Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava

Neural Networks

Lecture 7

Self-organizing map

Igor Farkaš

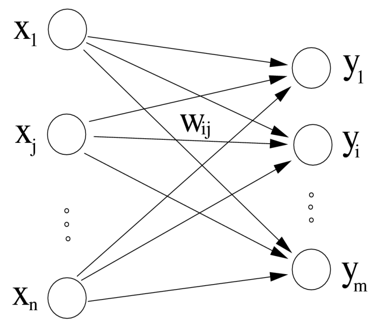
2018

Simple competitive learning

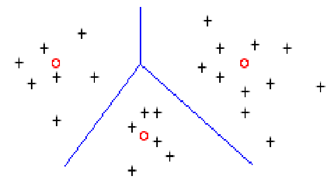
- unsupervised learning
- linear neurons
- winner: $y_{i*} = \max_i \{\boldsymbol{w}_i^{\mathrm{T}} . \boldsymbol{x}\}$
  - i.e. best matching unit *i**
- winner-take-all adaptation:

$$\Delta \boldsymbol{w}_{i*} = \alpha(\boldsymbol{x} - \boldsymbol{w}_{i*}) \qquad \alpha \in (0,1)$$

$$\|\boldsymbol{w}_{i*}\| = 1$$

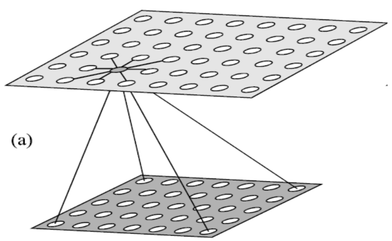


- risk of “dead” neurons
- algorithm: in each iteration:
  (1) find winner, (2) adapt its weights
- useful for clustering

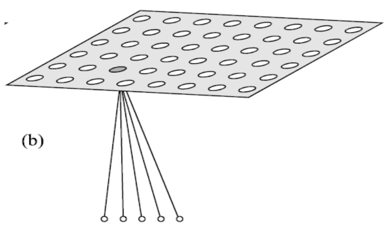


Feature mapping

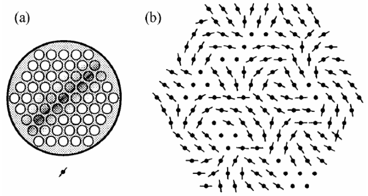
biologically motivated models



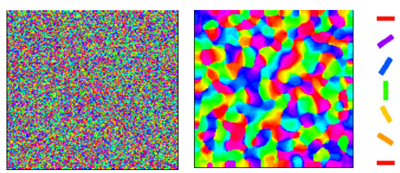
model with extracted features



e.g. mapping from retina to cortex -> orientation map

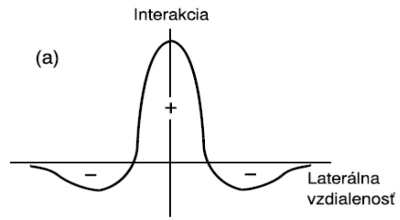


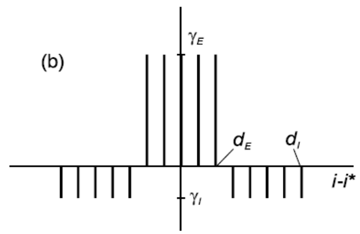
- introduced topology of neurons in the map
- winner-take-most due to neuron cooperation



Lateral interactions in the map

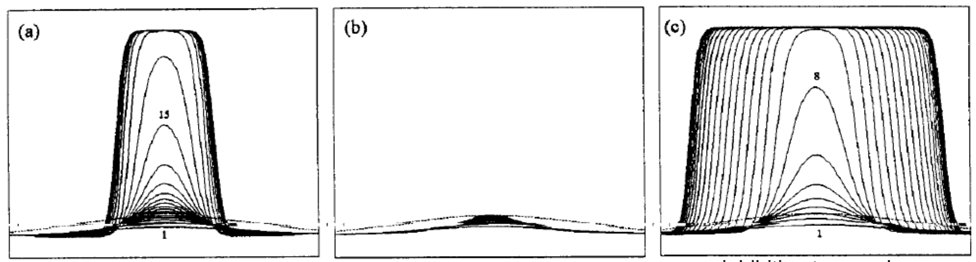
Mexican hat function (1D case)





