Self Organizing Feature Maps (SOM)

M.M. Pedram
pedram@khu.ac.ir
Kharazmi University
(Spring 2020)

1

Multi Dimensional Scaling (MDS)

Problem:

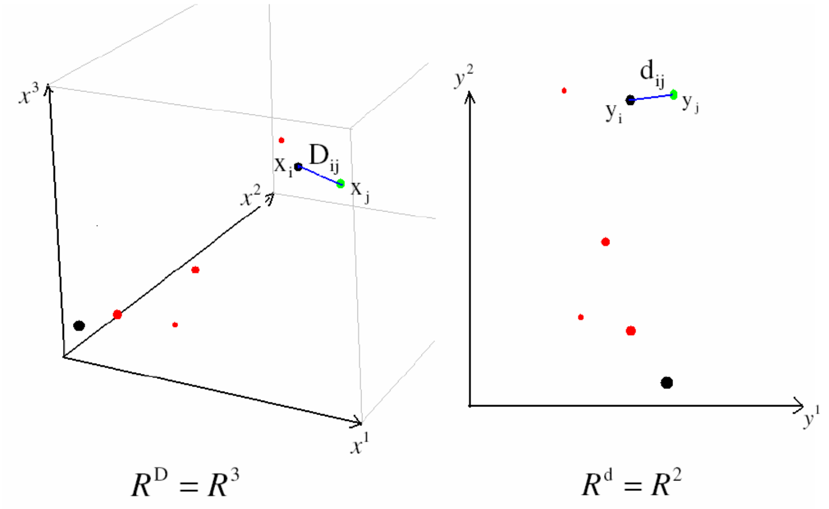
- high dimensionality of data makes it hard to analyze,
- clustering and classification are more efficient on low dimensional data,

Main ideas behind MDS:

- Project data points to low dimensional images AND
- respect constraints:
  - keep informational content
  - keep similarity / dissimilarity relationships

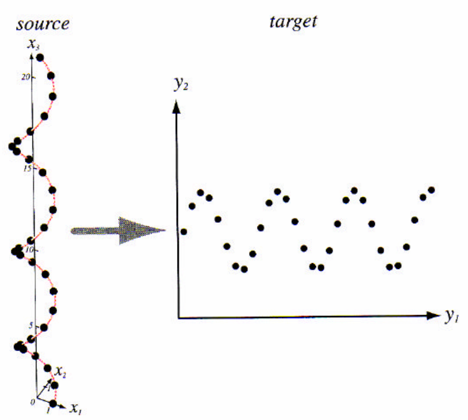
2

مثالی از MDS



مثالی از MDS (یافتن ارتباطات جالب)

30 data points; k = 0,...,29



Thirty points of the form $\mathbf{x} = (\cos(k/\sqrt{2}), \sin(k/\sqrt{2}), k/\sqrt{2})^t$

Teuvo Kohonen

شبکه‌های عصبی خودسازمانده

- Self-Organizing Feature Maps (SOFM) also known as Kohonen maps or topographic maps were first introduced by von der Malsburg (1973) and in its present form by Kohonen (1982).

- SOM is a special neural network that accepts N-dimensional input vectors and maps them to the Kohonen (competition) layer, in which neurons are organized in an L-dimensional lattice (grid) representing the feature space.

- Such a lattice characterizes a relative position of neurons with regards to its neighbours, that is their topological properties rather than exact geometric locations. In practice, dimensionality of the feature space is often restricted by its visualisation aspect and typically is L = 1, 2 or 3.

- The objective of the learning algorithm for the SOFM neural networks is formation of the feature map which captures of the essential characteristics of the N-dimensional input data and maps them on the typically 1-D or 2-D feature space.

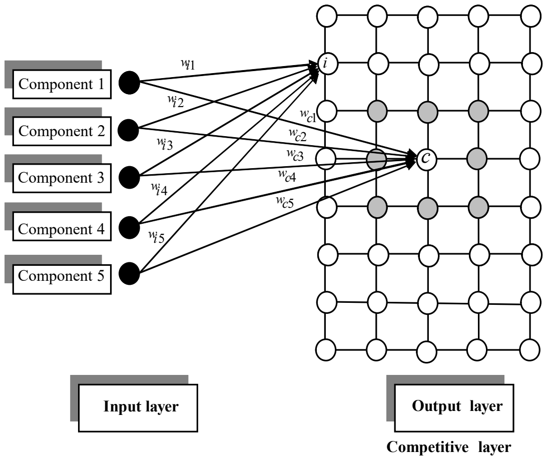
ساختار شبکه

- 2D-maps of neurons (sometimes 1D or 3D) was showen.

