Self-Organizing Neural Networks for Data Projection

Mu-Chun Su 蘇木春

Department of Computer Science & Information Engineering

National central University, Taiwan, R.O.C.

Outline

- Introduction
- Review of the SOM Algorithm
- The New Model of Self-Organizing Neural Networks
- Simulation Results
- Conclusions

Introduction (1)

- **Tool: Data projection algorithms**

  A. Conventional Approaches:

  Usually analysts choose a projection algorithm to project data and then explore the intrinsic dimensionality and analyze the clustering tendency of high-dimensional data by viewing the scatter plot of the projected data

  (1) Local minima problem:

  (2) Parameter selection problem:

  (3) Display problem

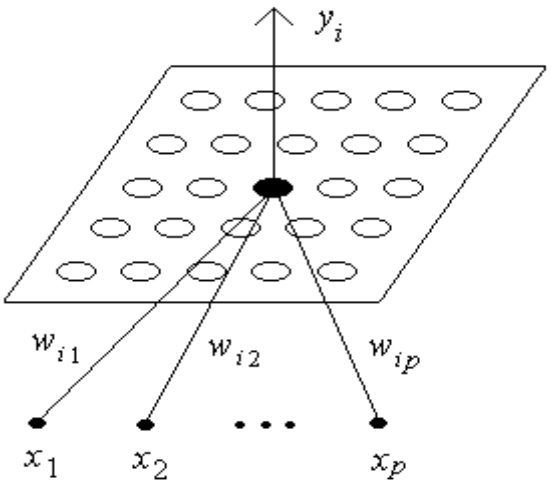
Introduction (2)

B. Our Approach:

- **Use of the SOM algorithm to project data**
- **Display a sequence of intermediate scatter plots during the projection process**

Review of the SOM Algorithm (1)

- **Appealing Properties:**
  (1) preserving topological property
  (2) easy to implement



Review of the SOM Algorithm (2)

- **Step 1: Initialization:** Choose random values for the initial weights $\underline{w}_j$.
- **Step 2: Winner Finding:** Find the winning neuron at time k, using the minimum-distance Euclidean criterion:

$$j^* = \arg\min_j \left\| \underline{x}(k) - \underline{w}_j \right\|, \quad j = 1, \cdots, N^2.$$

where represents the input pattern, is the total number of neurons, and indicates the Euclidean norm.

Review of the SOM Algorithm (3)

- **Step 3: Weights Updating:** Adjust the weights of the winner and its neighbors, using the following rule

$$\underline{w}_j(k+1) = \underline{w}_j(k) + \eta(k)\Lambda_{j^*}(k)[\underline{x}(k) - \underline{w}_j(k)]$$

where

$$\Lambda_{j^*}(k) = \exp\left(-\frac{d_{j^*,j}^2}{2\sigma^2}\right)$$

Simulations of SOM Algorithm (1)

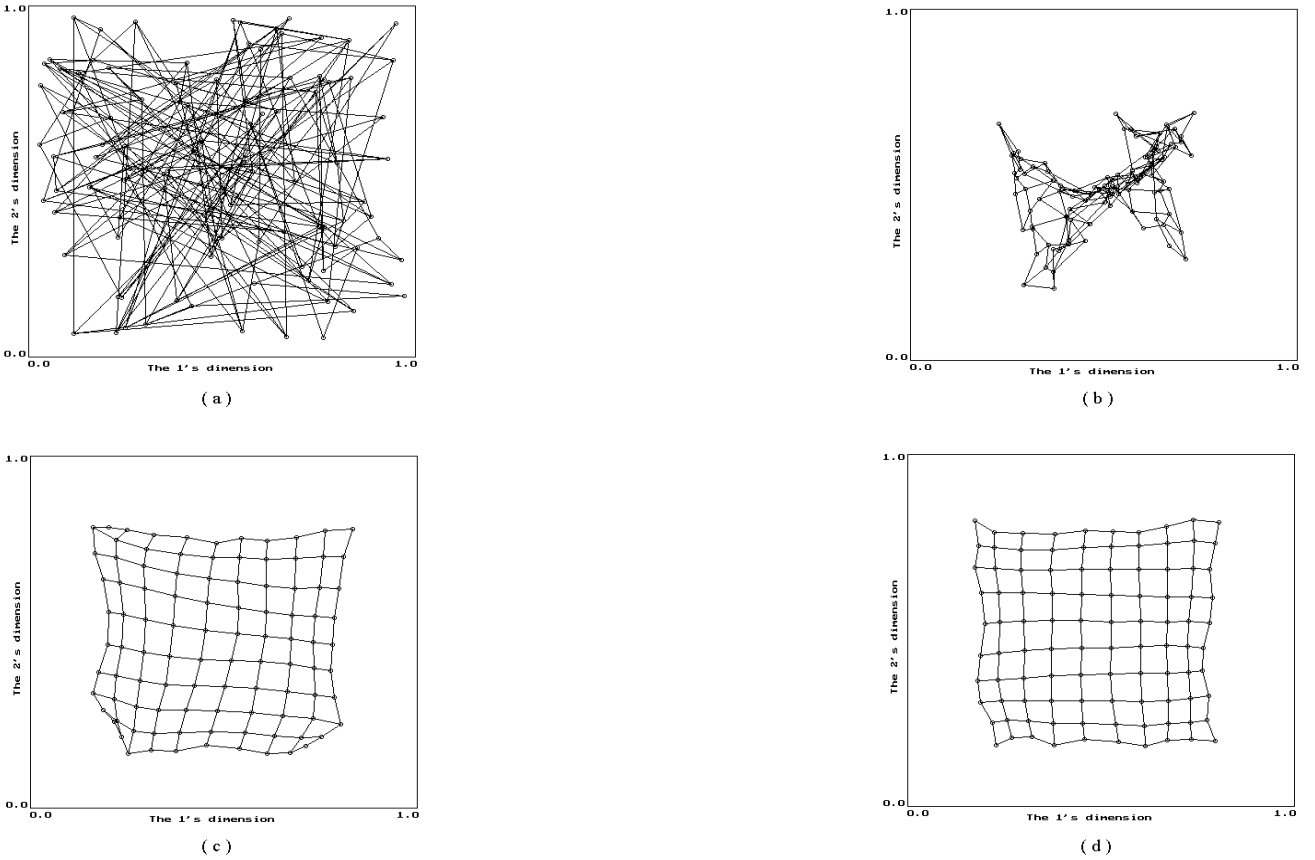
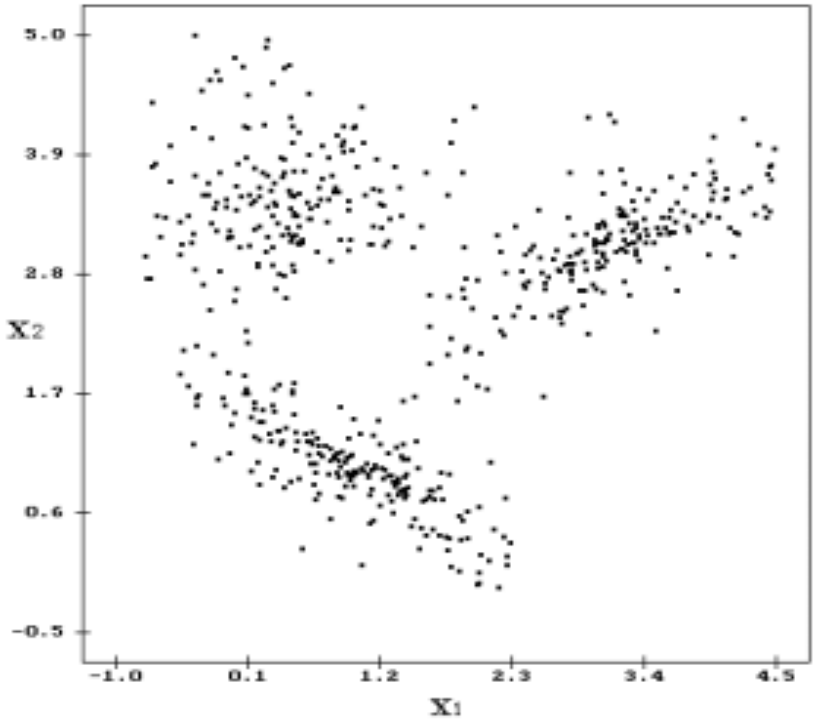
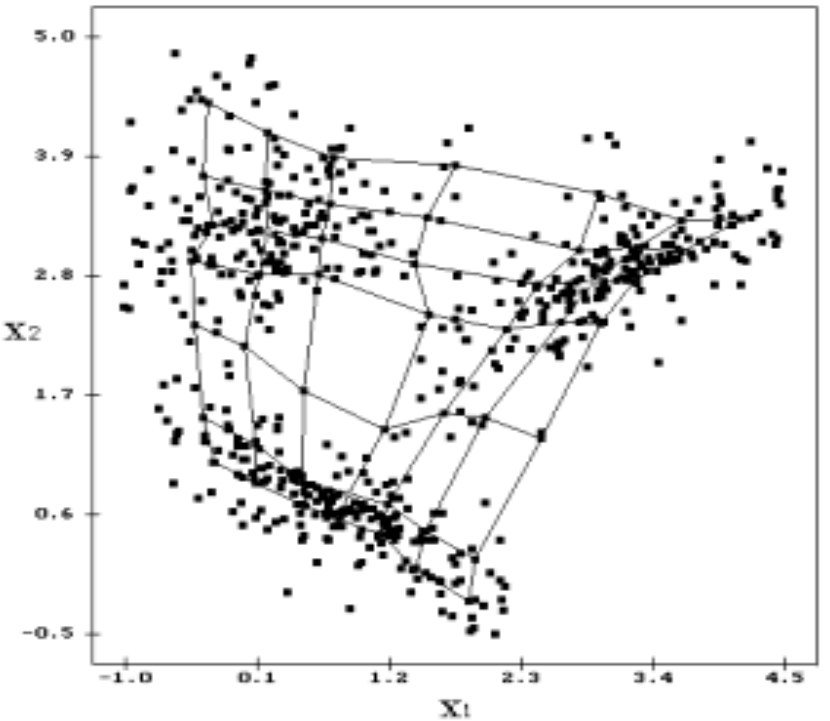


