



České vysoké učení technické v Praze



Fakulta elektrotechnická



**Katedra kybernetiky
Katedra počítačů**



Vytěžování dat – přednáška 8

Kompetiční učení, shluková analýza, SOM

Osnova přednášky

- Kompetiční učení
- Shluková analýza
- Neuronová síť SOM (bez učitele)
 - Biologická inspirace SOM
 - Historie
 - Architektura
 - Učení sítě
- Interpretace SOM
 - Sammonova projekce
 - U-matice

Kompetiční učení

Jedinci (elementy, neurony) spolu soutěží

- Příklad – bezdomovci a kontejnery
 - Pamatuji si, kde byla dobrá kořist
 - Vyhraje ten, kdo přijde dřív
 - Musím být poblíž, aby mě někdo nepředběhl
 - Když se dozvím o novém kontejneru, a mám šanci ho vybrat, musím se přesunout blíže k němu
 - Kdo se to nenaučí, umře hlady
 - Vede na teritoriální uspořádání, reflektující rozmístění kontejnerů a jejich využívanost

O tom bude dnešní cvičení ...

Kompetiční učení

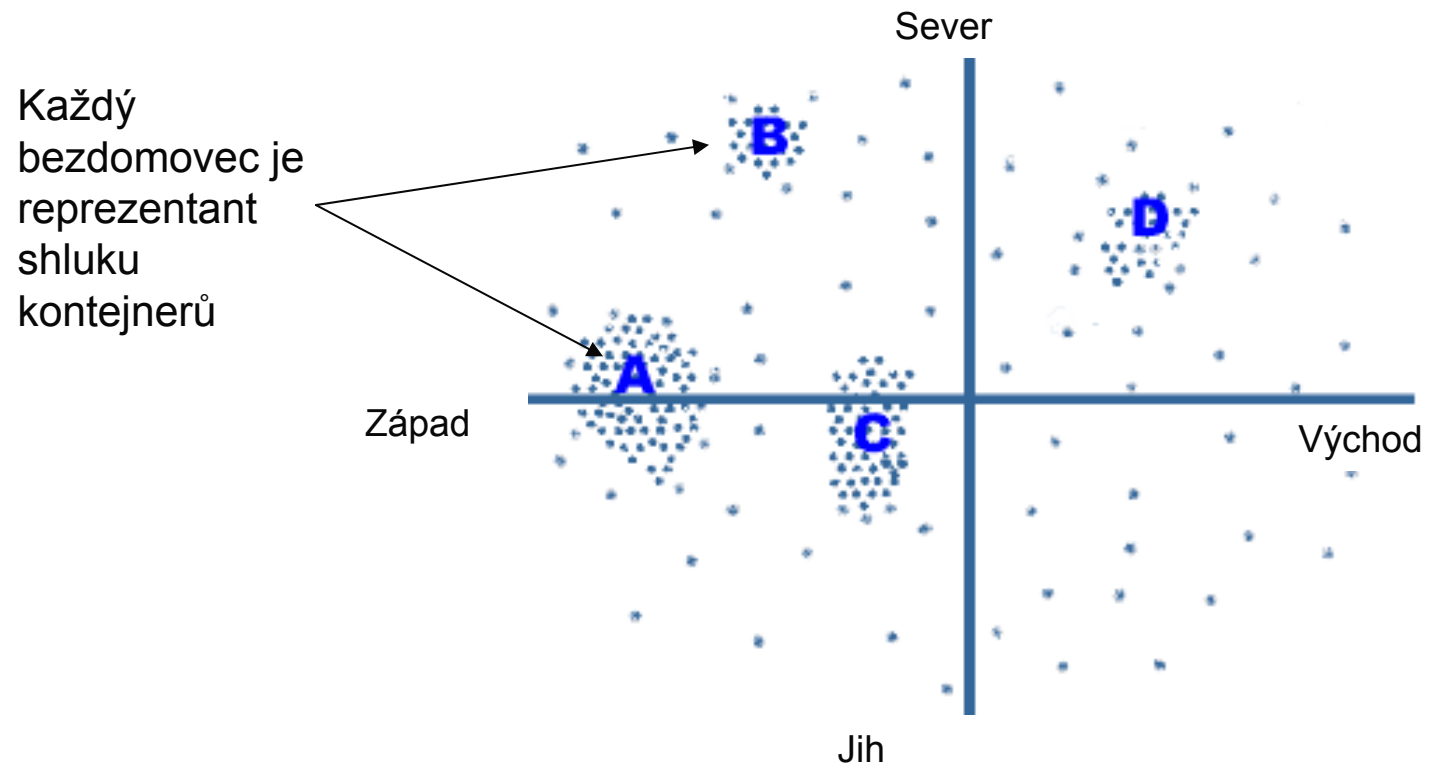
- Přírodou inspirované
- Nepotřebují žádného arbitra, který by jedincům stále říkal, kam mají jít – **učení bez učitele**
- Jedinci se učí z příkladů
- Systém se v průběhu času organizuje sám – **samoorganizuje**
- A teď to aplikujeme na shlukovou analýzu