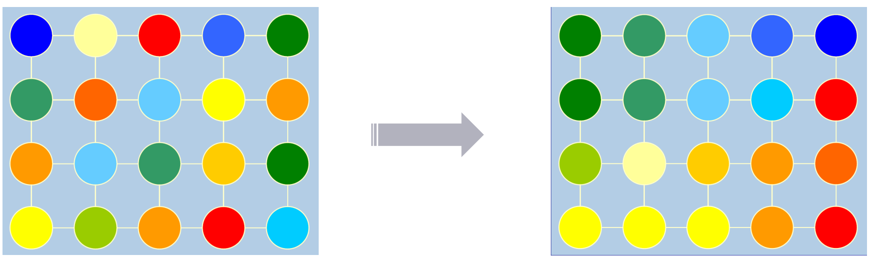
  It has N input nodes and m-by-r output nodes. Each output node j in the SOFM network has a connection from each input node i and w_{ij} denotes the connection weight between them..

- The weights of the connections from the input neurons to a single neuron in the competition layer are interpreted as a reference vector in the input space.

- That is, a SOFM basically represents a set of vectors in the input space: one vector for each neuron in the competition layer.



- Unsupervised artificial neural networks can group the objects of a dataset into different classes(putting entities geometrically close to each other), on the basis of their similarity.

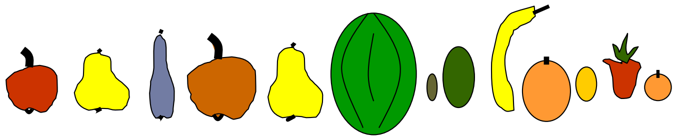


Dataset

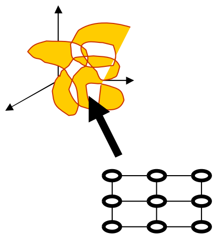
Similar objects are contained in the same neuron or in adjacent ones.

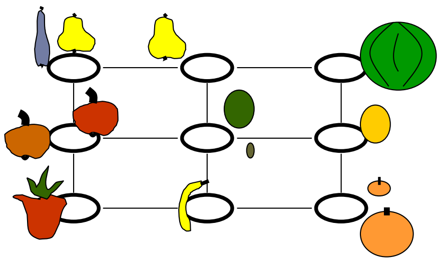
unsupervised methods:

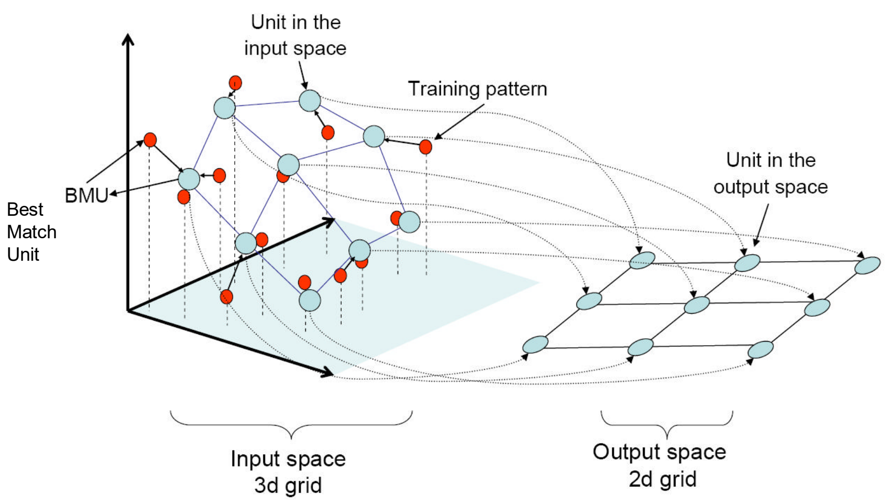
Visualize (noisy) fruits

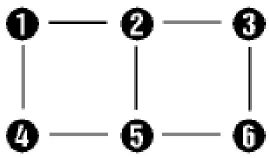


representation: (Øx,Øy,Øx/Øy,curvature,color,hardness,weight, …) $\in \mathbb{R}^n$



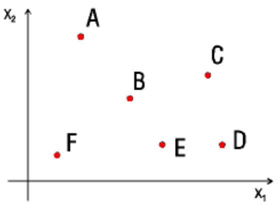




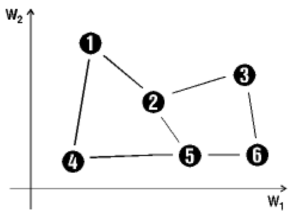


شبکه کوهنن 2بعدی با 6 نورون

توجه: خطوط فقط برای نمایش نزدیکی فضایی (همسایگی در شبکه)است، نه پیوند فیزیکی.