Fig.1 Simulation of SOM algorithm: (a) initial map; (b) after 50 iterations; (c) After 1,000 iterations; (d) after 10,000 iterations

Simulations of SOM Algorithm (2)





Simulations of SOM Algorithm (3)

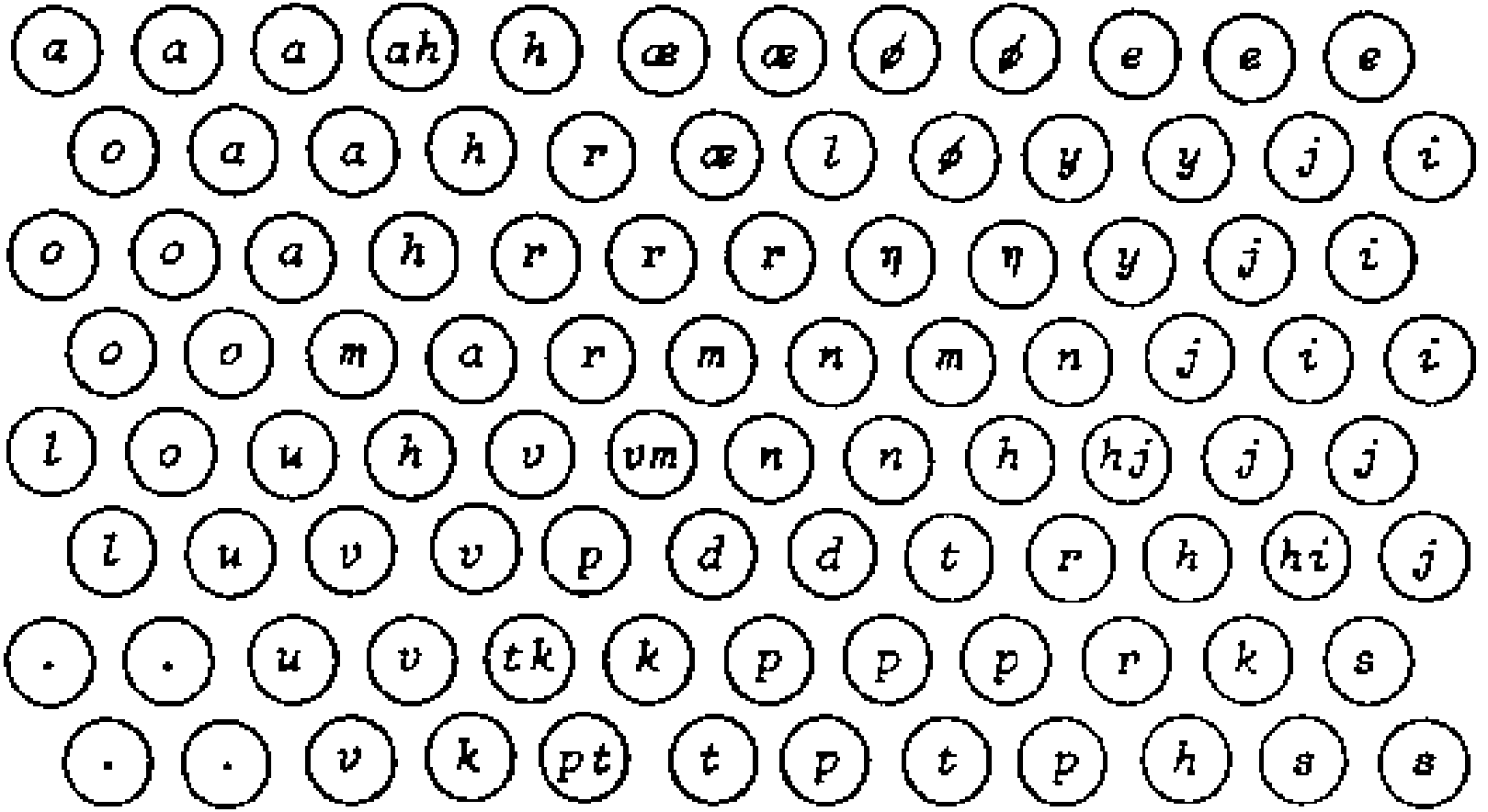


Fig. 3 The phoneme map.