Shluková analýza a kompetice

- Prozradí nám něco pozice bezdomovců o rozmístění kontejnerů?
- Co to je shluk? Množina bodů, které jsou si blízko a mají daleko k ostatním.
- Co to znamená „daleko“? Metriky – viz. minulá přednáška
- Bezdomovec – reprezentant shluku kontejnerů

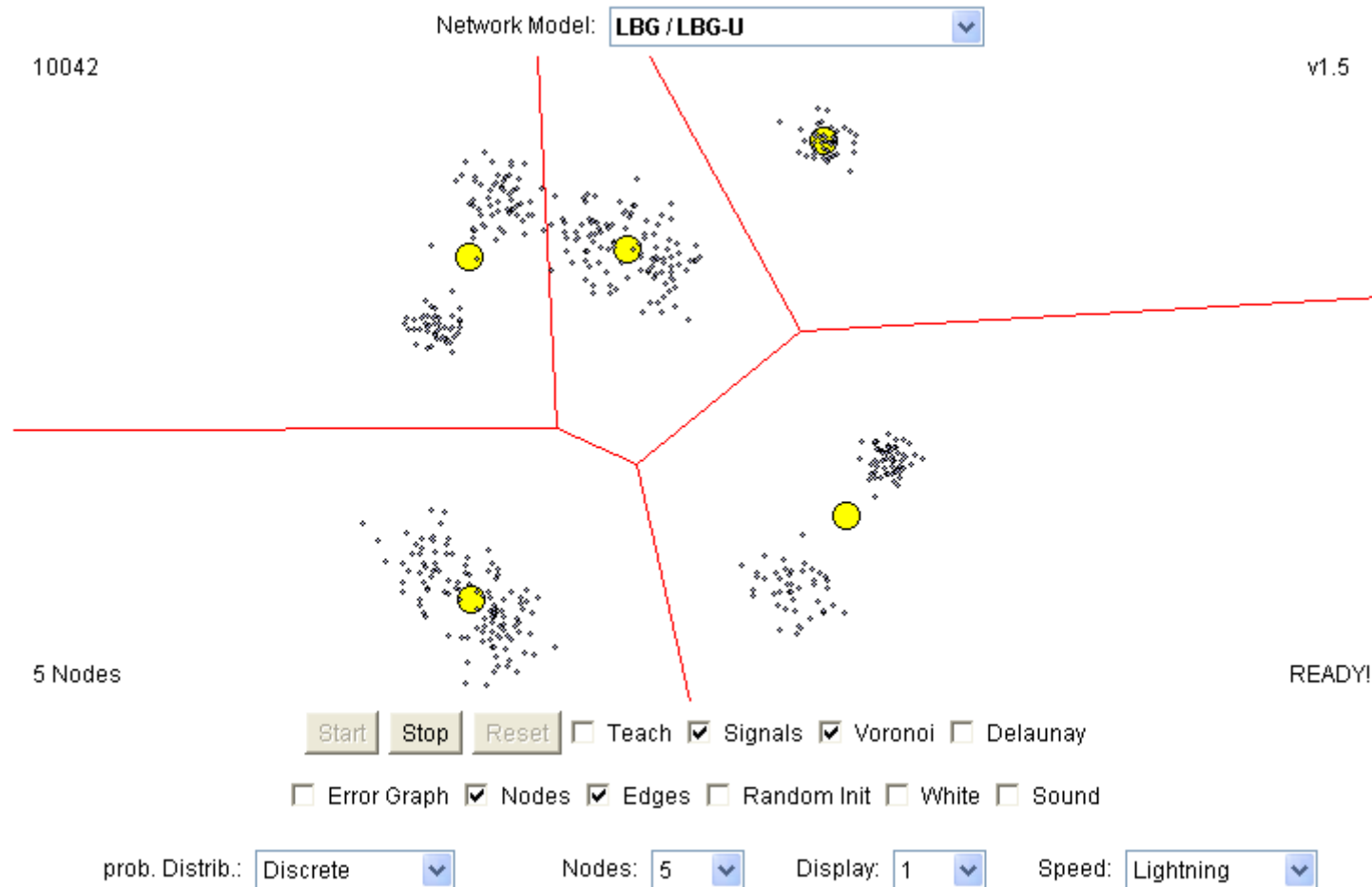
Shluky, reprezentanti

- Obrázek si můžeme představit jako teritoria čtyř bezdomovců - veteránů



- Jak simulovat pohyb bezdomovců? K-means?

K-means



<http://www.neuroinformatik.ruhr-uni-bochum.de/VDM/research/gsn/DemoGNG/GNG.html>

