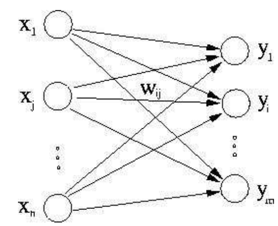
Introduction to artificial neural networks

Competition in neural networks
Self-organizing map

Igor Farkaš
Department of Applied Informatics
Comenius University in Bratislava

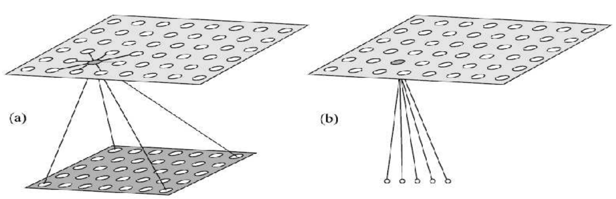
Simple competitive learning

- a kind of unsupervised learning
- Features:
  - linear neurons
  - winner: $y_c = \max_i \{\mathbf{w}_i^T.\mathbf{x}\}$
    - i.e. best matching unit *c*
  - winner-take-all adaptation:
    $\Delta \mathbf{w}_c = \alpha(\mathbf{x} - \mathbf{w}_c) \qquad \alpha \in (0,1)$
    $\|\mathbf{w}_c\| \leftarrow 1$
  - risk of "dead" neurons
- algorithm: in each iteration
  - find winner, adapt its weights
- useful for clustering

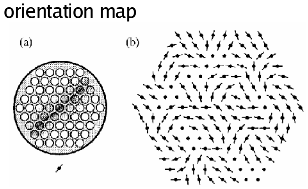


Feature mapping

biologically motivated models Self-organizing map (SOM)



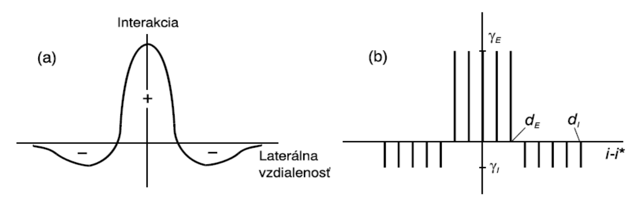
e.g. mapping from retina to cortex -> orientation map



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

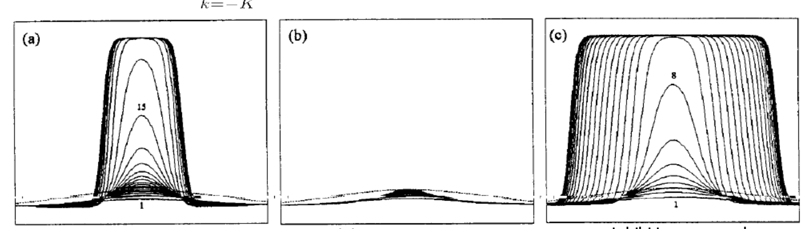
Lateral interactions in the map

Mexican hat function
(1D case)



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

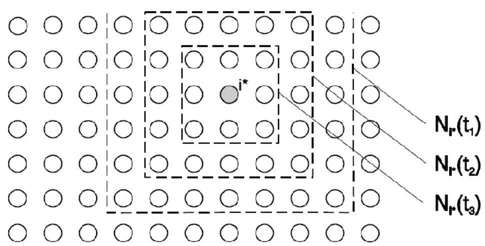
initial response $z_i = \mathbf{w}_i^T.\mathbf{x}$



inhibition too strong inhibition too weak

Neighborhood function in SOM

- computationally efficient substitute for lateral interactions
- neurons adapt only within the winner neighborhhod
- neighborhhod radius decreases in time
- rectangular neighborhood (below)



- alternative: gaussian neighborhood

$$h(i^*, i) = \exp\{-\frac{d_E^2(i^*, i)}{\lambda^2(t)}\}$$

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

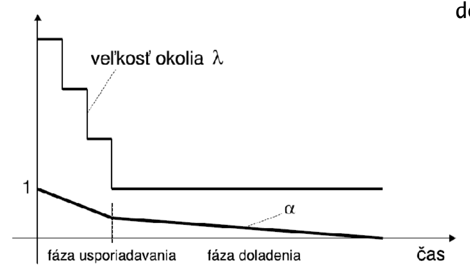
SOM algorithm (Kohonen, 1982)

- randomly choose an input ***x***
- find winner *i** for ***x***
- adapt weights within the neighborhood

$$i^* = \arg\min_i \|\mathbf{x} - \mathbf{w}_i\|$$

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

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



derived from general hebbian form:

$$\Delta \boldsymbol{w}_i = \alpha\, y_i \boldsymbol{x} - g(y_i)\, \boldsymbol{w}_i$$

