



KHAI PHÁ DỮ LIỆU

Bài 8

Gom cụm bằng mạng Kohonen



Mai Xuân Hùng

Nội dung



- **Giới thiệu**
- **Ứng dụng mạng Kohonen để gom cụm dữ liệu**
- **Thuật toán gom cụm bằng mạng Kohonen**
- **Bài tập**



Giới thiệu



- Học có giám sát
- Học không giám sát



Học có giám sát



- Là kỹ thuật học sử dụng cho các bài toán phân lớp (Classification)
- Để thực hiện được bài toán trên, trước tiên cần phải có 2 điều kiện:
 - Điều kiện 1: phải biết trước số nhãn lớp cần phân loại, tức là phải biết trong công-ten-nơ đó có những loại quả gì. Giả sử trong công-ten-nơ đó có 5 loại quả là xoài, cam, táo, ổi, đào (đây chính là 5 loại nhãn lớp).
 - Điều kiện 2: phải có tập đặc trưng của mỗi loại quả, ví dụ các đặc trưng là: hình dáng, màu sắc, trọng lượng, độ cứng mềm, v.v... Tập đặc trưng này có được thông qua học một tập dữ liệu huấn luyện (chính là các công-ten-nơ của các chuyên hàng trước đó)
- Khi thực hiện phân loại các loại quả trong công-ten-nơ đang xét, dựa vào đặc trưng của các loại quả (điều kiện 2), quả sẽ được đưa vào 1 trong 5 nhóm đã biết (điều kiện 1).

Học không giám sát



- Là kỹ thuật học sử dụng cho các bài toán phân cụm, gom cụm (Clustering)
- Để thực hiện được bài toán trên, cần phải có tập đặc trưng của mỗi loại quả. Tập đặc trưng này có được cũng thông qua học một tập dữ liệu huấn luyện (như điều kiện 2 của Học có giám sát).
- Điểm khác của Học không giám sát so với Học có giám sát là: trước khi phân cụm, không biết trong công-ten-nơ đang xét có bao nhiêu loại quả và đó là những loại quả gì.





Thuật toán gom cụm bằng mạng Kohonen là một kỹ thuật học không giám sát





Why cluster?

Nhu cầu

Ta cần gom cụm các đối tượng





Giới thiệu về mạng kohonen



- Mạng Kohonen được phát triển bởi Teuvo Kohonen vào năm 1980.
- Dùng để ứng dụng trong việc gom cụm phẳng các đối tượng
- Ưu điểm của Kohonen là không cần chỉ định trước số cụm



Kiến trúc cụm phẳng



- Cho tập đối tượng O, gom cụm phẳng là tiến trình gom các đối tượng thành các cụm (tập con của O) sao cho:
 - ❖ Các đối tượng trong cụm có mức độ tương tự cao
 - ❖ Các đối tượng trong các cụm khác nhau có mức độ tương tự thấp.
- Kết quả gom cụm phẳng sẽ tạo ra một phân hoạch tập đối tượng. Gọi C₁, C₂, ..., C_k là một kiến trúc cụm phẳng, các cụm thỏa các tính chất sau:
 - $\forall i, j \in [1, \dots, k]$, C_i \cap C_j = \emptyset



Ý tưởng chính của mạng Kohonen

- Tạo mảng hai chiều (map) với số dòng và số cột được khởi tạo trước. Trọng số của các phần tử (nơron) được khởi tạo ngẫu nhiên
- Trong quá trình học, dùng độ đo khoảng cách để tìm ra nơron chiến thắng
 - Nơron chiến thắng là nơron có khoảng cách tới mẫu học là nhỏ nhất
- Cập nhật trọng số của nơron chiến thắng và vùng lân cận của nơron chiến thắng
- Lặp đi lặp lại nhiều lần
- Kết quả ta thu được 1 ma trận 2 chiều có trọng số



Chi tiết thuật toán Kohonen



Bước 1: Khởi tạo ngẫu nhiên các trọng số trên lớp ra Kohonen và gán $Nc(t)$ là bán kính của vùng láng giềng. Khởi gán biến chu kỳ $t=1$

Bước 2: Đưa vào một mẫu học $v(t)$ và chuẩn hóa vector nhập $v(t)$. Tính khoảng cách Euclide từ vector nhập $v(t)$ đến tất cả các vector trọng của tất cả các nơron trên lớp ra Kohonen và chọn nơron có khoảng cách Euclide d_E nhỏ nhất từ vector học $v(t)$ đến trọng ứng với nút đó. Nơron ở nút này được gọi là nơron chiến thắng.

$$d_E(v, w_{ic\ jc}) = \min (d_E(v_i, w_{ij}))$$

Trong đó i, j là các chỉ số hợp lệ được xác lập theo kích thước của lớp ra Kohonen.