Jiný pohled na K-means

- Středý (reprezentanti) soutěží o data
- Používá strategii vítěz bere vše (Winner Takes All)
- Všechno jídlo zkonzumuje bezdomovec, který ke kontejneru dorazí první
- Chyba se počítá jako $E = \sum_{l=1}^K \sum_{x_i \in X_l} \|x_i - \mu_l\|^2$
- Nebo také $E = \frac{1}{2} \sum_{i=1}^K \sum_{j=1}^N 1_{WTA}(x_j, \mu_i) \cdot \|x_j - \mu_i\|^2$
Kde funkce 1_{WTA} je 1, pokud je i střed nejblíže j tému vzoru, 0 jinak
- Tato chyba se také jmenuje *kvantizační*

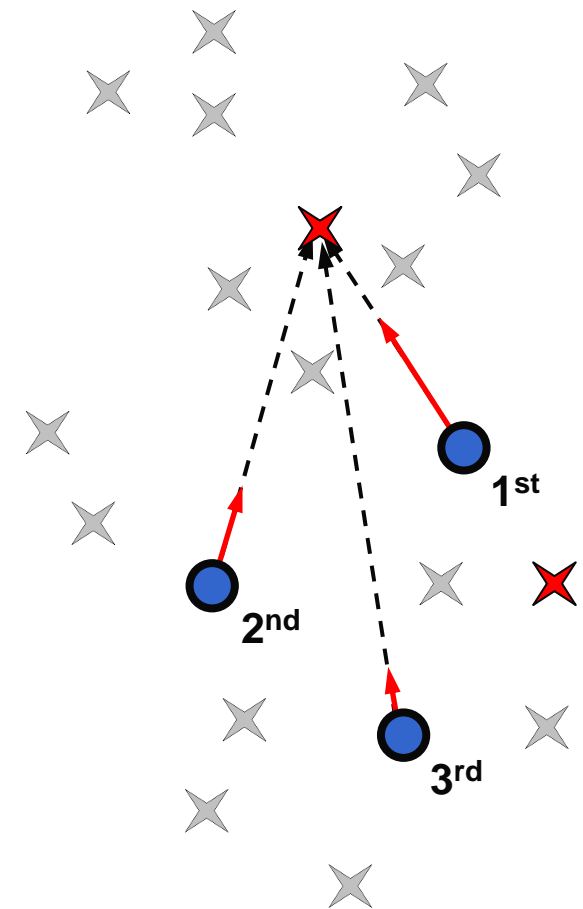
Co to je vektorová kvantizace?

