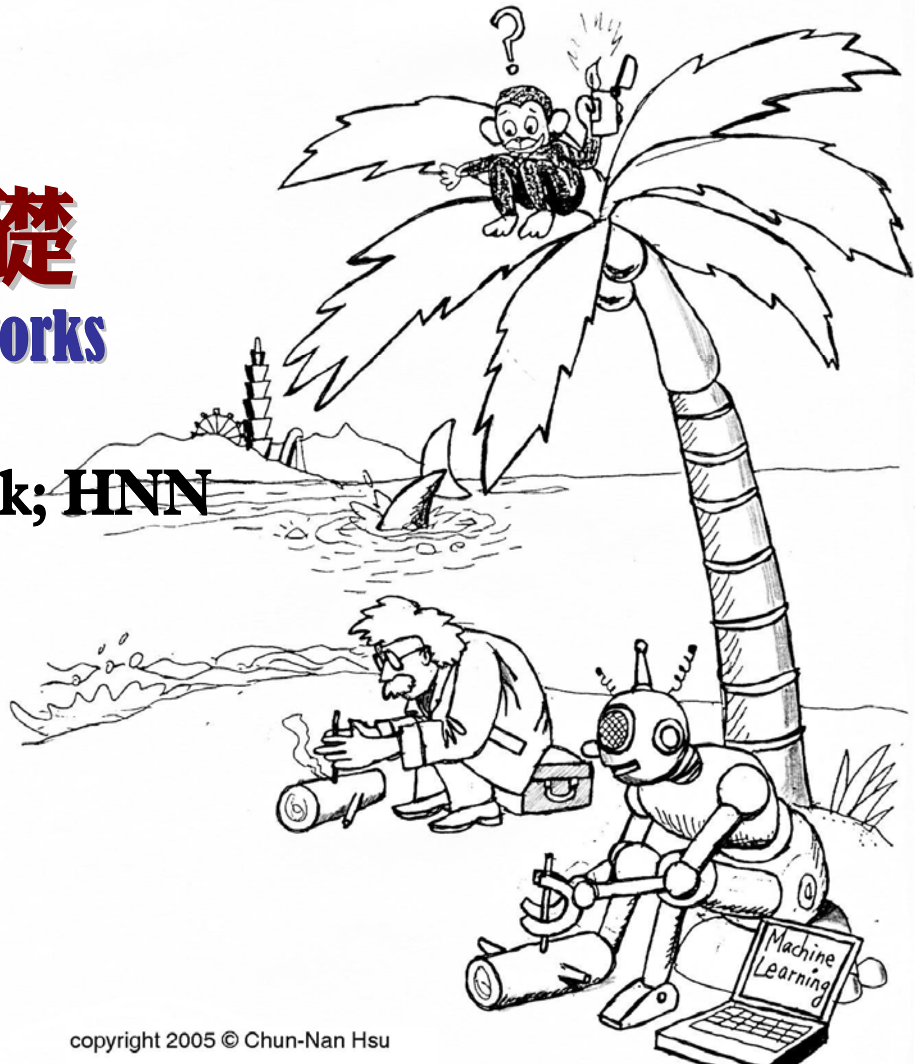
類神經網路基礎

Foundations of Neural Networks

-- 霍普菲爾網路

Hopfield Neural Network; HNN



授課教師：陳士杰

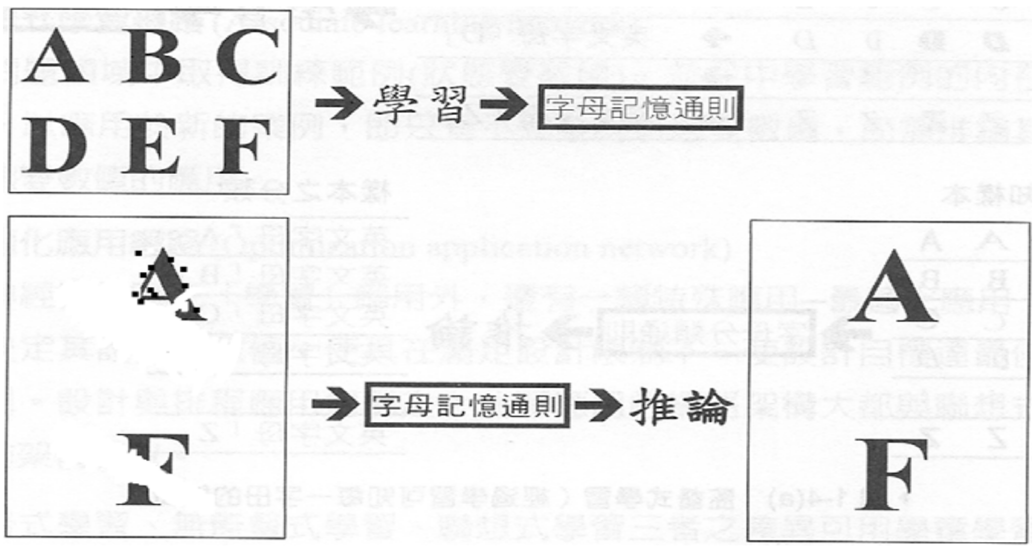
國立聯合大學 資訊管理學系

■本章授課內容

- 導論
- 網路架構
- 網路演算法
- 網路手算範例
- 結論

■導論

- **霍普菲爾網路 (Hopfield Neural Network; HNN)是一種聯想式學習網路，它是由 J. Hopfield 於1982年提出。**
- **所謂的聯想式學習，是從訓練範例中，學習範例的內在記憶規則，以應用於當有不完整輸入狀態資料時，需推論其完整資料之應用。**

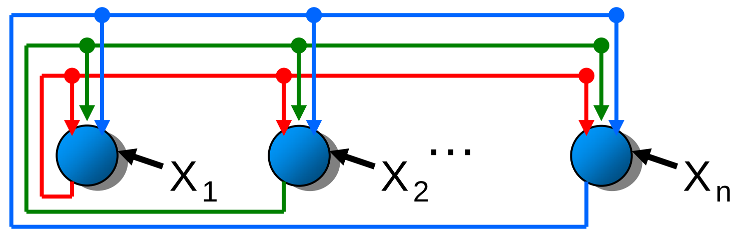


- **適用於資料擷取、雜訊過濾與誤差校正之應用。**

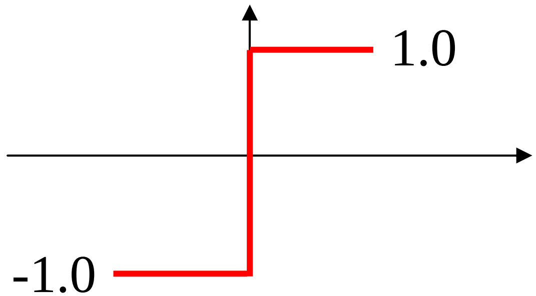
- **聯想式學習分成兩類：**
  - **自聯想 (Auto-associative)：由一個樣式聯想同一個樣式者。**
    - **例如：聽到某一歌曲片段，可以聯想起該首歌曲的完整旋律。**
  - **異聯想 (Hetero-associative)：由一個樣式聯想另一個樣式者。**
    - **例如：聽到電腦不同類型的嗶嗶聲時，可以聯想起相對應的系統問題。**
- **霍普菲爾網路屬於「自聯想」模式。**
  - **先提供許多訓練範例，每個範例有一組二極值{-1, +1}的輸入特徵向量，由這些範例學習一個聯想記憶規則，使網路能記憶這些訓練範例的特徵向量。**
  - **接著，若有不完整或是有雜訊的資料輸入，網路即能聯想起與其最相近的訓練範例資料。**

■網路架構

- 霍普菲爾網路的架構如下，只有單層：



- 處理單元：
  - 用以輸入訓練範例的資料，輸入變數必須是二極值 {-1, +1} (二元值 {0, 1}亦可)。
  - 其處理單元數目依問題而定，使用非線性轉換函數。