$$y_i(t+1) = s(z_i + \sum_{k=-K}^{K} l_{ik} . y_{i+k}(t))$$

initial response $z_i = \boldsymbol{w}_i^{\mathrm{T}} . \boldsymbol{x}$



inhibition too strong

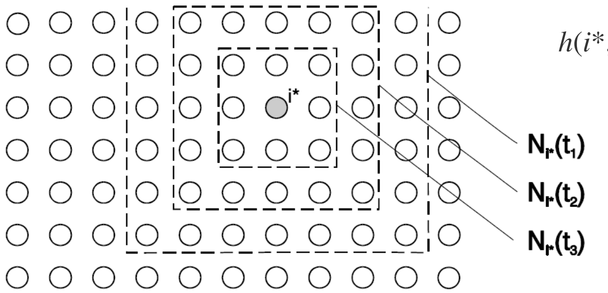
inhibition too weak

(Farkaš, 1997)

Neighborhood function in SOM

- computationally efficient substitute for lateral interactions
- neurons adapt only within the winner neighborhood
- neighborhood radius decreases in time

- rectangular neighborhood (below)



$h(i^*,i) = 1$ if $i \in N_{i^*}(t)$
0 otherwise

- alternative: gaussian neighborhood, e.g. $h(i^*,i) = \exp\{-\frac{d_E^2(i^*,i)}{\lambda^2(t)}\}$

$\lambda(t) = \lambda_i.(\lambda_f/\lambda_i)^{t/t_{max}}$

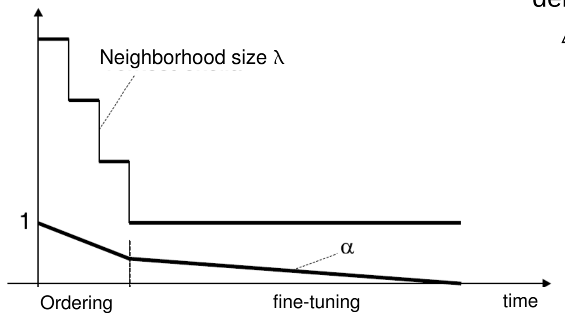
SOM algorithm (ED version*)

(Kohonen, 1982)

- randomly choose an input $\boldsymbol{x}$
- find winner i^* for $\boldsymbol{x}$ $\qquad i^* = \arg\min_i \|\boldsymbol{x} - \boldsymbol{w}_i\|$
- update weights within the neighborhood

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + \alpha(t) h(i^*, i)[\boldsymbol{x}(t) - \boldsymbol{w}_i(t)]$$

- update SOM parameters (neighborhood, learning rate)
- repeat until stopping criterion is met

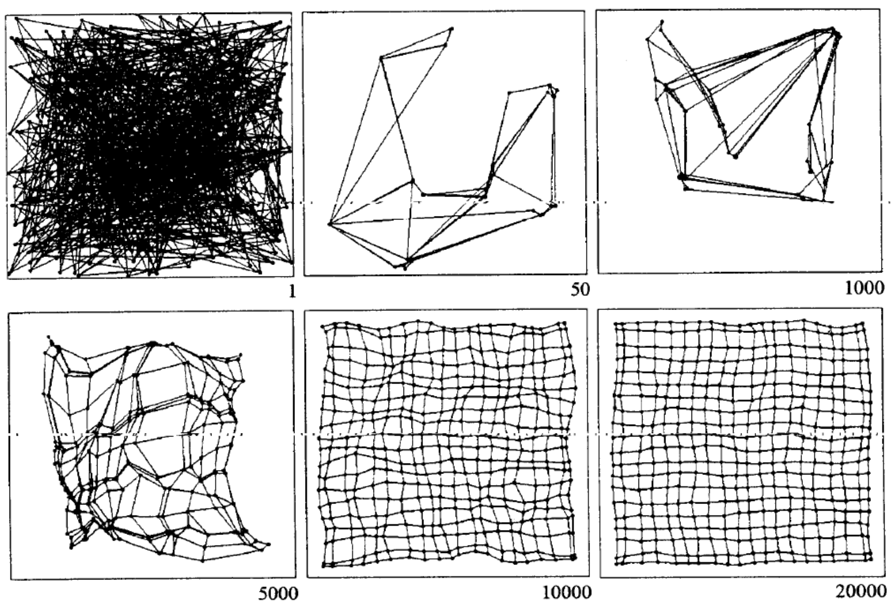


derived from a general Hebbian form:
$\Delta \boldsymbol{w}_i = \alpha\, y_i \boldsymbol{x} - g(y_i)\, \boldsymbol{w}_i$