Chi tiết thuật toán Kohonen



Bước 3: Cập nhật các trọng số của các nút nằm trong vùng lân cận của nút chứa nơron chiến thắng (i_c, j_c) theo công thức:

$$w_{ijk}(t+1) = w_{ijk}(t) + [\frac{x_k(t)}{w_{ijk}(t)} * [x_k(t) - w_{ijk}(t)]]$$

Với $i_c - N_c(t) \leq i \leq i_c + N_c(t)$ và $j_c - N_c(t) \leq j \leq j_c + N_c(t)$

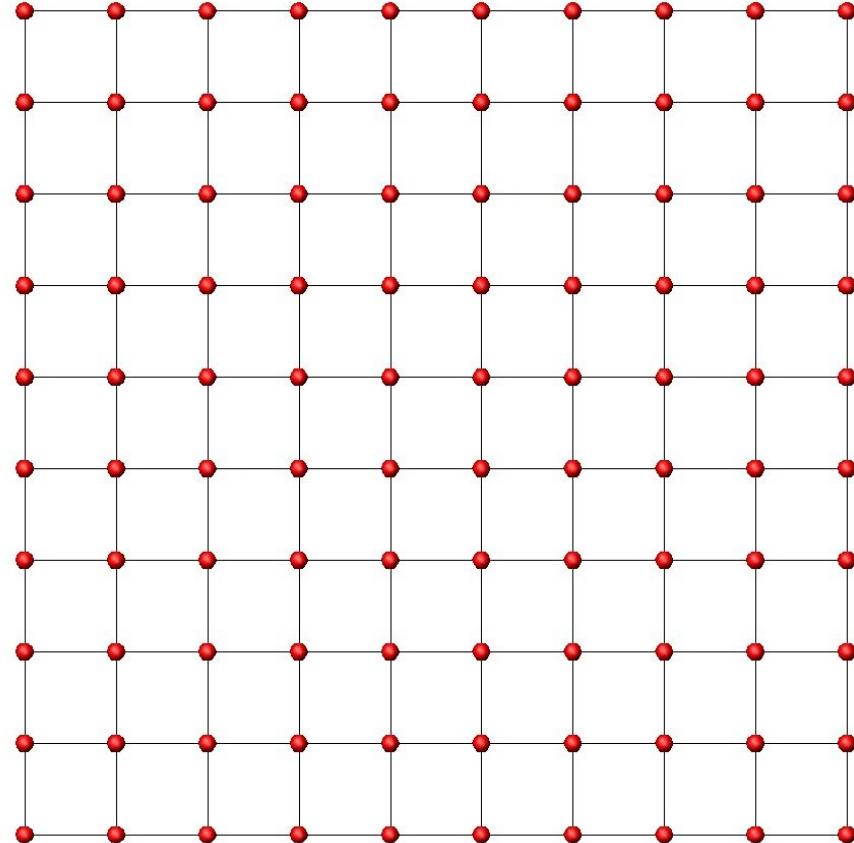
Bước 4: Cập nhật $t = t + 1$, đưa mẫu nhập kế tiếp vào mạng Kohonen và quay về **bước 2** cho đến khi đạt được điều kiện **hội tụ** hay vượt qua **số lần lặp** qui định.



