute of Technology, Center for Computational Research in Economics and Management Science, Cambridge, MA, 1985.

[2] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., Learning Internal Representations by Error Propagation. In Rumelhart, D. E. and McClelland, J. L. (editor), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, chapter 8. MIT Press, Cambridge, MA, and London, England, 1986.

[3] Fahlman, S. E. and Lebiere, C., The Cascade-Correlation Learning Architecture (Technical Report CMU-CS-90-100). Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, 1990.

[4] Kitano, H., Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4:461-476. 1990.

[5] Harp, S. A., Samad, T., and Guha, A., Towards the genetic synthesis of neural networks. In Schaffer, J. D., editor, *Proceedings of third International conference on Genetic Algorithms*, pages 360-369, George Mason University, USA. Morgan Kaufmann, 1989.

[6] Geva, S., Production rule extraction. *Neural Networks*. Proceedings, IEEE International Conference, vol. 4, pp. 1806 –1811. 1995.

[7] Copland, H.& Hendtlass, T., Engram decay in artificial neural networks. *Neural Networks*. Proceedings, IEEE International Conference, vol. 1, pp. 669 –673. 1995.