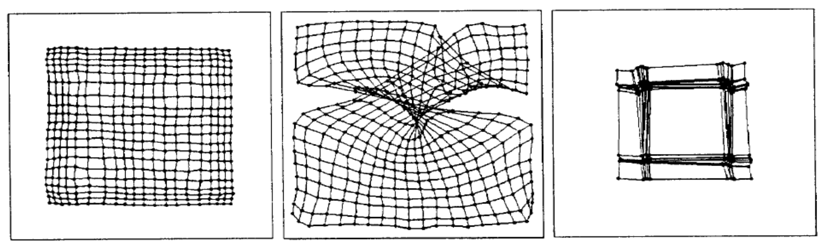
SOM as input-output mapping:
$\boldsymbol{x} \rightarrow \{1, 2, ..., m\}$ or
$\boldsymbol{x} \rightarrow \boldsymbol{y}$, $\quad \boldsymbol{y} = [y_1, y_2, ..., y_m]$
where e.g. $y_i = \exp(-\|\boldsymbol{x} - \boldsymbol{w}_i\|^2)$

* i.e. based on Euclidean distance

Example: 2D random inputs, 20x20 neurons



Special effects

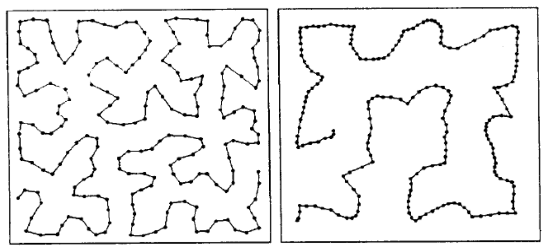


too fast α↓ too fast λ↓ too slow λ↓

Neighborhood effect

2D data, 1D map

rectangular

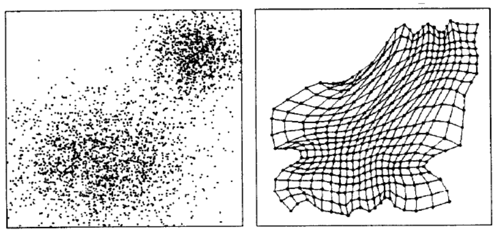


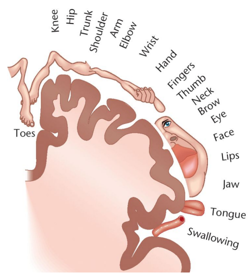
Gaussian

Magnification property

• SOM roughly approximates input data distribution

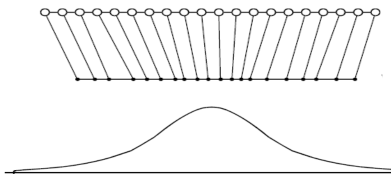
2D:

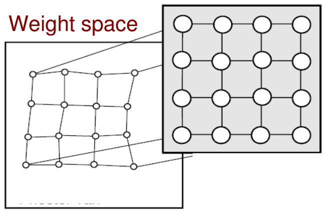




Somatosensory homunculus in the brain

1D:





Theory for 1D: $w(x) \propto p(x)^{2/3}$

(Ritter, 1991)

SOM simultaneously performs two tasks

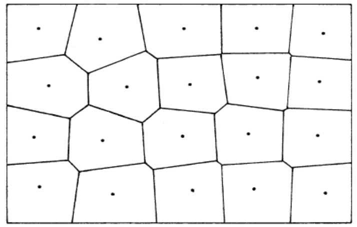
Vector quantization

(if number of inputs > number of neurons)