Interpretation Methods

- 累積響應法
- DSOM Algorithm
- U-matrix Method
- Kraaijveld et al

累積響應法 (1)

1. 形成特徵映射圖：將所有資料加以訓練，以形成自我組織特徵映射圖之拓樸順序。
2. 累積響應：將每個神經元對所有輸入資料之輸出響應加以累積，而累積方式如下

$$h_j = \sum_{k=1}^{M} Out_j(\underline{x}_k), \quad j = 1,2,\cdots,N^2$$

$$Out_j(\underline{x}_k) = e^{-\left\|\underline{x}_k - \underline{w}_j\right\|^2}$$

在此，M為輸入資料的總數，為神經元*j*所累積的總反應，而則為神經元總數，則代表輸入資料對神經元*j*之輸出響應。

累積響應法 (2)

- 尋找響應尖峰：

  (1) 我們可以將所有神經元之累積響應正規化成0~255後，視為N×N的數位影像之像素灰度值，而此N×N的數位影像矩陣與N×N的神經元矩陣剛好每個元素相對應。

  (2) 每個數位影像上的灰階度恰與神經元之累積輸出響應成反比，亦即響應越大，影像越趨近於黑色，而響應越小則趨近於白色。

  (3) 而群聚的中心，因其累積輸出響應較周圍大，所以大略的中心位置，可以由與周圍像素相較為黑者之對應神經元鍵結值找出，亦即累積輸出響應較大者。

累積響應法 (3)

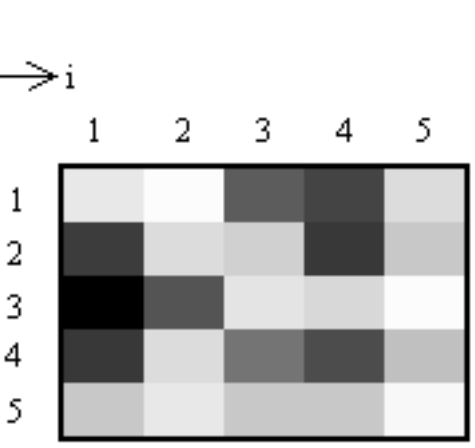
- 我們由數位影像矩陣的左上角，由左至右，由上往下，每個元素皆透過此3×3遮罩來與周圍相比較，令p_0為每次搜尋的該元素，而$p_1,\ldots,p_8$則為p_0之8鄰居，若p_0滿足下述：

$$h_{p_0} > h_{p_i}, \quad i = 1,2,\cdots,8$$

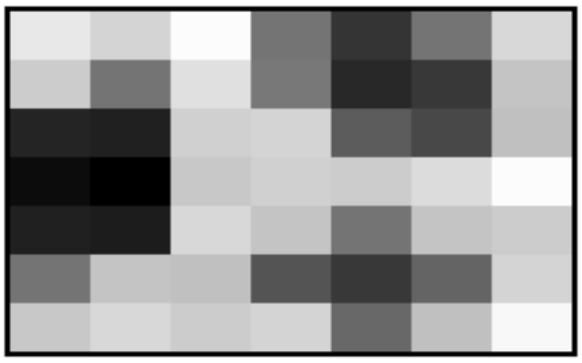
則可發現p_0存在一累積響應尖峰，亦即靠近群聚中心點的類神經元所在。

模擬結果 (1)

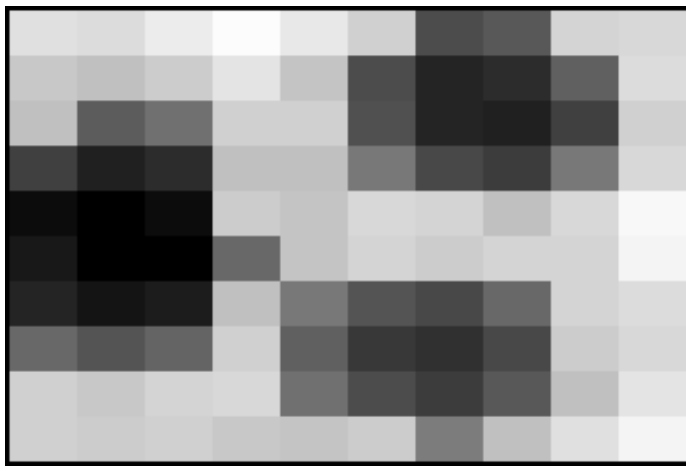
二維資料集



(a)



(b)



(c)

圖5.18 579筆二維資料集之累積響應二維影像對照圖 (a) 5x5; (b) 7x7;(c)10×10。

模擬結果 (2)

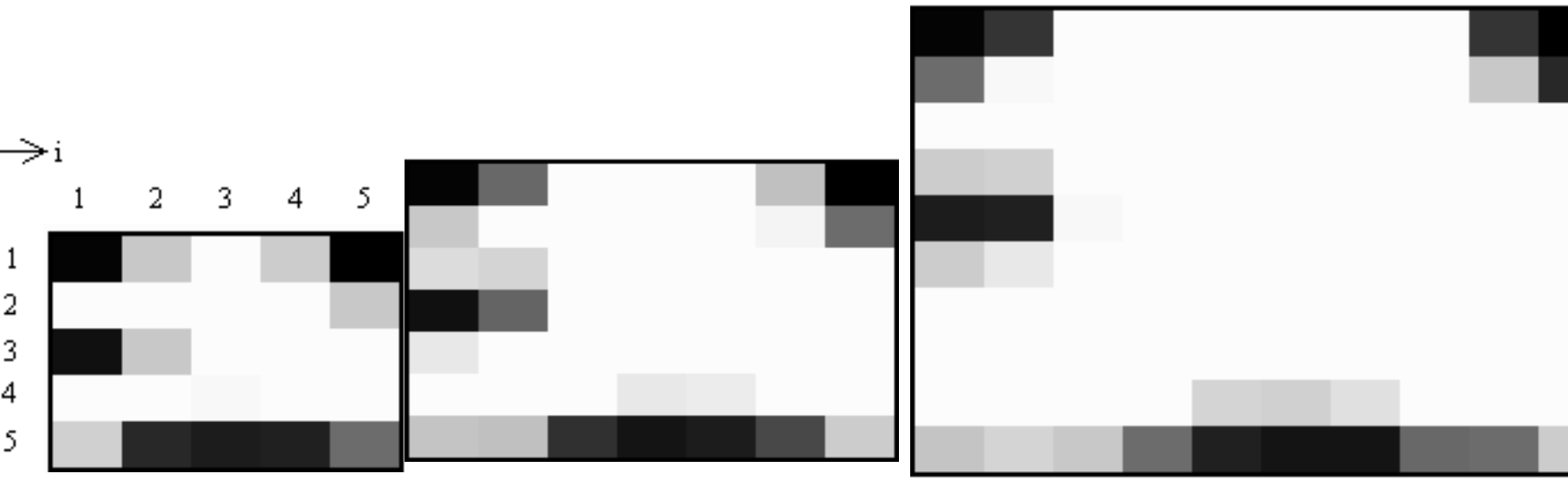
- 三維資料集之測試

(a) (b) (c)

圖5.1 三維空間中200筆資料之累積響應二維影像對照圖(a) 5x5; (b) 7x7;(c)10×10。

模擬結果 (3)