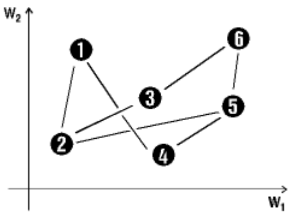


داده‌های آموزشی



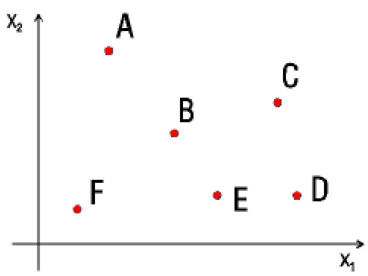
بردارهای وزنی پس از آموزش شبکه

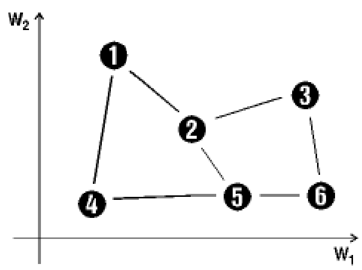
در شبکه آموزش دیده، دیاگرام در فضای وزن ها باید همان توپولوژی فضای داده ها را دارا باشد و ویژگی آنها را نشان دهد.



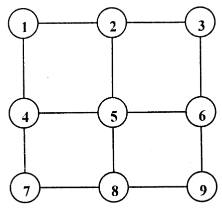
بردارهای وزنی اولیه

در شبکه آموزش دیده، دیاگرام در فضای وزن‌ها باید همان توپولوژی فضای داده‌ها را دارا باشد و ویژگی آنها را نشان دهد.

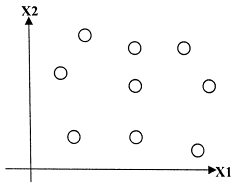




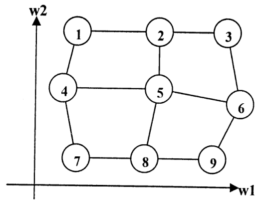
(ادامه)



شبکه کوهنن 2بعدی با 9 نورون

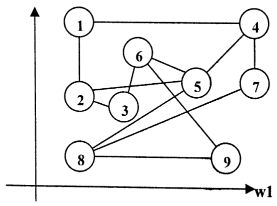


داده‌های آموزشی

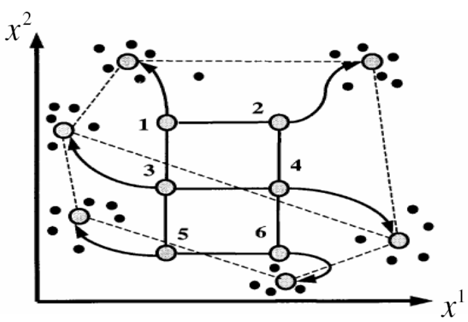


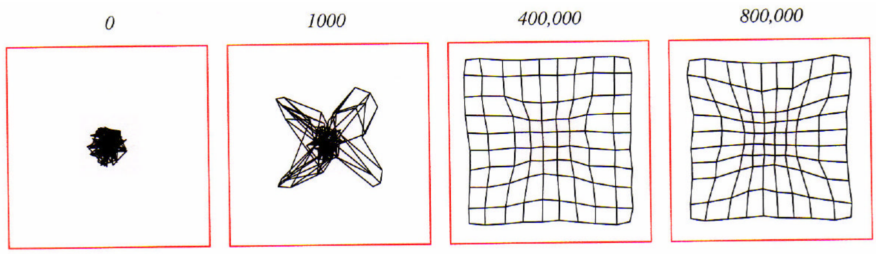
بردارهای وزنی پس از آموزش شبکه

توجه: خطوط فقط برای نمایش نزدیکی فضایی است، نه پیوند فیزیکی.



بردارهای وزنی اولیه





Steps to clustering by SOFM (SOM)

Inputs space : continuous and often high dimension

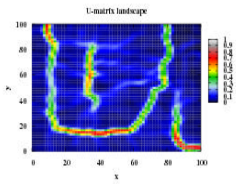
Projection

Neurons space: one neuron is associated with a prototype in the inputs space

Discrete and 2D

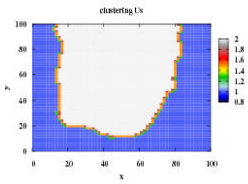
Visualization

Visualization tool of neurons space at the end of the learning process: U-matrix



Clustering

Clustering by flooding U-matrix



آموزش

Competitive learning

Neurons 'compete' to respond to (some feature of) the input – the 'winner' is activated.

- Cooperative learning (using a neighbourehood function)
- Competitive learning (*winner takes most*)

Algorithm

- 1) Choose new data sample x_i
- 2) Compute „nearest" prototype vector p^* satisfying:

$$dist(x_i, p^*) = \min_j\{dist(x_i, p_j)\}$$

- 3) Update step:
  - Move „winning" reference vector closer to the data point:

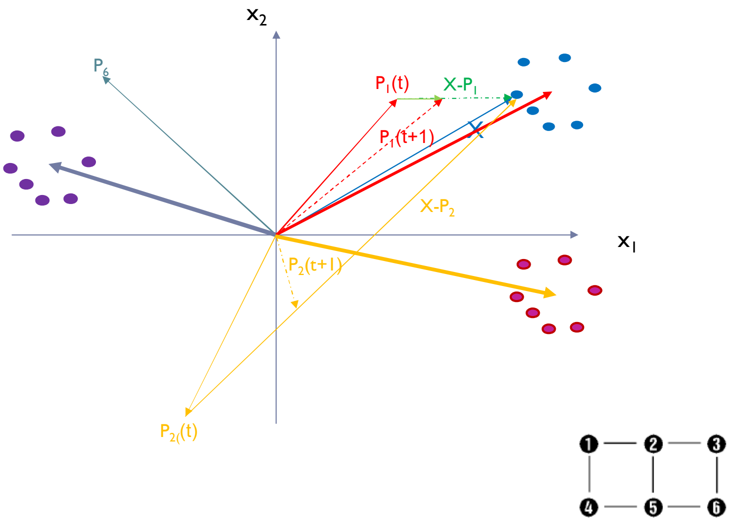
$$p^*(t+1) \leftarrow p^*(t) + \overbrace{\alpha(t)}^{\substack{learning\\rate}} \overbrace{[x_i - p^*(t)]}^{\substack{correction\\factor}}$$

  - Move its neighbouring reference vectors closer to the data:

$$p_j(t+1) \leftarrow p_j(t) + \alpha(t)\underbrace{h(t, p^*, p_j)}_{\substack{measures\\neighbourhood}}[x_i - p_j(t)]$$

- Repeat steps 1-3 while cycling repeatively through the trainings set

Algorithm



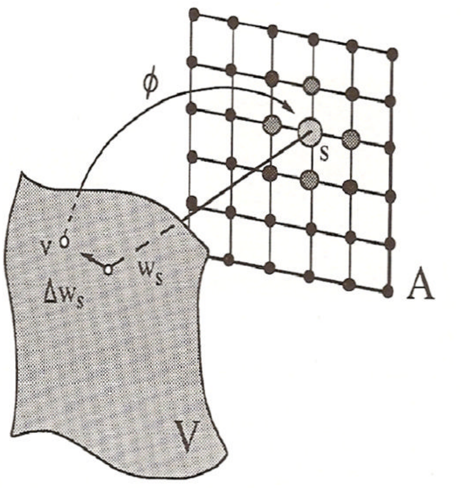
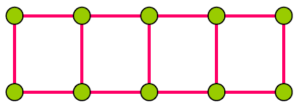


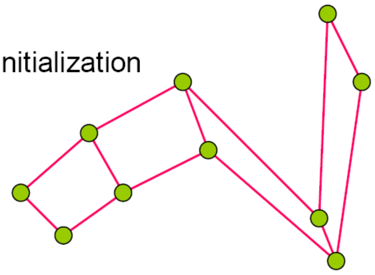
Figure 4.2 The adaptation step in Kohonen's model. The input value **v** selects a center s in whose neighborhood all neurons shift their weight vectors $\mathbf{w_s}$ towards the input **v**. The magnitude of the shift decreases as the distance of a unit from the center s increases. In the figure, this magnitude is indicated by different sizes and gray values. The shift of weights is only depicted, though, for unit s.

2*5 SOM

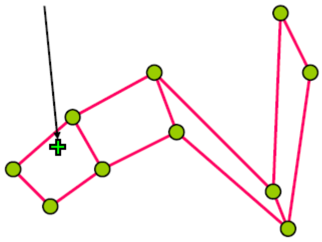


- Regular SOFM node
- Winner Node (bmu)
- Neighbor unit
- Pattern (ξ {v1,v2,v3})