Voronoi compartments:

$$V_i = \{x \mid \|x - w_i\| < \|x - w_j\|\}, \ \forall j \neq i$$

Quant. error: $QE = \frac{1}{N} \sum_p \|x_p - w_{i*}\|^2$



Voronoi tessellation

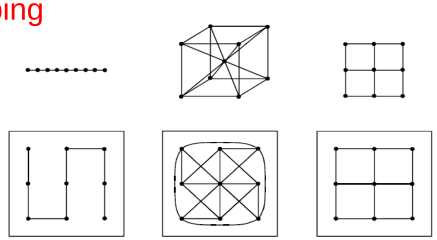
Topology preserving mapping

Cost function

e.g. (Kohonen, 1991)

$$E(w) = \frac{1}{N} \sum_p \sum_i h(i*, i) \|x_p - w_{i*}\|^2$$

neurons



X←Y X→Y X↔Y

various measures of topology preservation proposed

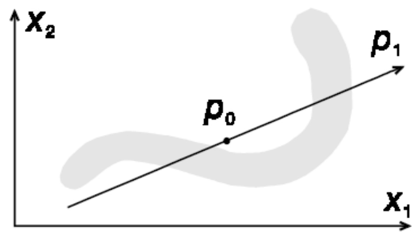
Main properties of SOM

- Approximation of the input space (input data) by the grid of neurons → vector quantization theory
- Topological ordering – preservation of similarities between input and output spaces
- Density matching – reflecting the variations in the statistics of input distribution
- Feature selection – via nonlinear mapping → principal curves or surfaces (Hastie and Stuetzle, 1989)
  - SOM as a nonlinear generalization of PCA

Comparison of SOM to PCA

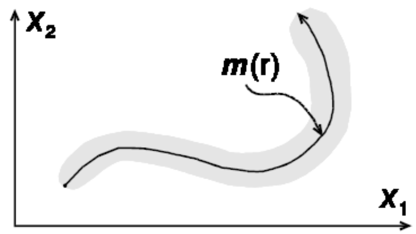
- feature extraction and mapping, difference in feature representation

PCA



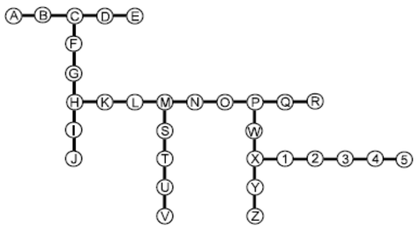
(linear) principal components
One unit represents 1 dimension

SOM

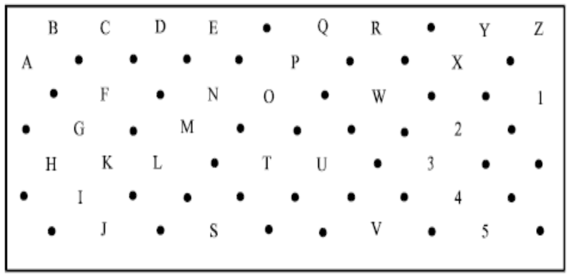


(nonlinear) principal manifold
More units represent 1 dimension

Application: Minimum spanning tree



(Kohonen,1990)

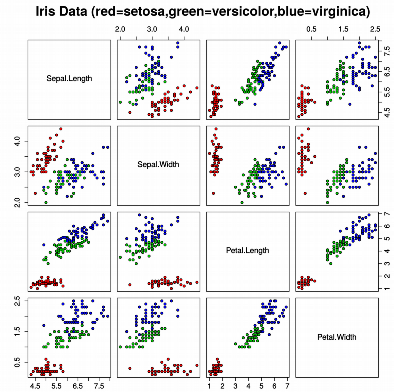


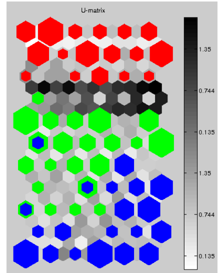
Input vector encoding:

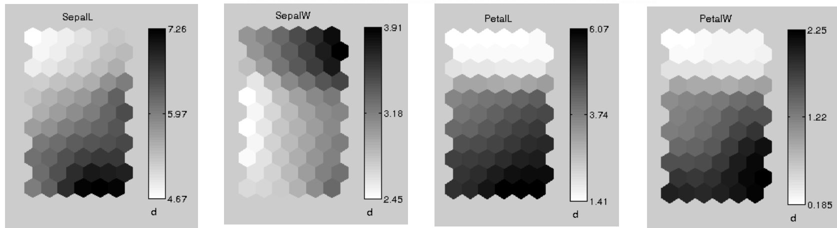
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5
1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0	0	0	0	0	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	3	3	3	3	6	6	6	6	6	6	6	6	6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	1	2	3	4	2	2	2	2	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5