網路連結：

- 任意兩處理單元間皆有連結存在。
- 每個處理單元之間的連結加權值，代表兩者間的互動關係。公式如下：

$$W_{ij} = \Sigma_p X_i^p \times X_j^p$$

其中 $W_{ii} = 0$；p 表示第 p 個訓練範例；X_i^p 表示第 p 個訓練範例中，第 i 個輸入變數值。

- 從公式可得知，如果單元間的加權值為正，代表兩個單元間傾向同號 (同為負或同為正)；反之，若加權值為負，代表兩個單元間傾向異號，即：一正一負。

- 霍普菲爾網路有兩種類型：
  - 離散型
    - 網路處理單元的狀態只有兩個值，可取-1和+1，或是0和+1皆可。
  - 連續型
    - 網路處理單元的狀態可取0和1之間的任一實數
- 在此，我們介紹離散型霍普菲爾網路。

■網路演算法

- **類神經網路的運作模式有兩種：**
  - **學習過程 (Learning)**
    - **網路從範例中學習，以調整網路連結加權值的過程。**
  - **回想過程 (Recalling)**
    - **網路由輸入資料決定輸出資料結果的過程。**

學習過程：2個步驟

1. **設定網路參數**
   - $N_{in} = n$ (視每一組訓練範例的輸入變數之個數而定)
2. **設定加權矩陣W的值**
   - $W_{ij} = \Sigma_p X_i^p \times X_j^p$，其中 $W_{ii} = 0$

回想過程：5個步驟

1. 設定網路參數。
2. 讀入加權值矩陣。
3. 輸入一組測試範例的狀態向量X。
4. 計算新的狀態向量X。
   - 先算出所有處理單元的加權乘積和：$net_j = \Sigma_i W_{ij} \times X_i$
   - 利用 net_j 判斷所有處理單元的輸入狀態值是否要更新，方式如下：
     - 若 $net_j > 0$，則處理單元 $X_j = 1$。
     - 若 $net_j = 0$，則處理單元 X_j 不變。
     - 若 $net_j < 0$，則處理單元 $X_j = -1$。
5. 重覆步驟 4，直到收斂 (狀態向量不再有明顯變化) 或執行一定數目的學習循環。

■網路手算範例

- 利用2×3的黑白點所產生的4組圖樣來做範例。

圖樣	■□■ □■□	□■□ ■□■	■■■ ■■■	□□□ □□□
編號	1	2	3	4

- 在此，以 -1 代表白點，+1 代表黑點，0 表示未知。以由左到右且由上到下的方式做編碼。四個圖樣所產生的二極值如下所示：
  - {+1 , -1 , +1 , -1 , +1 , -1 }
  - {-1 , +1 , -1 , +1 , -1 , +1 }
  - {+1 , +1 , +1 , +1 , +1 , +1 }
  - {-1 , -1 , -1 , -1 , -1 , -1 }

	X_1	X_2	X_3	X_4	X_5	X_6
p=1	+1	-1	+1	-1	+1	-1
p=2	-1	+1	-1	+1	-1	+1
p=3	+1	+1	+1	+1	+1	+1
p=4	-1	-1	-1	-1	-1	-1

- 本題有四個訓練範例，每個訓練範例共有六個輸入變數值。故上述訓練範例可用一個二維陣列表示。

學習過程：

- 輸入訓練範例：

	X_1	X_2	X_3	X_4	X_5	X_6
p=1	+1	-1	+1	-1	+1	-1
p=2	-1	+1	-1	+1	-1	+1
p=3	+1	+1	+1	+1	+1	+1
p=4	-1	-1	-1	-1	-1	-1

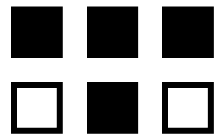
- 由以下公式，可以得到一個加權值矩陣。
  - $W_{ij} = \Sigma_p X_i^p \times X_j^p$，其中 $W_{ii} = 0$

$$W_{ij} = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 & 0 & 4 \\ 4 & 0 & 0 & 0 & 4 & 0 \\ 0 & 4 & 0 & 0 & 0 & 4 \\ 4 & 0 & 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 & 0 \end{bmatrix}$$

$$W_{13} = \Sigma_p X_1^p \times X_3^p$$
$$=(+1)(+1) + (-1)(-1) + (+1)(+1) + (-1)(-1)$$
$$= 4$$