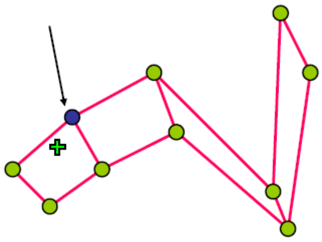
(Random) initialization

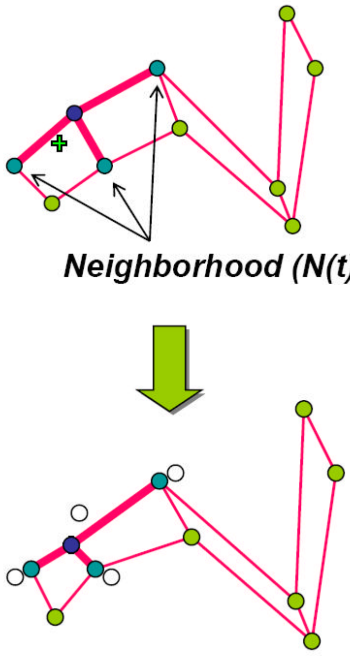


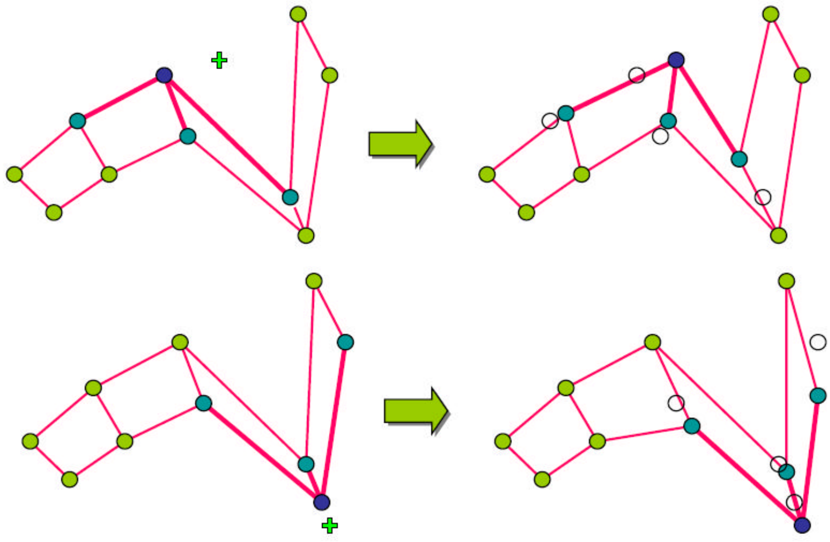
Training Pattern (ξ)



Winner Node




Neighborhood (N(t))



پارامترهای فرایند یادگیری

- If a data set consists of P input vectors or cases, then 1 learning epoch is equal to P single learning cycles
- After a number of N learning epochs, the size of the neighbourhood is decreased.
- After a number of M learning epochs, the learning rate, α, may be decreased;

همسایگی

- What is the 'shape' of the neighbourhood?
  - Circle around winner
  - 'Manhattan'-type (i.e. measured in unit blocks)
- How is the 'degree of neighbourhood' calculated?
  - All neighbours equivalent
  - Degree of nbhd. diminishes with distance
  - Inhibitory effects for distant nodes
- How does the width of the neighbourhood decrease with time?
  - Linear
  - Ordering / convergence phases

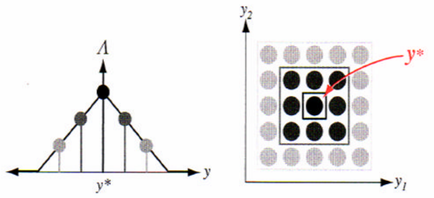
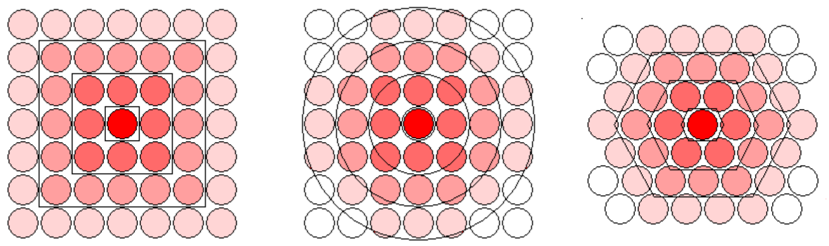
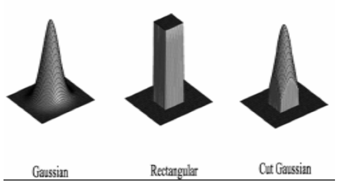


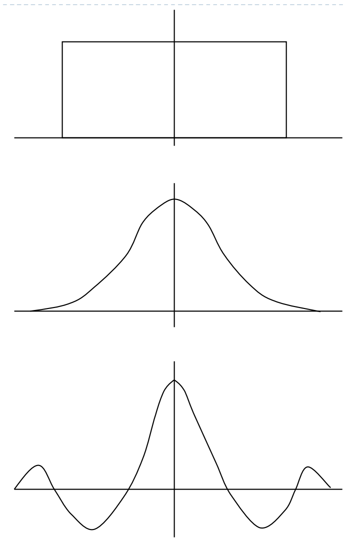
FIGURE 10.29. Typical window functions for self-organizing maps for target spaces in one dimension (left) and two dimensions (right). In each case, the weights at the maximally active unit, y^*, in the target space get the largest weight update while units more distant get smaller update.