- Cílem kvantizace vektorů (Vector Quantization) je aproximovat hustotu pravděpodobnosti $p(x)$ rozložením reálných vstupních vektorů $x \in \mathbf{R}^n$ pomocí konečného počtu reprezentantů $\mathbf{w}_i \in \mathbf{R}^n$.
- Tedy přesně to, o co se snažíme.

Problém se komplikuje

- Co se stane, když bezdomovec nestihne kontejner vybrat celý?
- Zbytek dostanou nejbližší
- Neplatí vítěz bere vše!
- Okolí = definuje vzdálenost, ze které se ještě vyplatí přijít.

Velmi malé okolí - vítěz bere vše
Velmi velké okolí - komunismus



Neuronový plyn (perestrojka 😊)

- Název berte z rezervou, nepracuje se s neurony, ale spíše s agenty (středky).
- Pseudokód:
 - Náhodně inicializuj středky, zvol velké okolí
 - ▶ ▪ Předlož vektor x_j
 - Pro všechny středky
 - Spočítej pořadí vzdáleností od vektoru
 - Uprav vzdálenosti $(x_j - \mu_i)$ v závislosti na pořadí a velikosti okolí - (exp)
 - Přemísti středky
 - opakuj (s menším okolím)

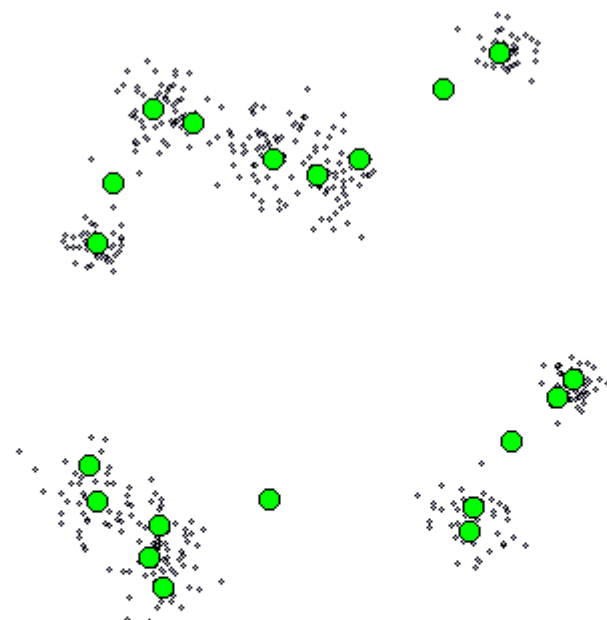
Neuronový plyn

1714

Network Model: **Neural Gas**

v1.5

20 Nodes



Teach Signals Voronoi Delaunay

Error Graph Nodes Edges Random Init White Sound

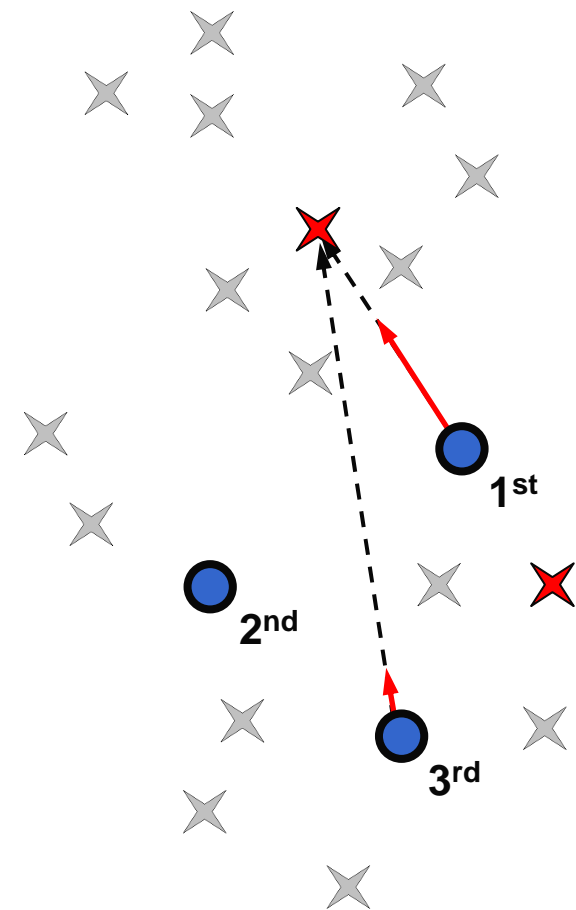
prob. Distrib.: **Discrete** Nodes: **20** Display: **1** Speed: **Lightning**