回想過程：

- 假設有一個具有雜訊或不完全的圖樣如下：



即：$X = \{+1, +1, +1, -1, +1, -1\}$

- 計算新的狀態向量X
  - (第一次迭帶)：先計算每個處理單元的加權乘積和：

$$net_j = \Sigma_i W_{ij} \times X_i \quad \begin{bmatrix} net_1 \\ net_2 \\ net_3 \\ net_4 \\ net_5 \\ net_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 & 0 & 4 \\ 4 & 0 & 0 & 0 & 4 & 0 \\ 0 & 4 & 0 & 0 & 0 & 4 \\ 4 & 0 & 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} +1 \\ +1 \\ +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} +8 \\ -8 \\ +8 \\ 0 \\ +8 \\ 0 \end{bmatrix}$$

- **再利用 net_j 判斷每個處理單元的狀態值是否要更新，方式如下：**
  - **若 $net_j > 0$，則處理單元 $X_j = 1$。**
  - **若 $net_j = 0$，則處理單元 X_j 不變。**
  - **若 $net_j < 0$，則處理單元 $X_j = -1$。**

原本的X = {+1, +1, +1, -1, +1, -1}：

新的X：

$$\begin{bmatrix} net_1 \\ net_2 \\ net_3 \\ net_4 \\ net_5 \\ net_6 \end{bmatrix} = \begin{bmatrix} +8 \\ -8 \\ +8 \\ 0 \\ +8 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ \\ \\ \end{bmatrix}$$

- (第二次迭帶)：利用上一次迭帶後所得之新的狀態向量X= {+1, -1, +1, -1, +1, -1} ，再次計算每個處理單元的加權乘積和：$net_j = \Sigma_i W_{ij} \times X_i$，然後利用 net_j 判斷每個處理單元的狀態值是否要更新：
  - 若 $net_j > 0$，則處理單元 $X_j = 1$。
  - 若 $net_j = 0$，則處理單元 X_j 不變。
  - 若 $net_j < 0$，則處理單元 $X_j = -1$。

新的X：

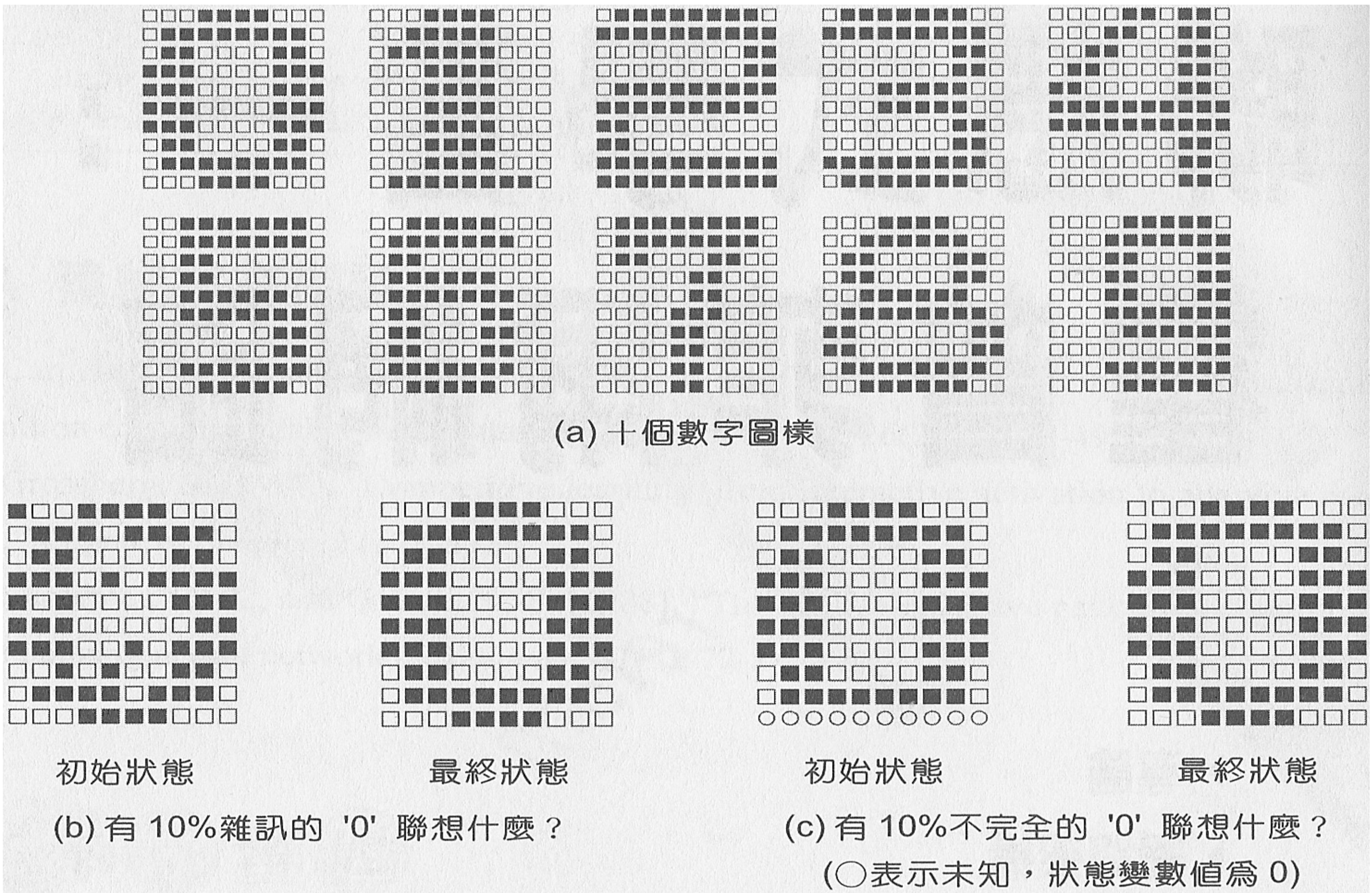
$$\begin{bmatrix} net_1 \\ net_2 \\ net_3 \\ net_4 \\ net_5 \\ net_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 & 0 & 4 \\ 4 & 0 & 0 & 0 & 4 & 0 \\ 0 & 4 & 0 & 0 & 0 & 4 \\ 4 & 0 & 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} +8 \\ -8 \\ +8 \\ -8 \\ +8 \\ -8 \end{bmatrix} \Rightarrow \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix} = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \\ +1 \\ -1 \end{bmatrix}$$