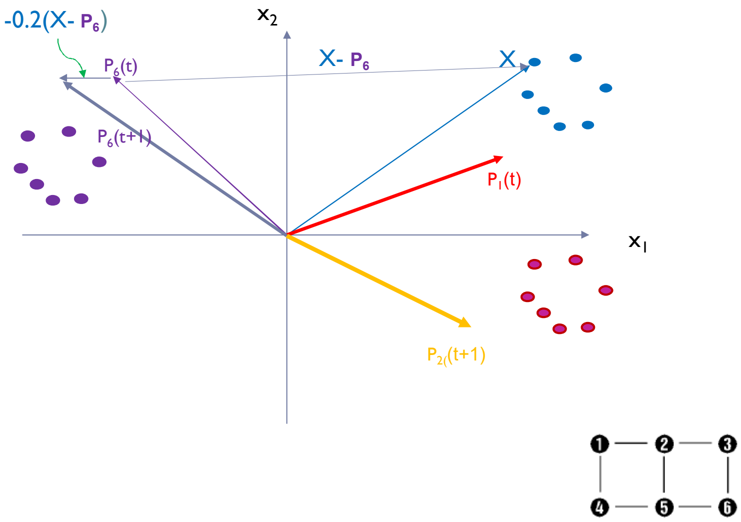
انواع توابع همسایگی



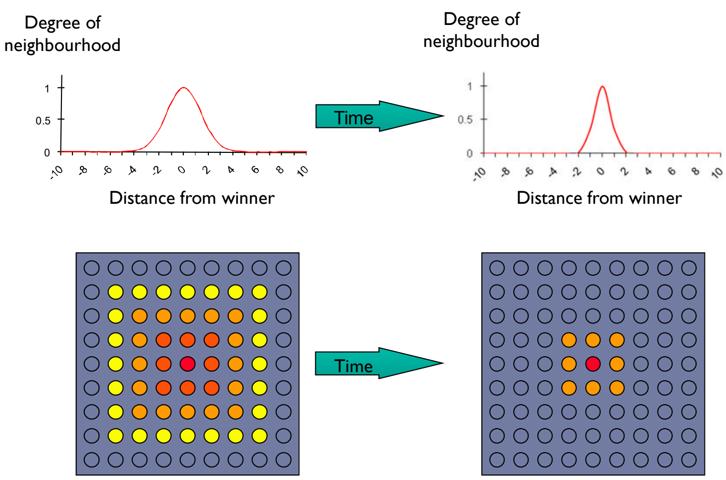
- Constant amplitude
- Gaussian
- 'Mexican hat'
  - x-axis shows distance from winning node (negative distance allowed?)
  - y-axis shows 'degree of neighbourhood' (max. 1)



Algorithm



-0.2(X- P6)
x2
X- P6
X
P6(t)
P6(t+1)
P1(t)
x1
P2(t+1)
1 2 3
4 5 6



Degree of neighbourhood
1
0.5
0
-10 -8 -6 -4 -2 0 2 4 6 8 10
Distance from winner
Time
Degree of neighbourhood
1
0.5
0
-10 -8 -6 -4 -2 0 2 4 6 8 10
Distance from winner
Time

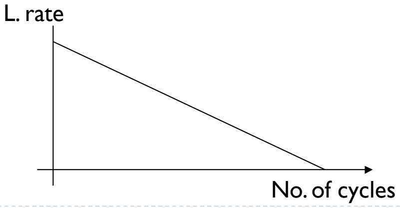
نرخ یادگیری

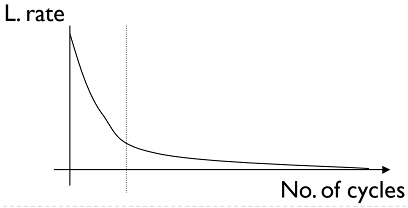
Why do we need to decrease the learning rate α ?

- If the learning rate α is kept constant, **it is possible** for weight vectors to oscillate back and forth between two nearby positions; Lowering α **ensures** that this does not occur and the network is stable.

Learning rate decreases with time
- Linear
- Ordering / convergence phases
- Stepped functions

(Similar graphs apply to the width of the neighbourhood)





تعریف فضای خروجی

- Dimensionality
  - 2d
  - 3d
- Shape
  - Square
  - Rectangular
  - Hexagonal
  - Others?
- Size
- Distance measure
  - Euclidean
  - ‘Manhattan’ (or hexagonal equivalent)

شرط توقف فرایند یادگیری

Repeat steps 1 to 3 until:

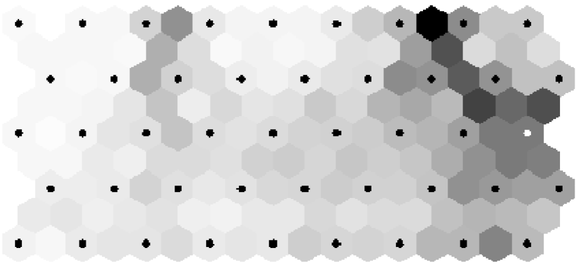
- the map has converged (i.e. no noticeable changes in the weights), or
- pre-defined no. of training cycles have passed.

معایب

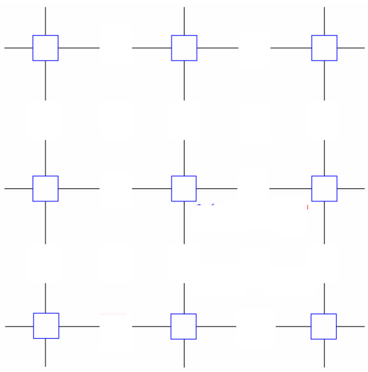
- Large quantity of good quality representative training data required.
- Possibly misleading visualisation?
  - Neurons close together in output space represent similar input patterns, but doesn't *necessarily* follow that those far apart will be very different.
- No generally accepted measure of 'quality' of a SOM
  - Average quantization error (how well the data is classified).
  - Topological preservation measures (how well the output neurons are ordered).