Kohonen SOMS: chi tiết bước 1

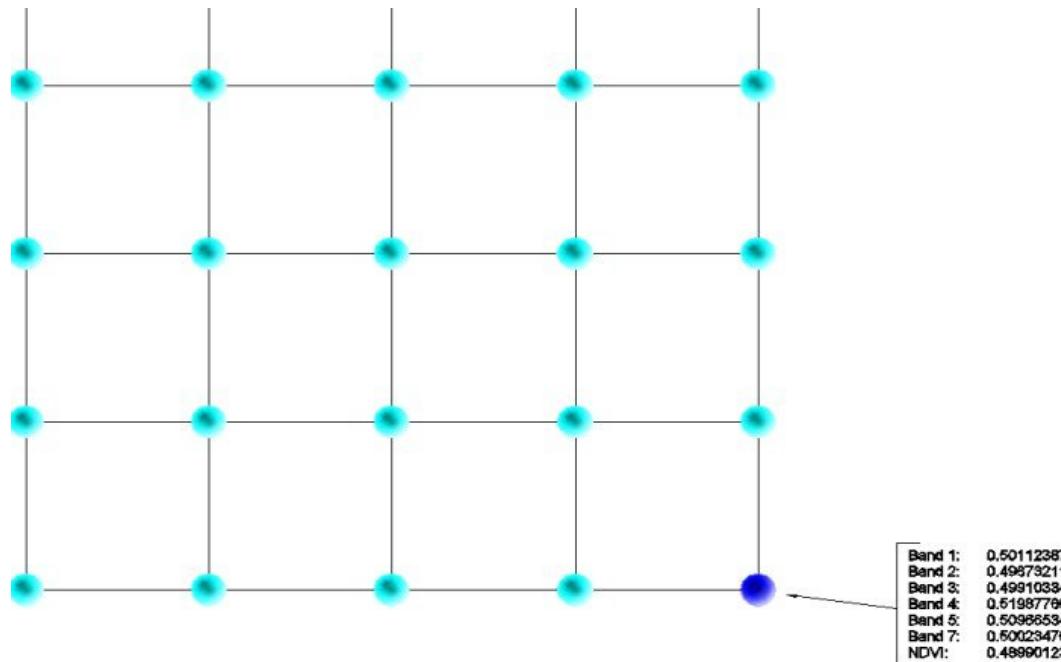


- ❖ Tạo mảng (map) hai chiều với số dòng = 10, số cột = 10



Kohonen SOMs: chi tiết bước 1

Mỗi thành phần của Nơron 0.500 +/- một giá trị nhỏ được phát sinh ngẫu nhiên



Kohonen SOMs: chi tiết bước 1



```
for (i = 0; i < X; i++)  
    for (j = 0; j < Y; j++)  
        for (k = 0; k < NumFeatures; k++)  
        {  
            KSOLayer[ i ][ j ][ k ] = 0.5;  
            val1 = rand() % 100;      // get 0 to 100  
            val1 -= 50.0;             // now get -50 to 50  
            val1 /= 500.0;            // finally get -0.10 to 0.10  
  
            val2 = rand() % 100;      // get 0 to 100  
            val2 -= 50.0;             // now get -50 to 50  
            val2 /= 500.0;            // finally get -0.10 to 0.10  
            KSOLayer[ i ][ j ][ k ] += (val1 * val2);  
        }  
    }
```