- 十維資料集之測試

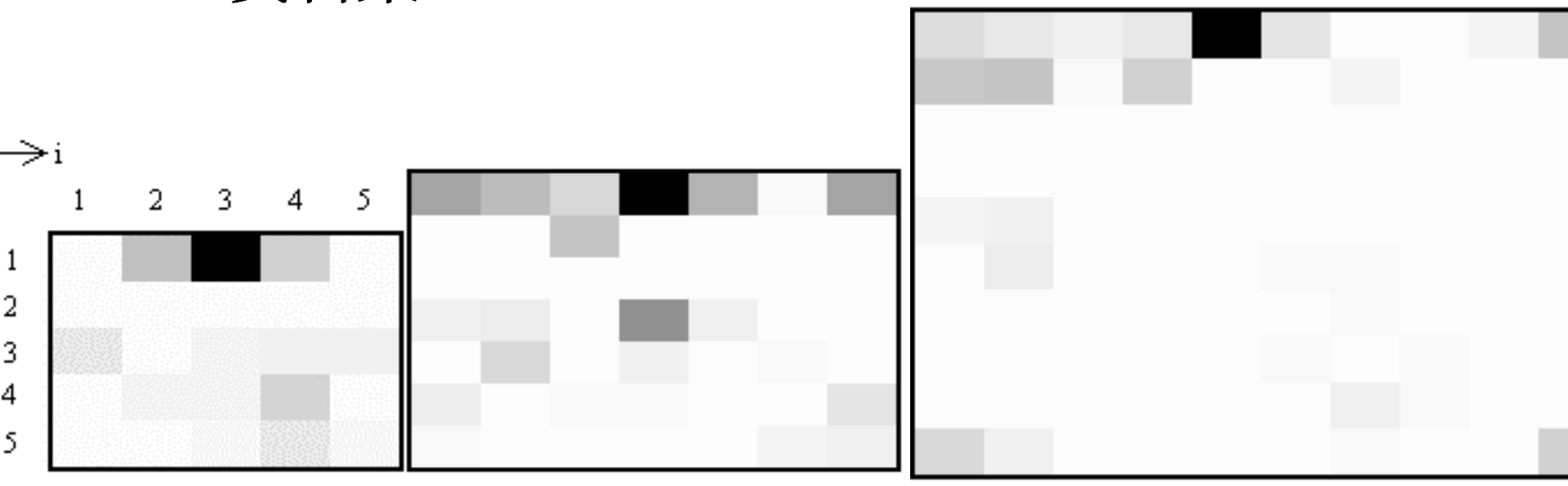


(a) (b) (c)

圖5.14 十維空間中200筆資料之累積響應二維影像對照圖(a) 5x5; (b) 7x7;(c)10×10。

模擬結果 (4)

- ***Iris*資料集**



(a) (b) (c)

圖5.20(a) Iris資料集的自我組織特徵映射經由解讀後之累積響應所對應之二維影像 (a) 5x5; (b) 7x7;(c)10×10。

The New Model of Self-Organizing Neural Networks (1)

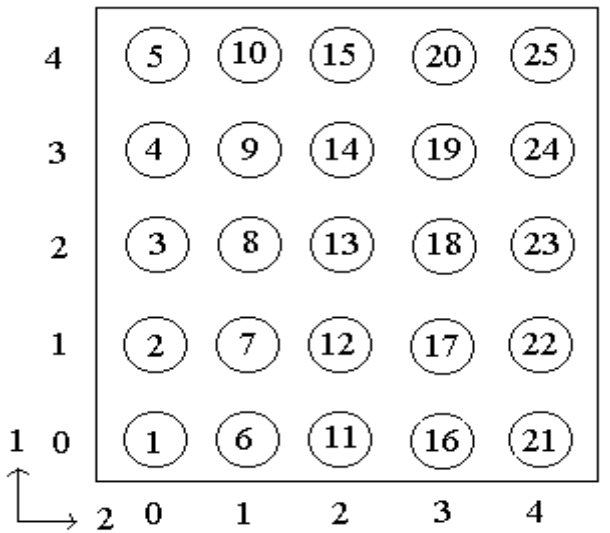
- The predetermined rigid structure of neural networks implies limitations on the data visualization and projections.

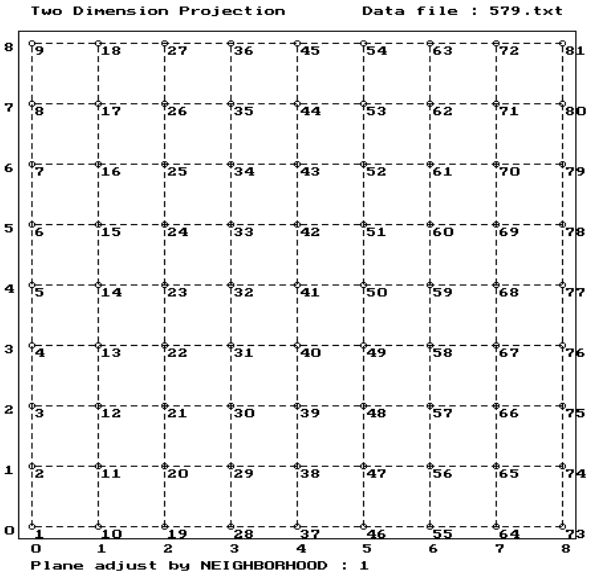
Solution: To adjust the architecture of the network during a self-organization process.

Network Architecture (1)

- The initial topology of the network is a rectangular grid. Each neuron j has an n-dimensional synaptic weight vector $\underline{w}_j$ attached. In addition to the vector, another two-dimensional position vector $\underline{p}_j$ is also attached to neuron j. The vector can be regarded as the location of neuron j in the continuous output space.

Network Architecture (2)





Network Dynamics (1)

- During the self-organization process, not only the synaptic weight vectors but also the position vectors are adapted. The whole adaptation procedure referred to as the “ Double Self-Organizing Feature Map “ (**DSOM**) algorithm is summarized as follows
- **Step 1: Present an input pattern into the network.**

Network Dynamics (2)

- **Step 2: Locate the best matching neuron using the minimum-distance Euclidean criterion:**

$$j^* = \arg\min_j \left\| \underline{x}(k) - \underline{w}_j \right\|, \quad j = 1, \cdots, N^2. \tag{1}$$

- **Step 3: Update the synaptic vectors 's using Eqs. (2)-(4)**

$$\underline{w}_j(k+1) = \underline{w}_j(k) + \eta(k) \Lambda_{j^*}(k) [\underline{x}(k) - \underline{w}_j(k)] \tag{2}$$

$$\eta(k) = \eta_0 \frac{1}{k+1} \tag{3}$$

$$\Lambda_{j^*}(k) = \exp[-s_w (1 + \frac{k}{k_{\max}}) \left\| \underline{p}_j(k) - \underline{p}_{j^*}(k) \right\|^2] \tag{4}$$

Network Dynamics (3)

- **Step 4: Update the position vectors ’s using Eqs. (5)-(7)**

$$\underline{p}_j(k+1) = \underline{p}_j(k) + \eta(k) h_{j^*}(k)[\underline{p}_j(k) - \underline{p}_{j^*}(k)] \quad (5)$$

$$\eta(k) = \eta_0 \frac{1}{1+k} \quad (6)$$

$$h_{j^*}(k) = \exp[-s_w(1+\frac{k}{k_{\max}})\left\|\underline{p}_j(k) - \underline{p}_{j^*}(k)\right\|] \quad (7)$$
$$\times \exp\{-s_x[\left\|\underline{w}_j(k) - \underline{x}(k)\right\| - \left\|\underline{w}_{j^*}(k) - \underline{x}(k)\right\|]^2\}$$

- **Step 5: Go to step 1 until it converges or some terminating criteria is satisfied.**

Network Dynamics (4)