I/O mapping:

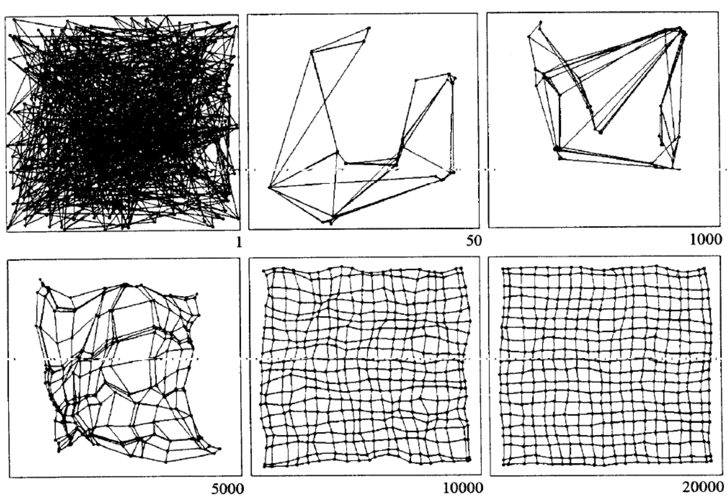
$\mathbf{X} \rightarrow \{1,2,\ldots,m\}$ or

$\mathbf{X} \rightarrow \mathbf{Y}$

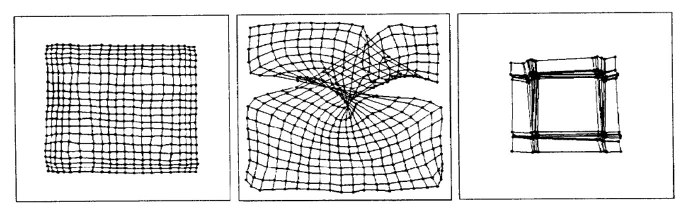
$\boldsymbol{y} = [y_1, y_2, \ldots, y_m]$

where e.g. $y_i = \exp(-\|\boldsymbol{x}-\boldsymbol{w}_i\|)$

Example: 2D inputs, 20x20 neurons

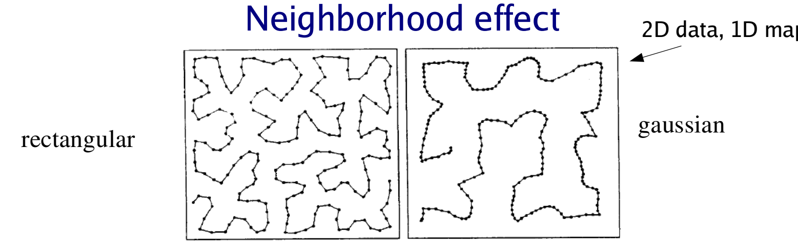


Special effects



too fast α↓ too fast λ↓ too slow λ↓

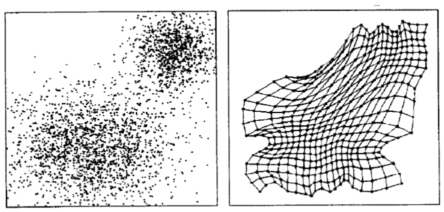
Neighborhood effect

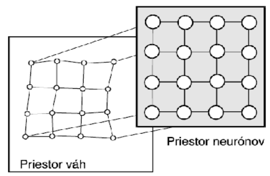


Magnification factor

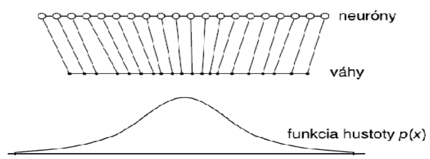
- SOM tends to approximate input distribution

2D:





1D:



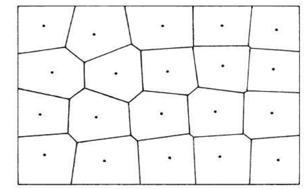
Theory for 1D: $\propto P(x)^{2/3}$

SOM performs 2 tasks simultaneously

Vector quantization

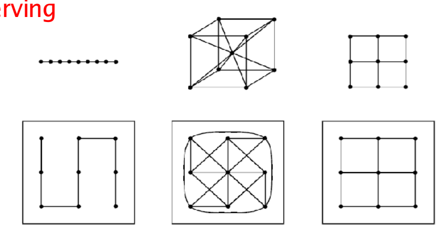
Voronoi compartments:

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



Voronoi tesselation

Topology preserving mapping



various measures of topology preservation proposed

Theoretical analysis

- restricted to simple cases (1D or 2D inputs, 1D map)