Kohonen SOMS: chi tiết bước 2



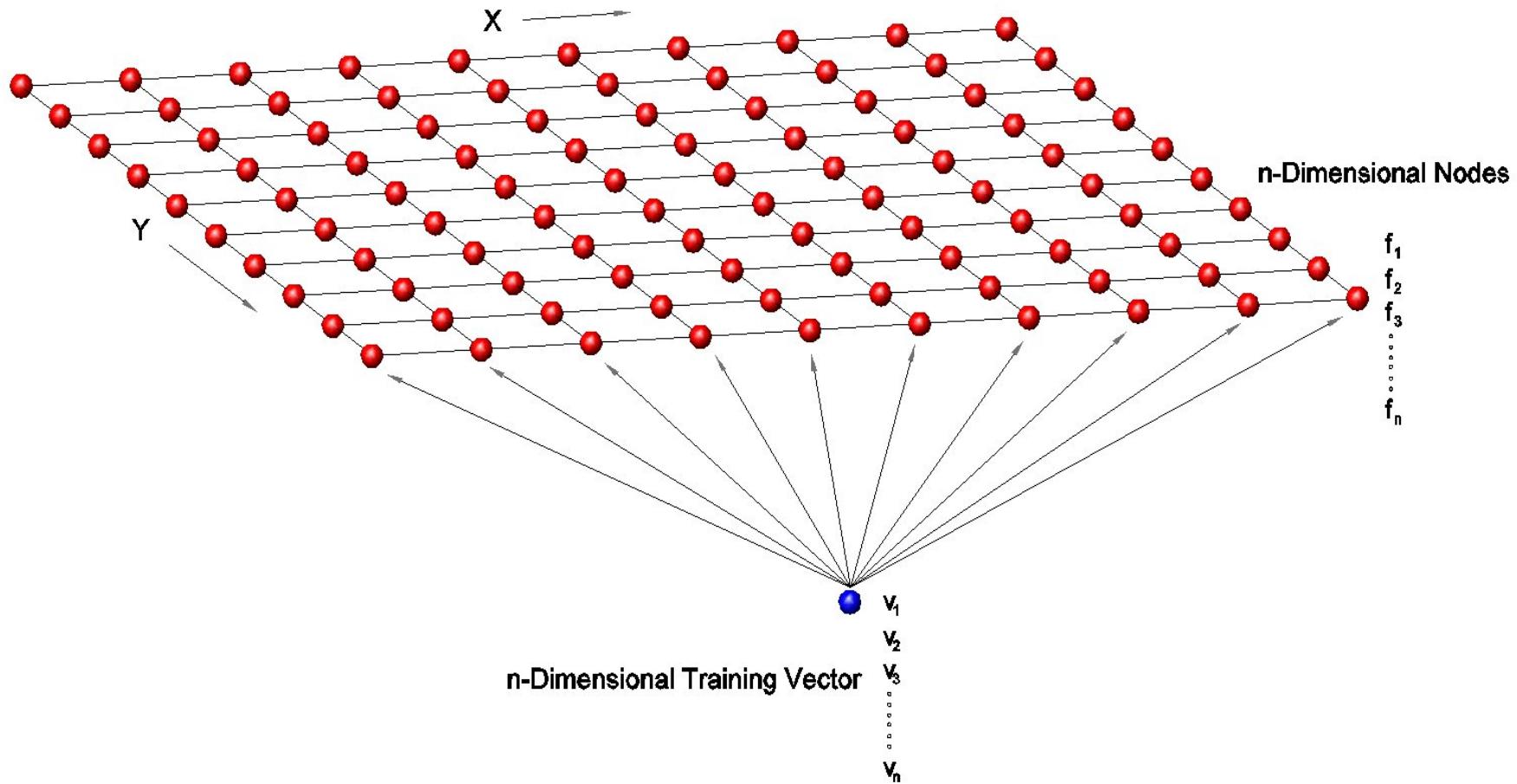
- Tính khoảng cách từ vector học đến từng Nơron của Map
- Tìm Nơron có khoảng cách nhỏ nhất từ Vector nhập đến Nơron đó. Nơron này gọi là Nơron chiến thắng



Tính khoảng cách từ vector học đến các Nơron



Vector Presentation to a KSOM



```

GetTrainVector();                                // get the vector to present
Smallest =10000;                            // default smallest value
SmallestPositionX = SmallestPositionY = 0;

for (x = 0; x < MapSizeX; x++)           // For each node in the map
    for (y = 0; y < MapSizeY; y++)
    {
        Distance = 0;

        for (z = 0; z < NumberFeatures; z++)
        {
            D = TrainVector[ z ] - KSOLayer[ x ][ y ][ z ];
            D *= D;
            Distance += D;
        }

        Distance = sqrt( Distance );

        if ( Distance <= Smallest)          // If new position has smallest distance
        {
            Smallest      = Distance; // Keep it and the new distance
            SmallestPositionX = x;
            SmallestPositionY = y;
        }
    }

UpdateWeights( SmallestPositionX, SmallestPositionY );

```



Kohonen SOMS: chi tiết bước 3



- Cập nhật trọng số của vùng lân cận Nơron chiến thắng bằng công thức:

$$w_{ijk} (t+1) = w_{ijk}(t) + [\boxed{W} * [x_k(t) - w_{ijk}(t)]]$$

trong đó

$w(t+1)$ Trọng số tại thời điểm $t+1$,

i, j : nơron tại dòng i cột j

$w(t)$ Trọng số tại thời điểm t

k Thành phần thứ k

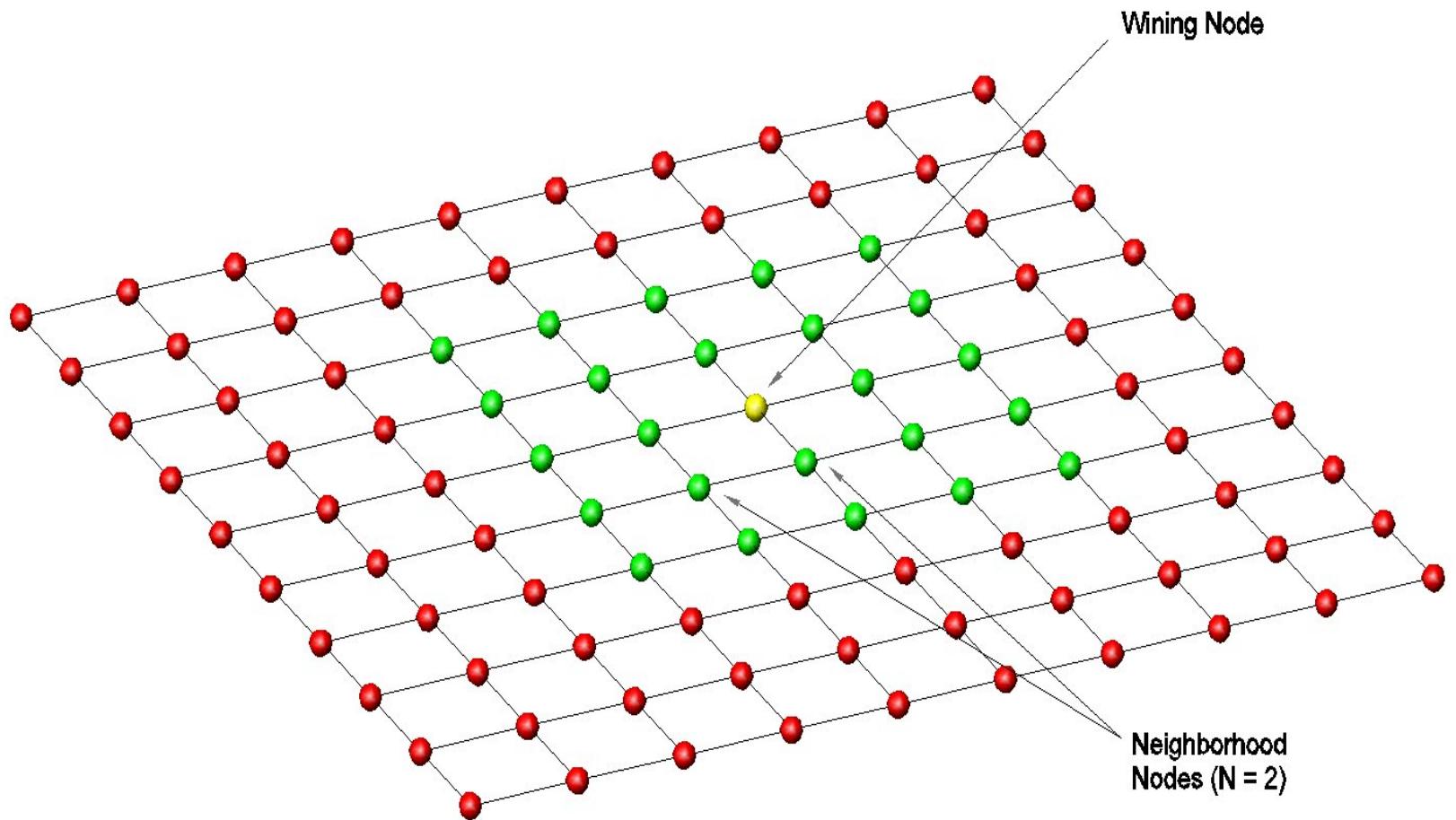
$x(t)$ Vector học x tại thời điểm t

\boxed{W} Hệ số $[0.01\dots 0.5]$





Kohonen Update Process



Kohonen SOMS: chi tiết bước 3



```
for (i = LowX; i <= HighX; i++)  
    for (j = LowY; j <= HighY; j++)  
        for (k = 0; k < NumFeatures; k++)  
        {  
            work = TrainVector[ k ] - KSOLayer[ i ][ j ][ k ];  
  
            KSOLayer[ i ][ j ][ k ] += Alpha * work;  
        }
```



Kohonen SOMs: cập nhật hệ số

Alpha = 0.2;

Alphalncr = Alpha / NumberTrainingEpochs;

TheHood = 10;

HoodDrop = 1000;

(Train loop starts here)

```
// Every epoch, update the gain constant using whatever strategy  
if (Alpha >= 0.01)  
    Alpha -= Alphalncr;
```

```
// Update the neighborhood
```

```
// Note: The (i + 1) accounts for epochs starting at 0  
if ( (Epochs % HoodDrop == 0) && (TheHood != 1) )  
    TheHood--;
```

(Train loop ends here)