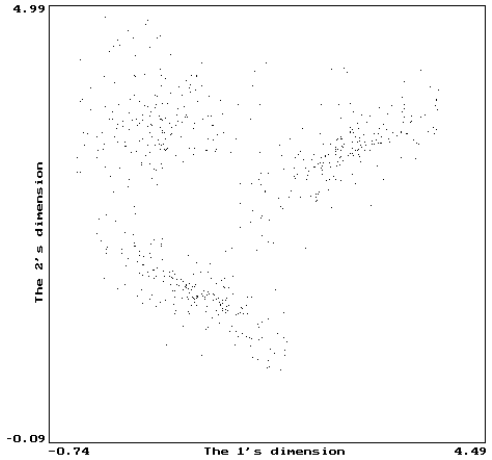
- Here S_w and S_x are two predetermined scalar parameters. This formula tells us that

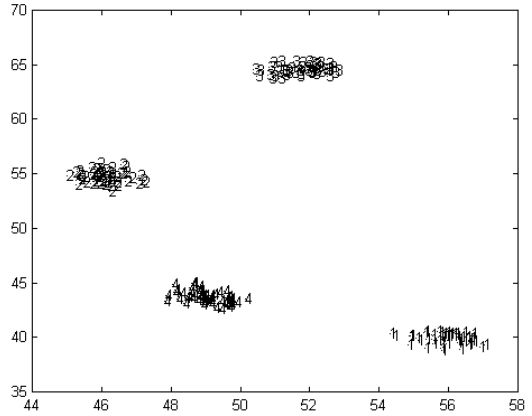
  (1) If neurons have similar responses as the winner neuron *j** (*i.e.* $\|\underline{w}_j(k) - \underline{x}(k)\| \approx \|\underline{w}_{j^*}(k) - \underline{x}(k)\|$) then their position vectors's $\underline{p}_j(k)$ should be adjusted to move near to the winner's position vector $\underline{p}_{j^*}(k)$.

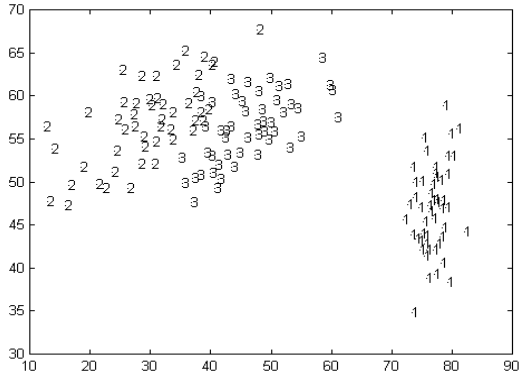
  (2) If neurons that are closer to the winner (*i.e.* $\underline{p}_j(k) \approx \underline{p}_{j^*}(k)$) then the updating step should be larger.

  (3) In addition, the weighting function is chosen as a decreasing function of iterations.

Simulation Results (1)







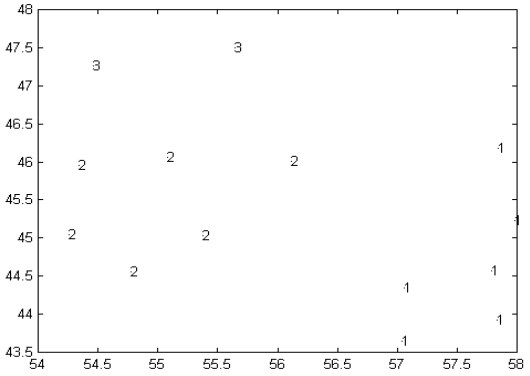


Figure 9: The projection results obtained by the Sammon's projection algorithm: (a) the 3D data set; (b) the iris data set; (c) the animal data set

Simulation Results (2)

- **Example 1: 2-dimensional artificial data set**

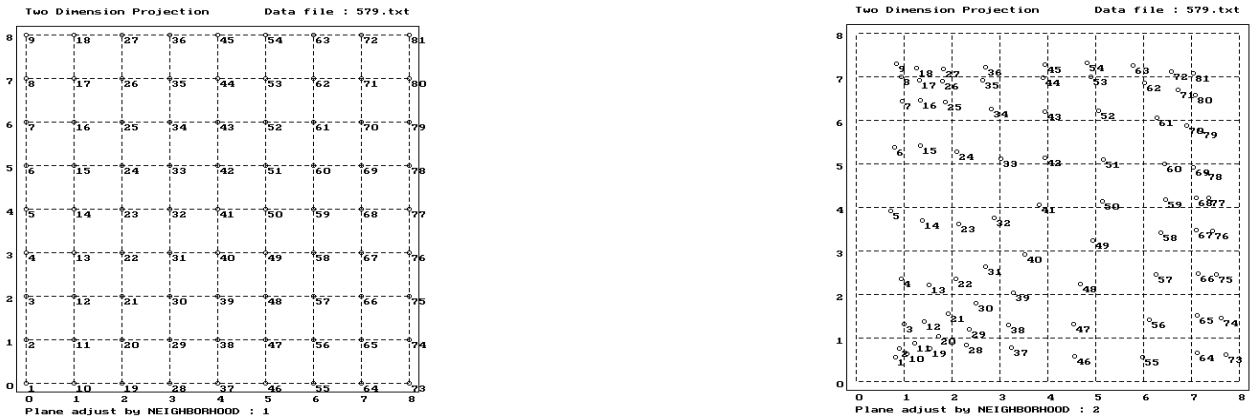
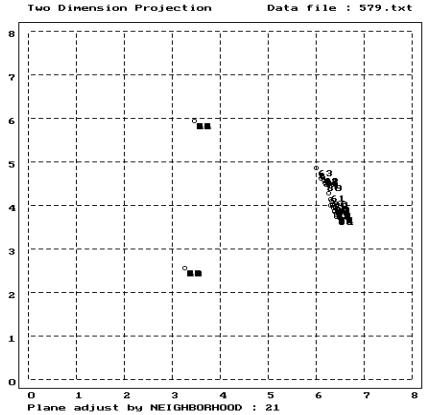
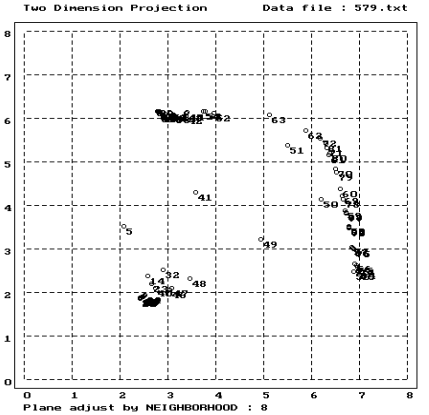
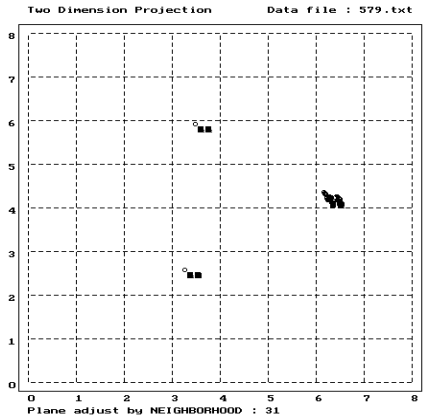


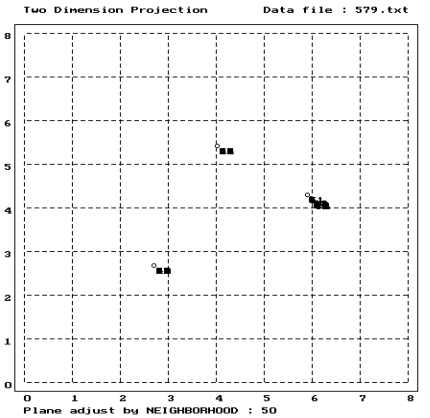
Figure 4: The training sequences for the 2D data set: (a) Initial position vectors. (b) Position vectors after 2 epochs. (c) Position vectors after 8 epochs. (d) Position vectors after 21 epochs. (e) Position vectors after 31 epochs. (f) Position vectors after 50 epochs.