lambda_i	lambda_f	epsilon_i	epsilon_f	t_max
30.0	0.5	0.3	0.1	1000.0

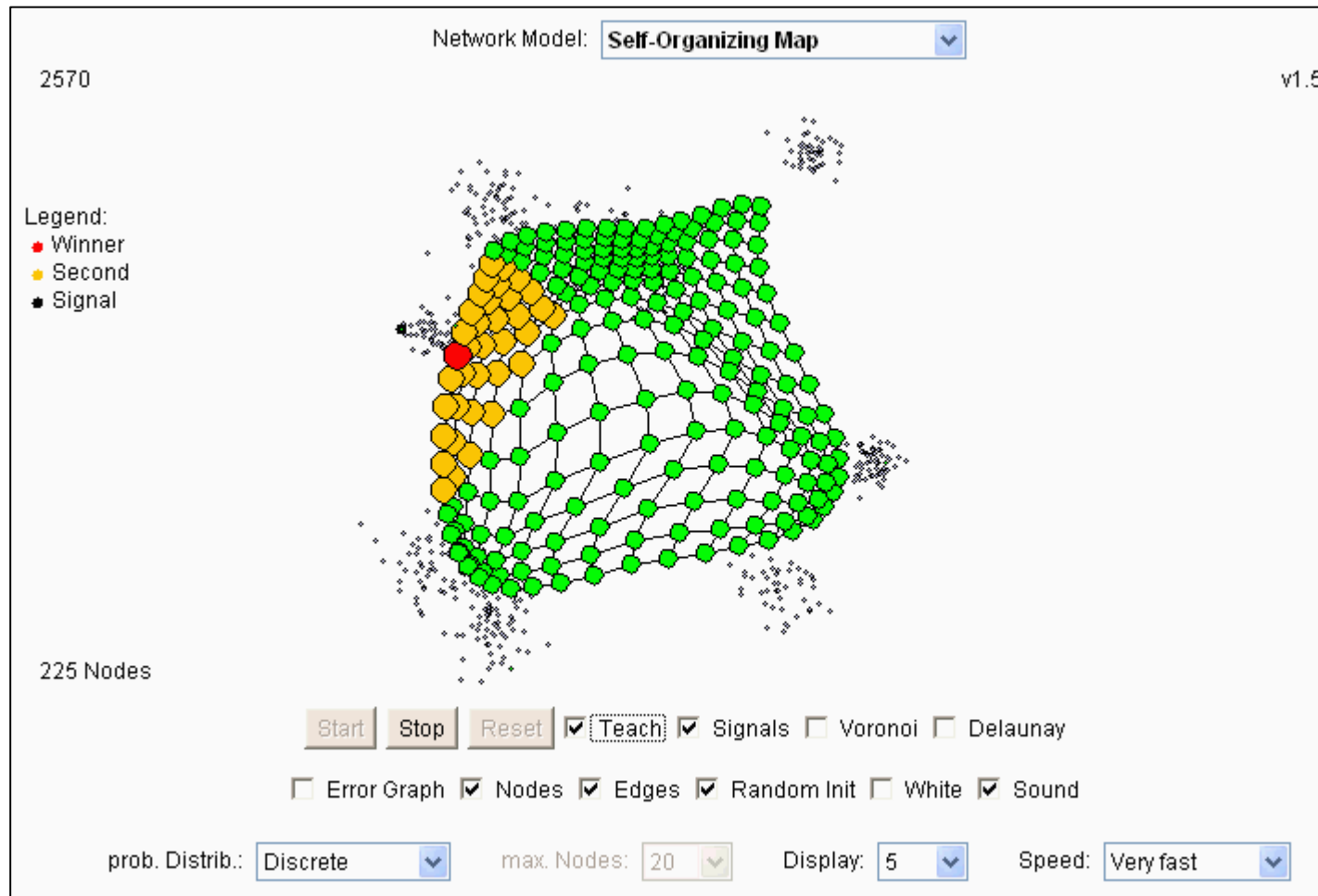
Problém se stále komplikuje

- Co se stane, když se vítězný bezdomovec rozdělí jen s kamarády?
- Okolí již neudává vzdálenost v původním prostoru dat, ale v prostoru kamarádství.