- 因已無狀態變數值改變，因此停止迭帶。結果圖樣為：■ □ ■

(同第一個訓練範例) □ ■ □

阿拉伯數字之應用

- 霍普菲爾網路屬於聯想式學習網路，可應用於雜訊過濾與資料擷取。



(a) 十個數字圖樣

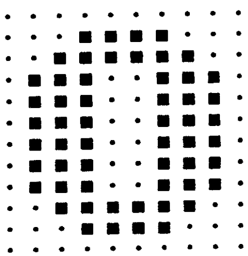
(b) 有 10%雜訊的 '0' 聯想什麼？

(c) 有 10%不完全的 '0' 聯想什麼？

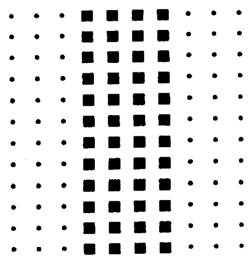
(○表示未知，狀態變數值為 0)

範例資料2：

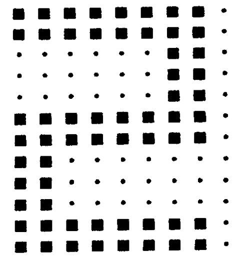
(本圖示摘自：S. Haykin, Neural Networks)



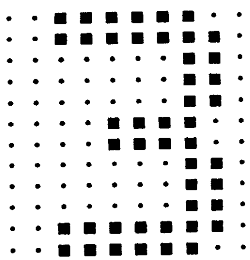
圖樣 "0"



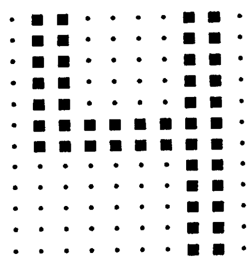
圖樣 "1"



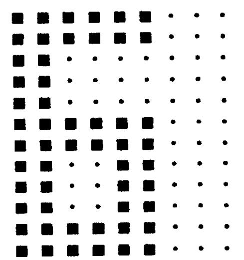
圖樣 "2"



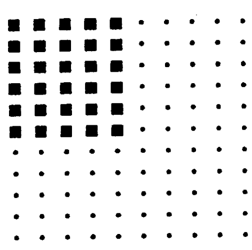
圖樣"3"