Simulation Results (3)









Simulation Results (4)

- **Example 2: 3-dimensional artifical data set**

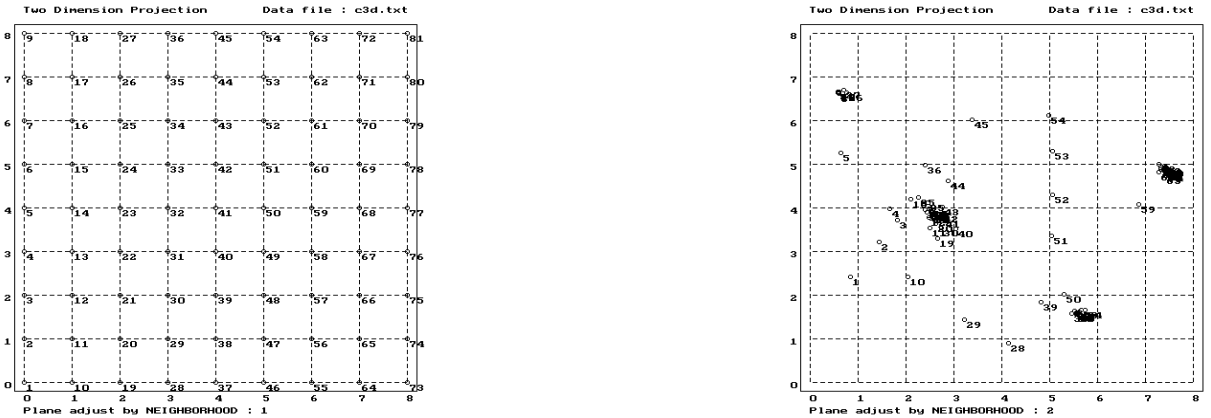
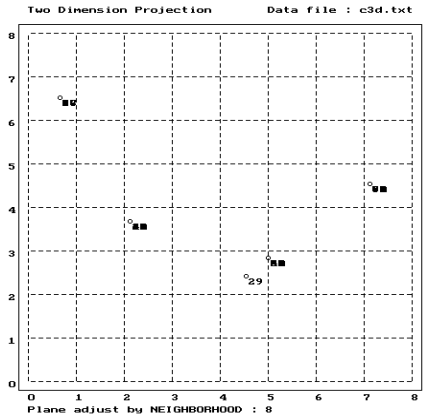
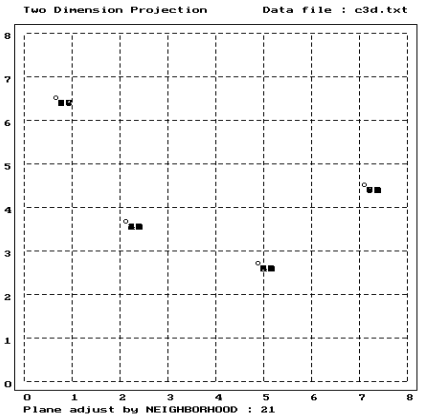
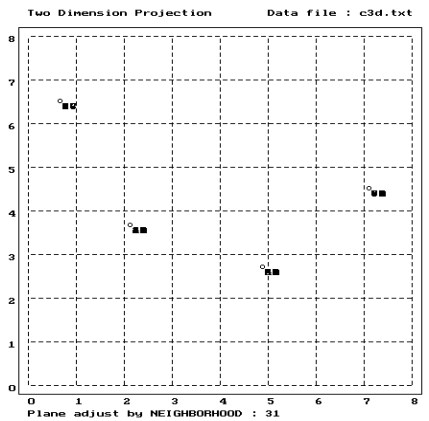


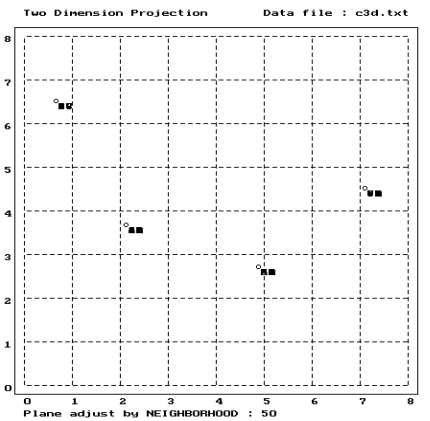
Figure 5: The training sequence for the 3D data set: (a) Initial position vectors. (b) Position vectors after 2 epochs. (c) Position vectors after 8 epochs. (d) Position vectors after 21 epochs. (e) Position vectors after 31 epochs. (f) Position vectors after 50 epochs.

Simulation Results (5)









Simulation Results (6)

- **Example 3: 4-dimensional iris data set**

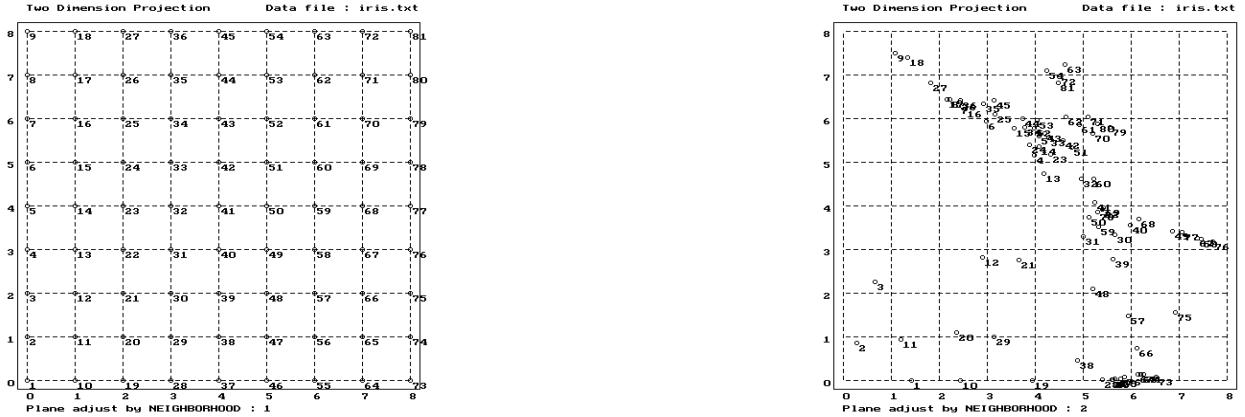
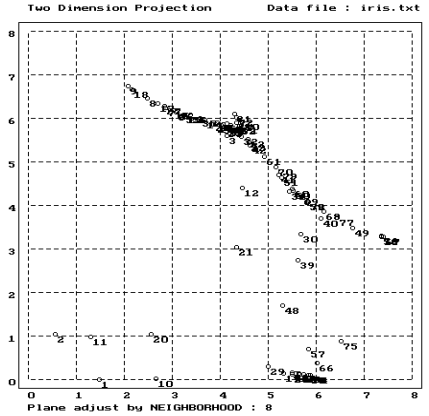
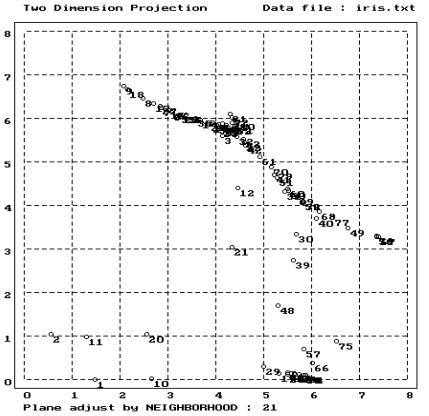
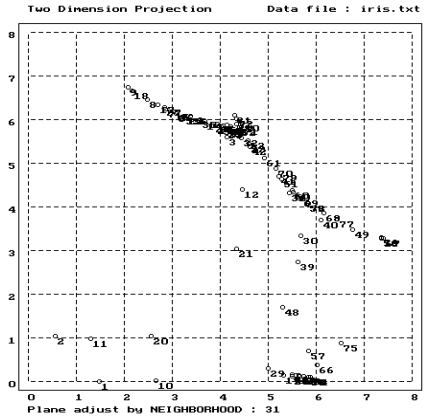