Velmi malé okolí - individualisté
Velmi velké okolí - hippies



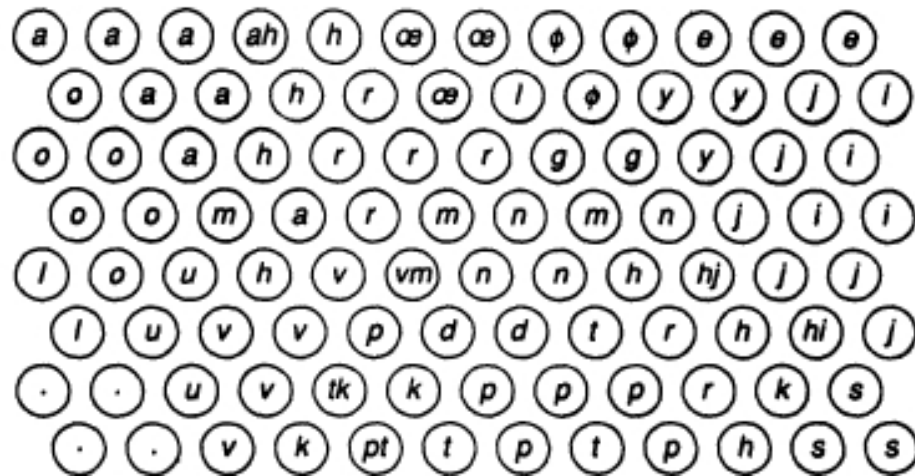
Samooorganizující se mapa (SOM)



Kamarádství pro jednoduchost znázorněno mřížkou

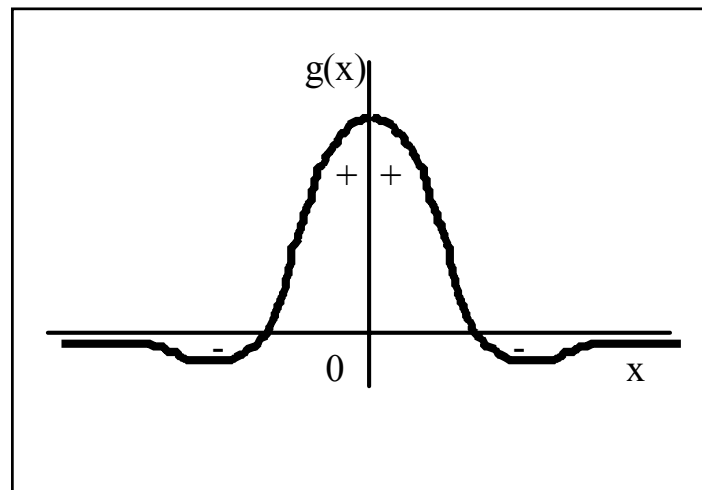
SOM?

- SOM = Self Organizing Maps,
- Prof. Teuvo Kohonen, Finsko,
- TU Helsinki, 1981, od té doby se eviduje několik tisíc vědeckých literárních odkazů.
- Původní aplikace: fonetický psací stroj.