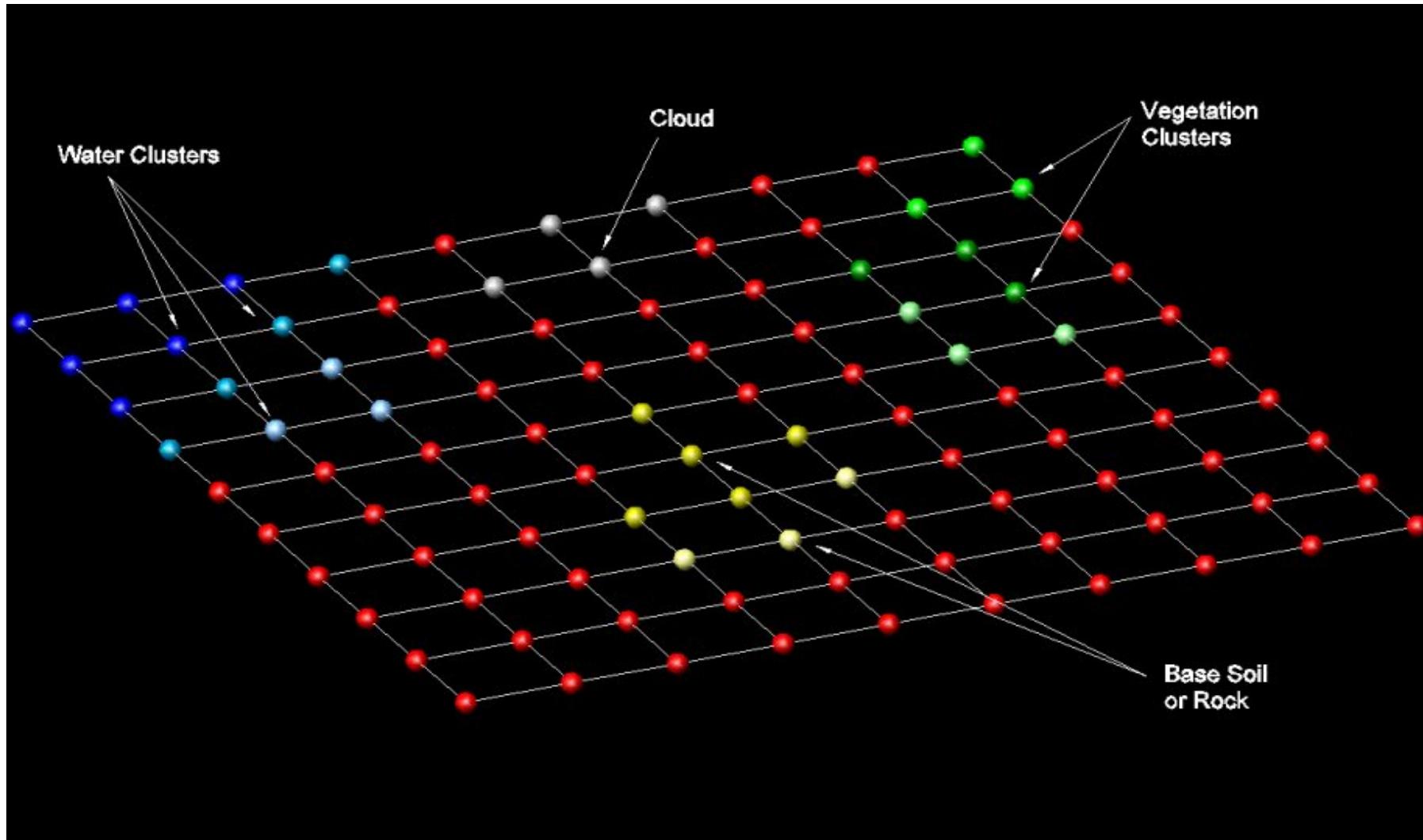
Kohonen SOMs: Kết quả



- ❖ Ta thu được một mảng 2 chiều (Map) các Nơron
- ❖ Sắp xếp các mẫu học vào Nơron có khoảng cách từ vector học đến Nơron đó là nhỏ nhất



Kết quả gom cụm bằng mạng Kohonen



Bài tập



- Cho tập điểm
 - $x_1=\{0.7, 0.45\}$, $x_2=\{2.8, 1\}$, $x_3 =\{2.6, 1\}$,
 $x_4=\{1, 0.8\}$, $x_5=\{2.5, 1.2\}$, $x_6=\{1.3, 1.4\}$
 $x_7=\{0.4, 0.7\}$, $x_8=\{1.7, 1.8\}$, $x_9=\{2, 2\}$
- Dùng thuật toán gom cụm bằng mạng Kohonen để gom các điểm trong không gian 2 chiều nói trên. Với Map có chiều dài và chiều rộng 5x5 (5 dòng, 5 cột)