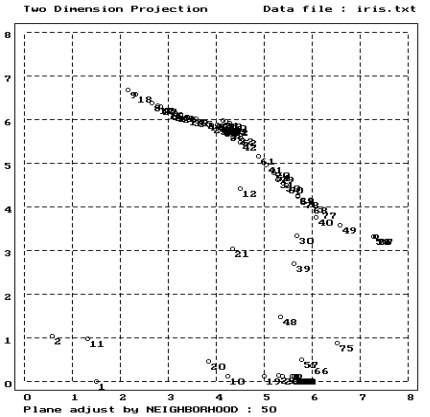
Figure 6: The training sequence for the iris data set: (a) Initial position vectors. (b) Position vectors after 2 epochs. (c) Position vectors after 8 epochs. (d) Position vectors after 21 epochs. (e) Position vectors after 31 epochs. Position vectors after 50 epochs.

Simulation Results (7)



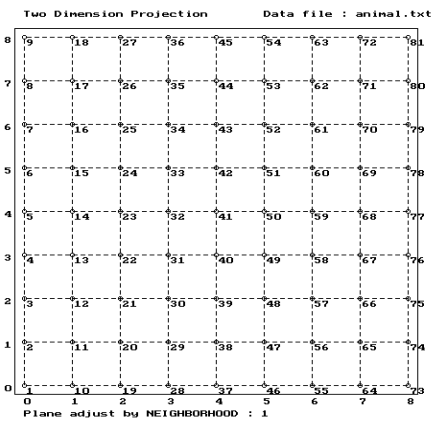






Simulation Results (8)

- **Example 4: 13-dimensional animal data set**



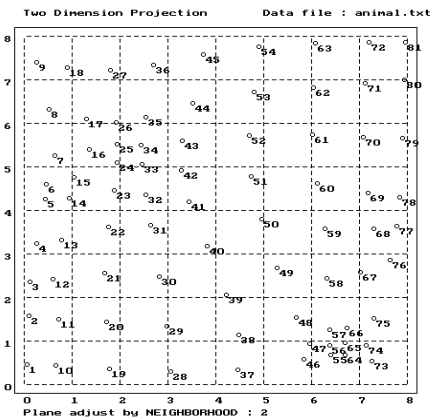
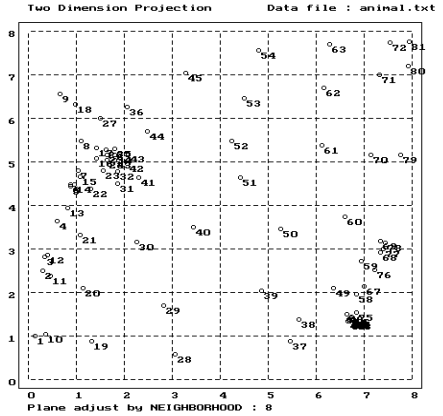
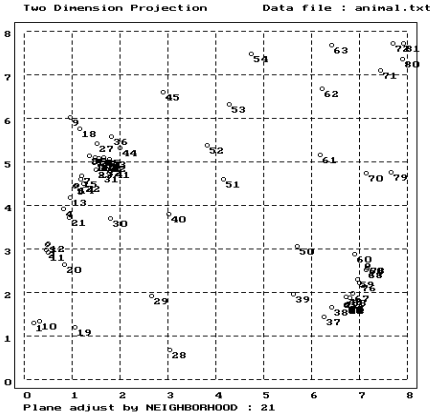
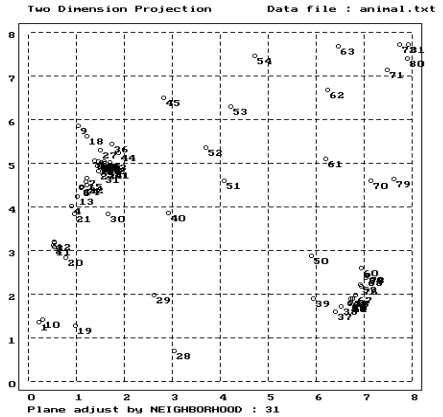


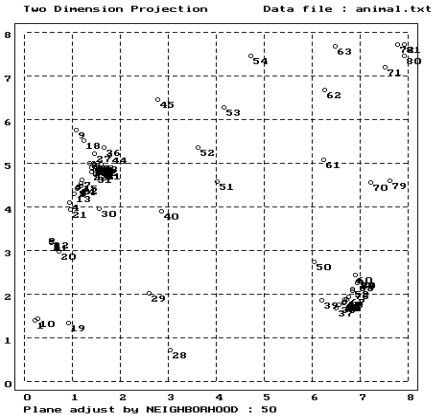
Figure 7: The training sequence for the animal data set: (a) Initial position vectors. (b) Position vectors after 2 epochs. (c) Position vectors after 8 epochs. (d) Position vectors after 21 epochs. (e) Position vectors after 31 epochs. (f) Position vectors after 50 epochs.

Simulation Results (9)





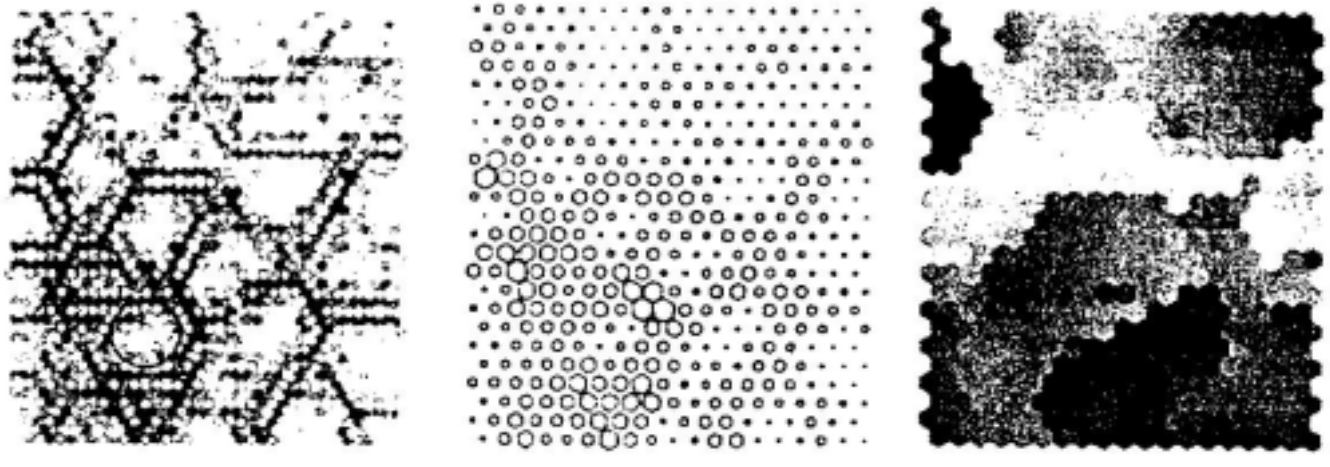




U-matrix Method (1)