- vector quantization
- error functions
- weight ordering
- weight convergence

$$\sum_{t=0}^{\infty} \alpha(t) = \infty$$

$$\lim_{t \to \infty} \alpha(t) = 0 \quad V) = \frac{1}{N} \sum_{i=1}^{n} \sum_{x_j \in \Omega_i} \sum_{m=1}^{n} h(i,m).\|x_j - w_m\|^2$$

(Růžička, 1993)

$$\sum_{t=0}^{\infty} \alpha(t) = \infty$$

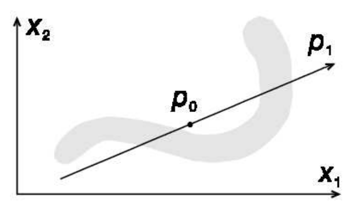
$$\lim_{t \to \infty} \alpha(t) = 0$$

(Ritter, Schulten, 1988)

Comparison of SOM to PCA

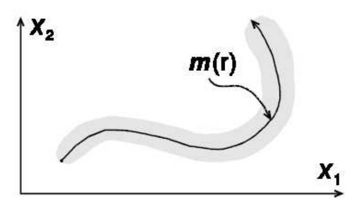
- feature extraction and mapping, difference in feature representation

PCA



(linear) principal components

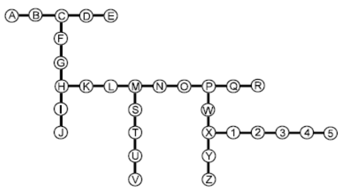
SOM

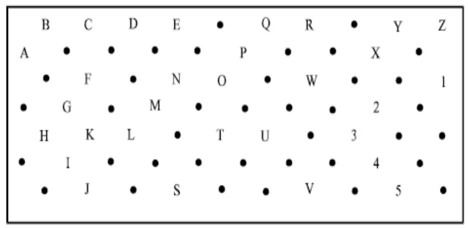


(nonlinear) principal manifold

Application: Minimum spanning tree

(Kohonen,1990)



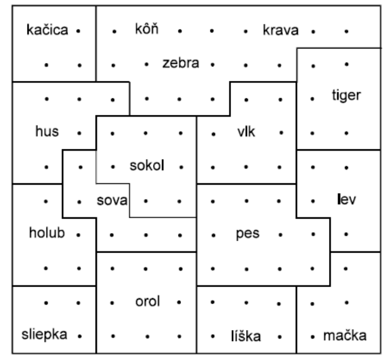


Input vector encoding:

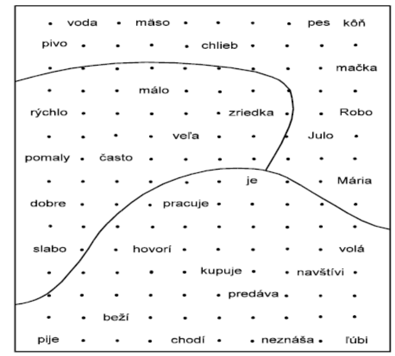
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5
1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0	0	0	0	0	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	3	3	3	3	6	6	6	6	6	6	6	6	6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	1	2	3	4	2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

Attribute map

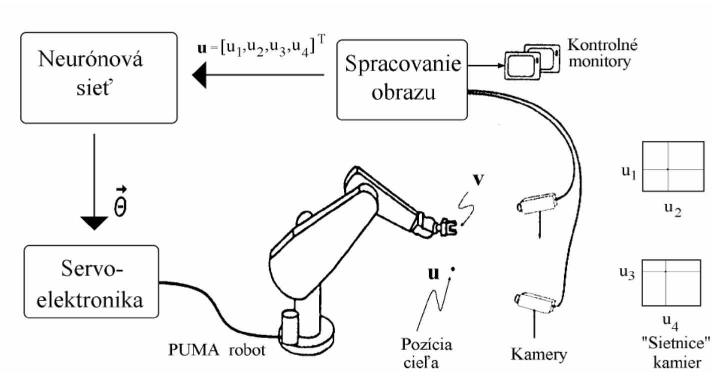


Contextual map



(Ritter & Kohonen, 1989)

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)

Related self-organizing algorithms

- common features: competition, cooperation
- distinctive feature: architecture, feature mapping

Algoritmus	Redukcia dim.	n = konšt.	Monodim. graf	Fix. topológia
SOM	áno	áno	áno	áno
VQP	áno	áno	áno	nie
TRN	nie	áno	nie	nie
GCS	nie	nie	áno	nie
DCS	nie	nie	nie	nie
GSOM	áno	nie	áno	nie

(Kvasnička a spol., 1997)