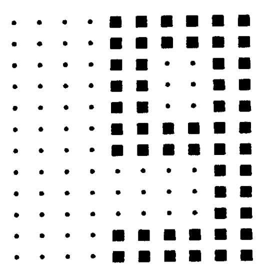
圖樣 "4"



圖樣"6"

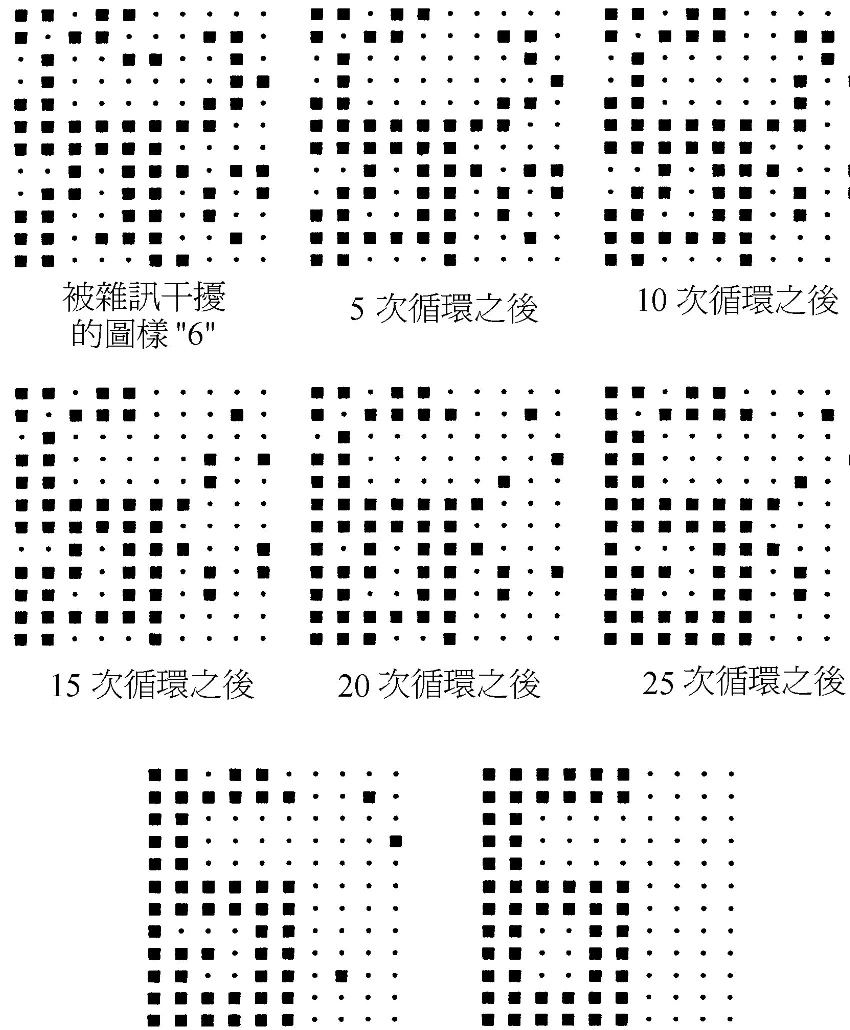


圖樣 "■"



圖樣 "9"

範例資料2正確回想：



被雜訊干擾的圖樣 "6"

5 次循環之後

10 次循環之後

15 次循環之後

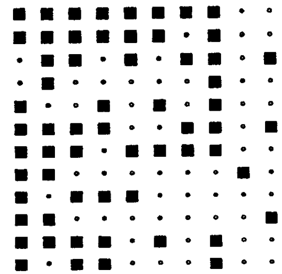
20 次循環之後

25 次循環之後

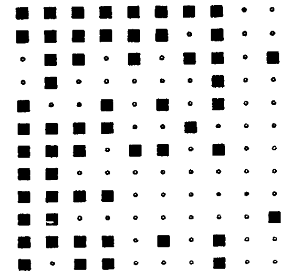
30 次循環之後

37 次循環之後

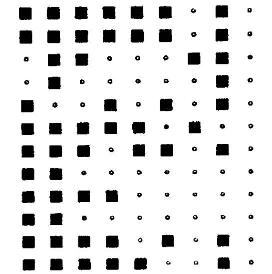
範例資料 2 錯誤回想：



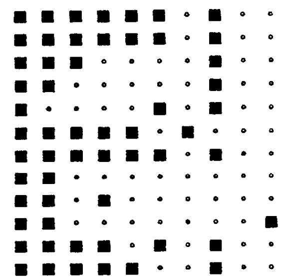
被雜訊干擾的圖樣"2"