U-Matrix : Unified Matrix Method

- U-matrix representation of SOM visualizes the distance between the neurons. The distance between the adjacent neurons is calculated and presented with different colorings between the adjacent nodes.

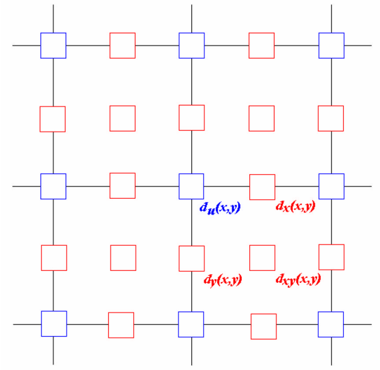


U-matrix representation of the SOM



$b(x, y)$: matrix of neurons, of size $n_x \times n_y$.
$w_i(x, y)$: matrix of weights.
$u(x, y)$: U-matrix of size $(2n_x - 1) \times (2n_y - 1)$.

$d_x(x, y)$: $\|b(x, y) - b(x+1, y)\| = \sqrt{\sum_i [w_i(x, y) - w_i(x+1, y)]^2}$
$d_y(x, y)$: $\|b(x, y) - b(x, y+1)\| = \sqrt{\sum_i [w_i(x, y) - w_i(x, y+1)]^2}$
$d_{xy}(x, y)$: $\frac{1}{2}[\frac{\|b(x,y) - b(x+1,y+1)\|}{\sqrt{2}} + \frac{\|b(x,y+1) - b(x+1,y)\|}{\sqrt{2}}]$
$d_u(x, y)$: the median of the surrounding elements.