Application: Data visualization

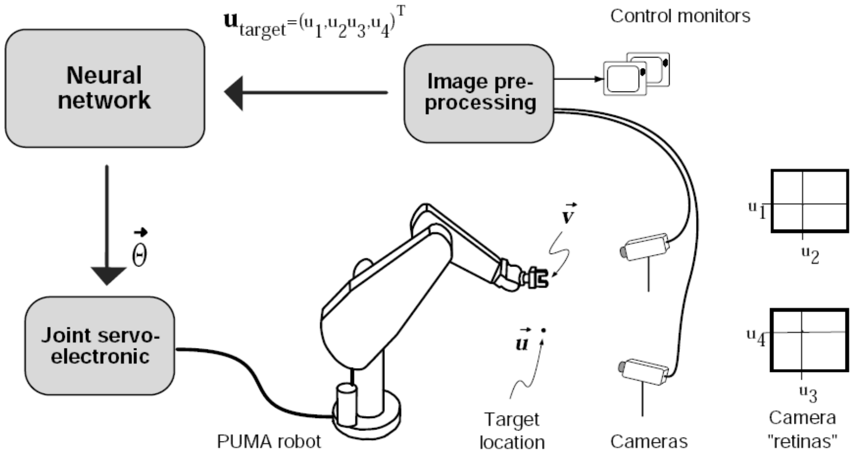
- 7x10 SOM with hexa grid
- trained on 4D Iris data
- 3 classes
- Neuron labels generated
- according to votes
- Plots of component weights reveal an order







Application: Robotic arm control



$$\theta(\mathbf{u}) = \theta_i + \mathbf{A}_i.(\mathbf{u} - \mathbf{w}_i)$$

$$\mathbf{w}_i \leftarrow \mathbf{w}_i + \varepsilon.h(i,i^*).(\mathbf{u} - \mathbf{w}_i)$$

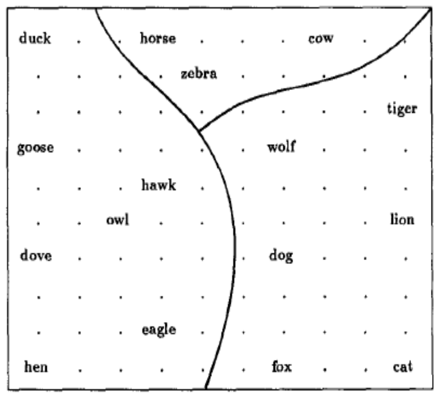
$$\theta_i \leftarrow \theta_i + \varepsilon.h(i,i^*).\Delta\theta_i$$

$$\mathbf{A}_i \leftarrow \mathbf{A}_i + \varepsilon.h(i,i^*).\Delta\mathbf{A}_i$$

(Walter & Schulten 1993)

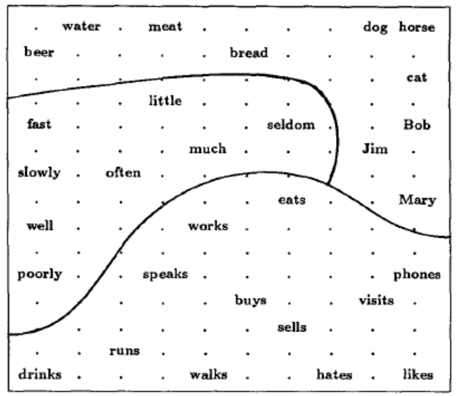
Application: Semantic maps

Attribute map



Words $\boldsymbol{x}$ = [$\boldsymbol{x}_{symbol}$; $\boldsymbol{x}_{attr}$], i.e. concatenation of (16-dim) symbolic (one-hot) and binary attribute vectors (13-dim), of 3 categories (is, has, likes to). Testing done using $\boldsymbol{x}$ = [$\boldsymbol{x}_{symbol}$; **0**].