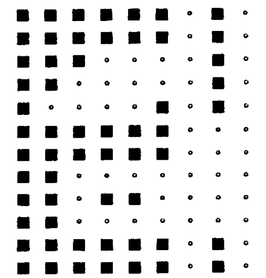
6 次循環之後



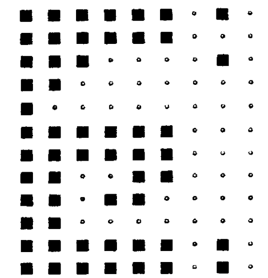
12 次循環之後



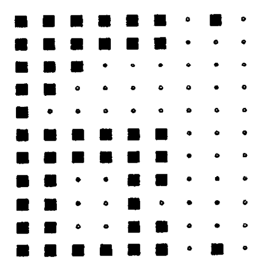
18 次循環之後



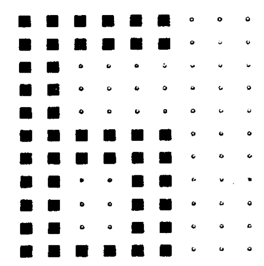
24 次循環之後



30 次循環之後



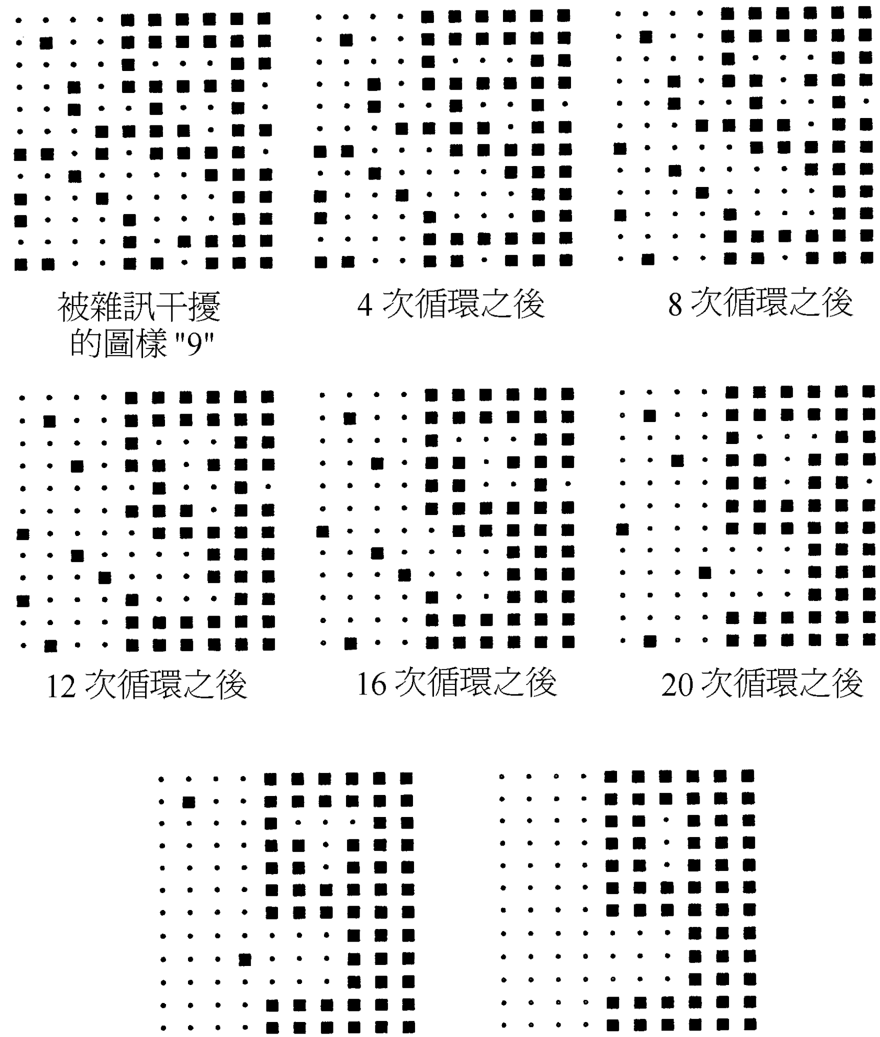
36 次循環之後



41 次循環之後

範例資料 2 部份錯誤

回想：



被雜訊干擾的圖樣 "9"　4 次循環之後　8 次循環之後

12 次循環之後　16 次循環之後　20 次循環之後

24 次循環之後　28 次循環之後

回想過程的變化

- 在回想過程的第4步驟，狀態變數更新的方式有兩種：
  - 串列式
  - 平行式
- 霍普菲爾網路一般是採用平行式。

串列式

- **狀態變數逐一更新。即：當一個變數更新後的值立即生效，可影響其它狀態變數更新。**
  - **若 $net_j > 0$，則處理單元 $X_j = 1$。**
  - **若 $net_j = 0$，則處理單元 X_j = 原本處理單元 X_j 之值。**
  - **若 $net_j < 0$，則處理單元 $X_j = -1$。**

新的X：

$$\begin{bmatrix} net_1 \\ net_2 \\ net_3 \\ net_4 \\ net_5 \\ net_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 & 0 & 4 \\ 4 & 0 & 0 & 0 & 4 & 0 \\ 0 & 4 & 0 & 0 & 0 & 4 \\ 4 & 0 & 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} +1 \\ \cancel{+1} \\ +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} \begin{array}{c} \\ -1 \\ \\ \\ \\ \\ \end{array} = \begin{bmatrix} +8 \\ -8 \\ \\ \\ \\ \\ \end{bmatrix} \Rightarrow \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix} = \begin{bmatrix} +1 \\ -1 \\ \\ \\ \\ \\ \end{bmatrix}$$

- 平行式