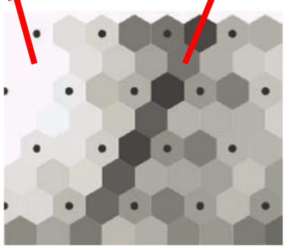


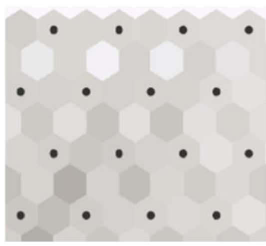
Interpreting Cluster Tendency Maps

Light areas show likely clusters

Dark areas show likely cluster boundaries

Uniform areas indicate that input data contains little information

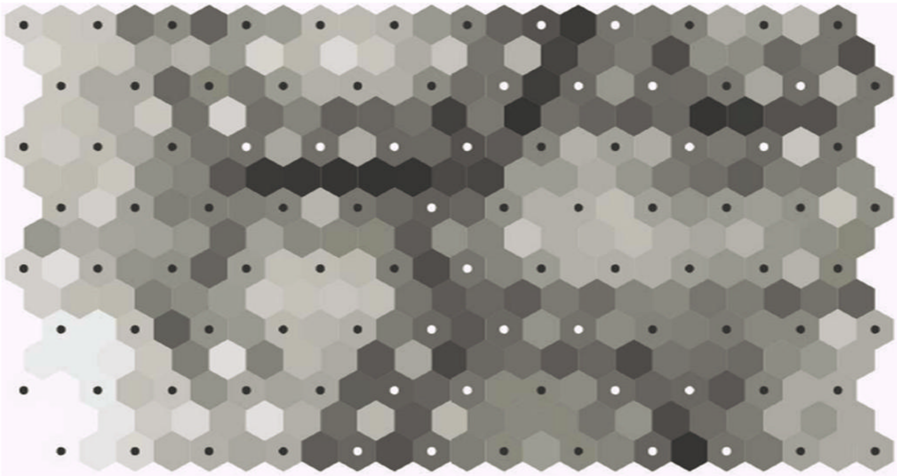




Example with high clustering tendency
= data contains lots of information

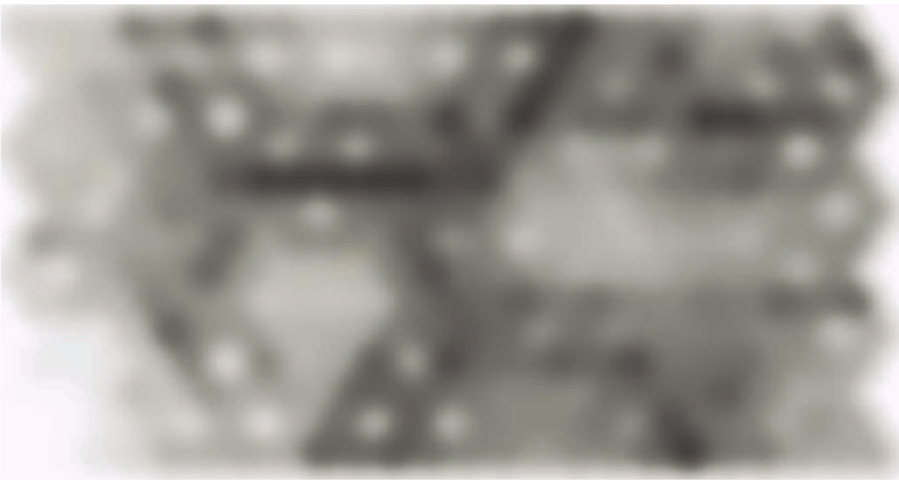
Example with low clustering tendency
= data contains little or no information

Cluster Tendency



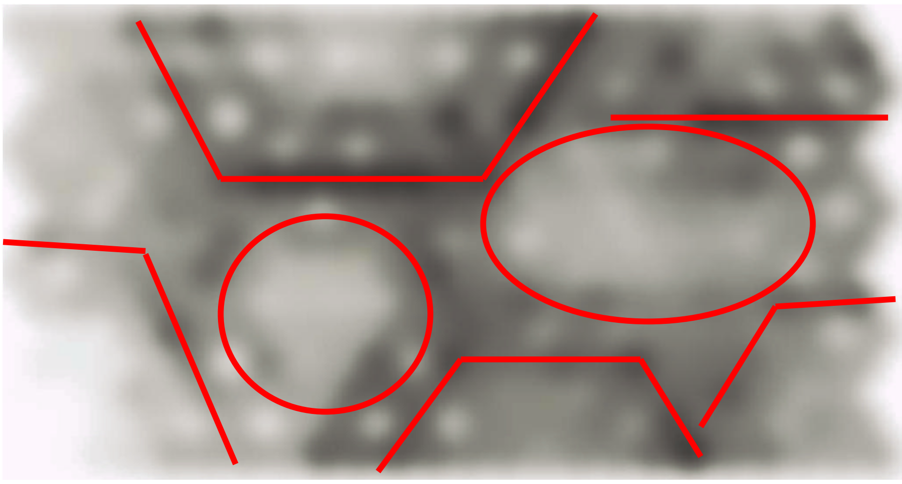
Light areas: high density of data samples – likely clusters,
dark areas: low density of data samples – likely cluster boundaries

Cluster Tendency - Smoothed



Smoothed cluster tendency map emphasizes likely clusters and cluster boundaries

Cluster Tendency - Smoothed



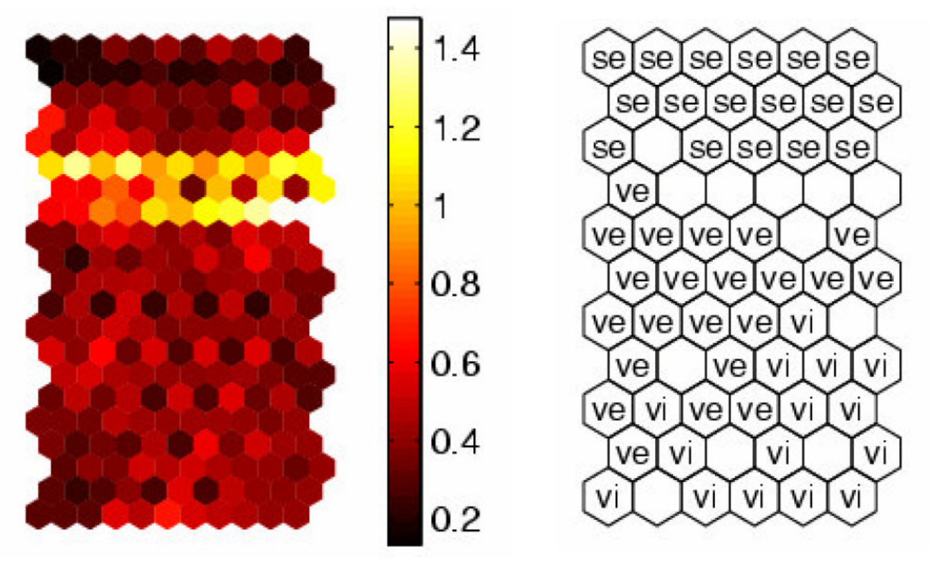
Likely clusters and cluster boundaries are highlighted

مثال- گل زنبق (iris)

3 classes of Iris flowers:

- setosa,
- versicolor
- virginica

- 150 examples with 4 attributes


1.4
1.2
1
0.8
0.6
0.4
0.2
se se se se se se
se se se se se se
se se se se se
ve
ve ve ve ve ve
ve ve ve ve ve ve
ve ve ve ve vi
ve ve vi vi vi
ve vi ve ve vi vi
ve vi vi vi
vi vi vi vi vi