Role-based map

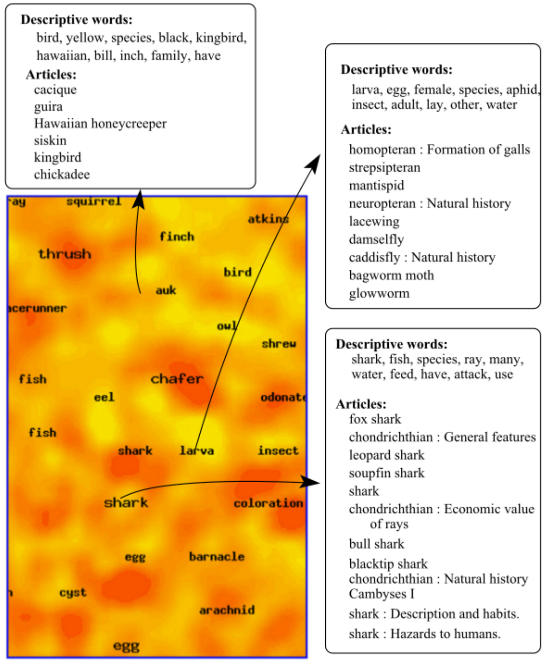


Words represented in contexts, i.e. $\boldsymbol{x}$ = [$\boldsymbol{x}_{symbol}$; $\boldsymbol{x}_{context}$], using sentence templates [noun - verb - adverb/noun]. Symbol = 7-dim. vector (of unit length), (average L-R) context = 14-dim. (random projection used). Labeling based on $\boldsymbol{x}$ = [$\boldsymbol{x}_{symbol}$; **0**].

(Ritter & Kohonen, 1989)

Application: WEBSOM

- e.g. collection of cca 12000 Encyclopaedia Britannica documents
- Used for information retrieval
- Random mapping (of word vectors) useful: preserves similarities
- Based on co-occurrence of 40000 words in documents
- SOM with cca 12000 nodes



(Kaski et al, 1998)

Dot-product version of SOM

- randomly choose an input $\boldsymbol{x}$
- find winner i^* for $\boldsymbol{x}$ $\quad i^* = \arg\max_i \{\boldsymbol{x}^T . \boldsymbol{w}_i\}$
- update weights within the neighborhood

$$\boldsymbol{w}_i(t+1) = \frac{\boldsymbol{w}_i(t) + \alpha(t).h(i^*, i).\boldsymbol{x}(t)}{\|\boldsymbol{w}_i(t) + \alpha(t).h(i^*, i).\boldsymbol{x}(t)\|}$$

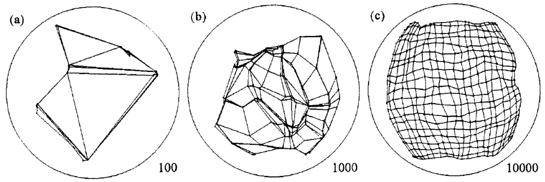
- update SOM parameters (neighborhood, learning rate)
- repeat until stopping criterion is met

Example: DP-SOM map trained on 3D vectors $\boldsymbol{s} = [s_1, s_2, s_3]$ created from 2D $\boldsymbol{x} = [x_1, x_2]$ as

$s_1 = 1 \cdot \cos(x_1) \cdot \cos(x_2)$

$s_2 = 1 \cdot \sin(x_1) \cdot \cos(x_2)$

$s_3 = 1 \cdot \sin(x_2)$



- Weight vector ordering independent of the input vectors' norms

Related self-organizing NN algorithms

- Can be viewed as unsupervised data approximation with undirected graph $G = (V,C)$, $V = \{\mathbf{w}_i\}$ ~ vertices, $C[m \times m]$ ~ (symmetric) connection matrix

Examples:

- Topology-Representing network (Martinetz & Schulten 1994)
  - Flexible net topology, fixed number of units
- Growing Cell Structures (Bruske & Sommer, 1995)
  - Flexible topology and number of units (they can be removed or added based on max. quantization error)
- useful for non-stationary data distributions

Summary

- self-organizing map – a very popular algorithm
  - principles of competition and cooperation, unsupervised learning
  - performs vector quantization and topology-preserving mapping
- useful for data clustering and visualization
- theoretical analysis of SOM limited to simple cases
- various self-organizing algorithms developed
  - main purpose: data clustering
  - not all implement dimensionality-reducing mapping
  - flexible architectures possible