  - 狀態變數一起更新。即：當一個變數更新後的值不會立即生效，不影響其它狀態變數的更新。使用一組暫時變數 new_X_i。
    - 若 $net_j > 0$，則 $new\_X_i = 1$。
    - 若 $net_j = 0$，則 $new\_X_i =$ 原處理單元 X_j。
    - 若 $net_j < 0$，則 $new\_X_i = -1$。
  - 利用暫時變數，同時對每一個狀態變數做更新：$Xi \leftarrow new\_X_i$

$$\begin{bmatrix} net_1 \\ net_2 \\ net_3 \\ net_4 \\ net_5 \\ net_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 & 0 & 4 \\ 4 & 0 & 0 & 0 & 4 & 0 \\ 0 & 4 & 0 & 0 & 0 & 4 \\ 4 & 0 & 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} +1 \\ +1 \\ +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} +8 \\ -8 \\ +8 \\ 0 \\ +8 \\ 0 \end{bmatrix} \Rightarrow \overset{new\_X_i}{\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix}} = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \\ +1 \\ -1 \end{bmatrix}$$

■結論

- **記憶容量的限制**
  - 理想上，若有 n 個處理單元的霍普菲爾網路，它所能夠表示與處理的所有可能圖樣有 2^n個，例如：n = 100，$2^n \fallingdotseq 10^{30}$。然而，它所能夠記憶與聯想的樣式數量遠低於此數。
    - 霍普菲爾本人於1982年以實驗得知此網路的記憶容量極限為 **0.15n**。而其他學者從理論探討，得知記憶容量為 **$n/(4\times\log_2 n)$**。
    - 例：當 n = 1000，理想量 = 2^{1000}，霍式實驗量 = 150，其他學者得出的記憶量 = 25。
    - 這是因為 1000個處理單元需要**100萬根連結**!!!
  - 若允許**部份不完美的聯想**，則記憶容量可以提高。
- **局部最小值問題**
  - 由於霍普菲爾網路在處理自聯想問題時，還是以**梯度下降法**對能量函數 (即：李亞普諾夫函數) 做最佳化處理，因此無法保証所找到的解一定的全域最佳解。