- 1. Distances of each map neuron to each of its immediate neighbors are calculated and visualized using gray shade.
- The distances can also be visualized using the shape and size of each map neuron.
- Ref: A. Ultsh and H. P. Siemon, “Kohonen’s self organizing feature maps for exploratory data analysis,” Procd. Of Neural Network Conference (INNC’90), Springer, pp. 864-867, 1990. see http://www.cis.hut.fi/projects/ide/publications/

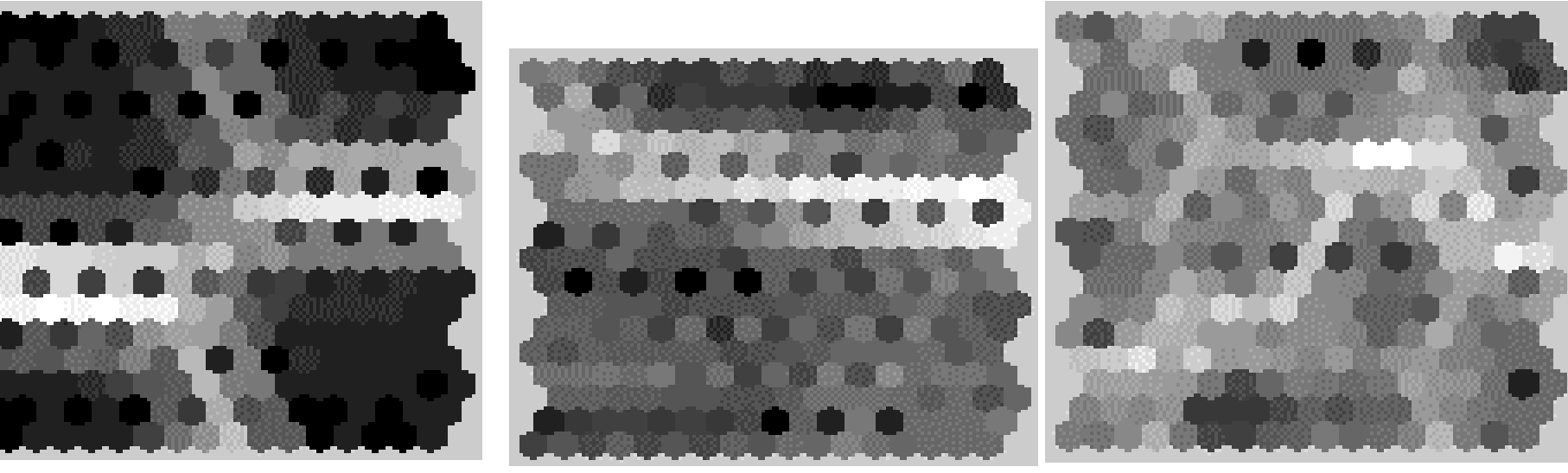
U-matrix Method (2)



(a) U-matrix (b) Distance matrix (c) Similarity coloring

Fig. 2. The cluster structure of the SOM is typically visualized using distance matrix techniques. (a) shows the U-matrix visualization. The white dots indicate locations of map units and hexagons between them show the actual values of the U-matrix. The gray shade of the hexagon denotes distance to the neighboring map unit. The darker the shade, the bigger the distance. Thus, clusters in the visualization can be seen as light areas with dark borders. There are several such clusters on the U-matrix. With further examination the characteristics of these clusters can be determined. For example, the area circled on the U-matrix could be named as 'Groundwood' cluster (cf. Fig. 7). (b) gives essentially the same information as (a). It is an averaged version of the U-matrix: the size of each map unit is proportional to its average distance to its neighbors. (c) illustrates the similarity visualization of the map: areas with similar hues are

U-matrix Method (3)



(a) (b) (c)

Figure 9: The projection results obtained by the U-matrix method: (a) the 3D data set; (b) the iris data set; (c) the animal data set

Display Method by Kraaijveld et al (1)

- Step1: A large network is chosen.
- Step 2: Every neuron corresponds to a pixel and the grey value of each pixel is determined by the maximum distance in the feature space of the corresponding unit to its four neighbors in the network. The larger the distance, the lighter the grey value is.
- Ref: M. A. Kraaijveld, J. Mac, and A. K. Jain, “A nonlinear projection method based on Kohonen’s topology preserving maps,” IEEE Trans. on Neural Networks, vol. 6, no. 3, pp. 548-559, 1995.

Display Method by Kraaijveld et al (2)

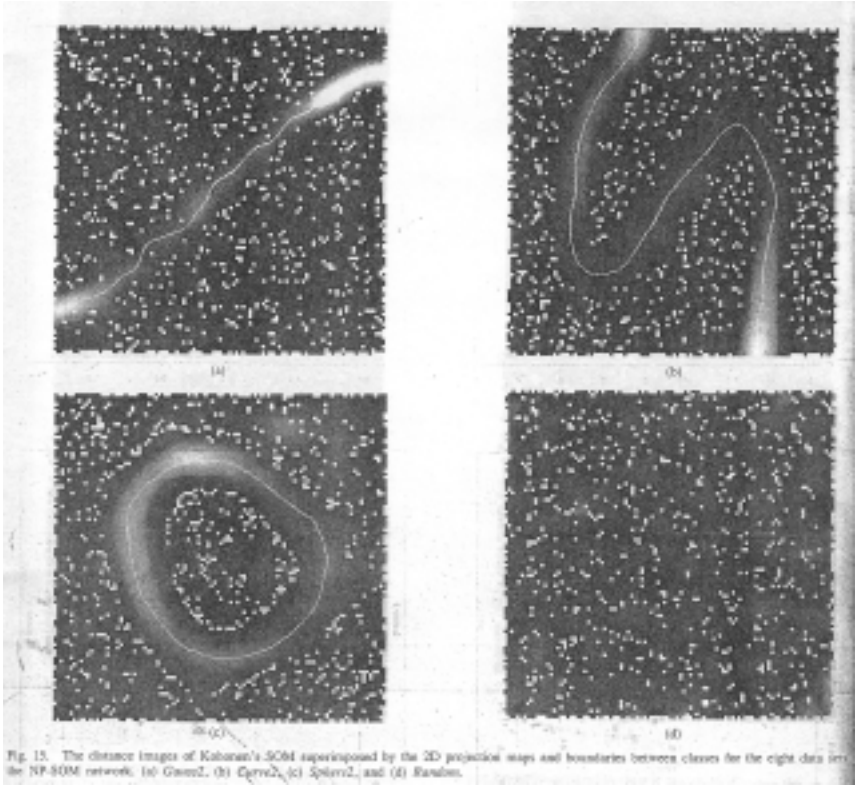


Fig. 15. The distance images of Kohonen's SOM superimposed by the 2D projection maps and boundaries between classes for the eight data sets the NP-SOM network. (a) Gauss2, (b) Curve2, (c) Spirrs2, and (d) Random.