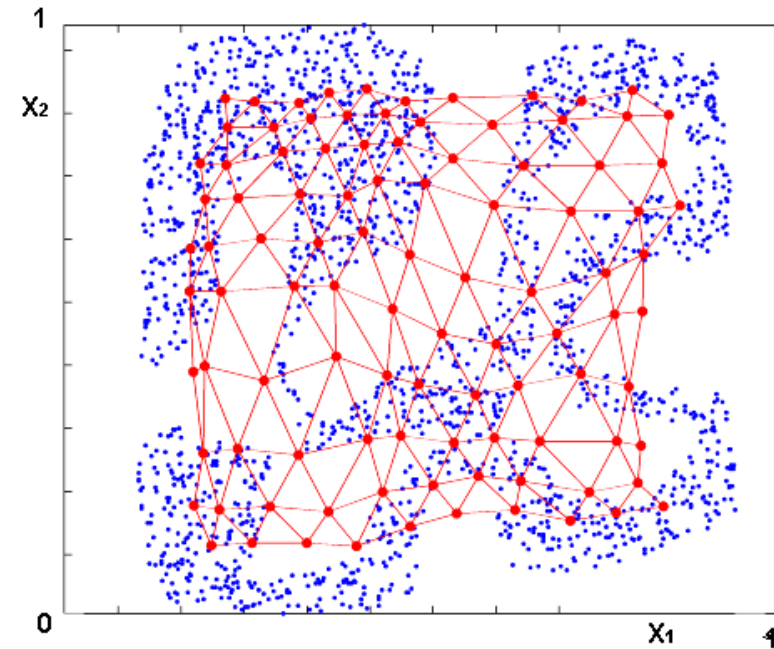
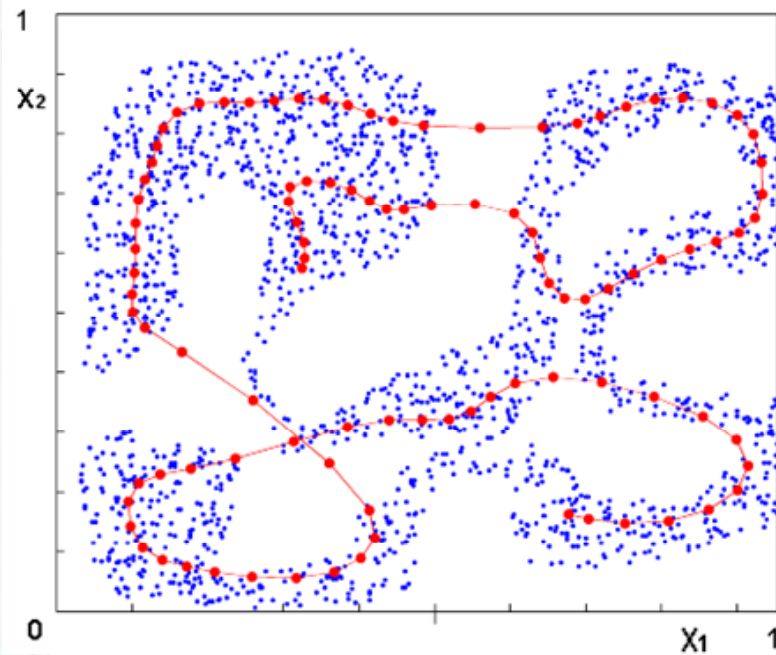
Jak se liší od k-means?

- Existence okolí!

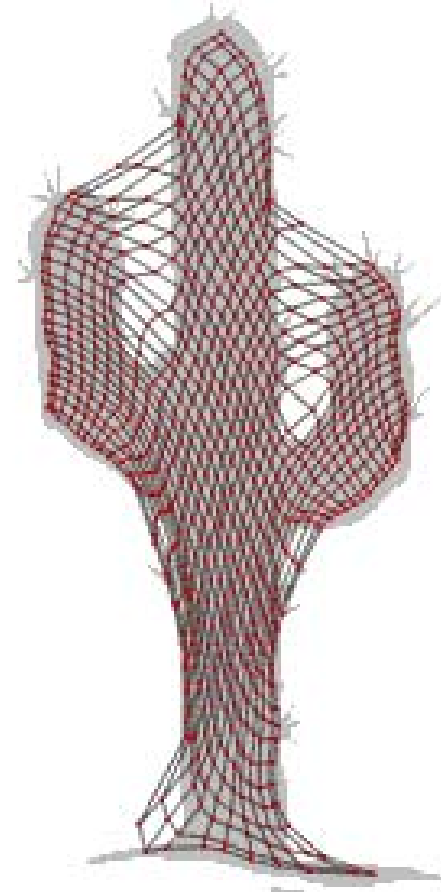
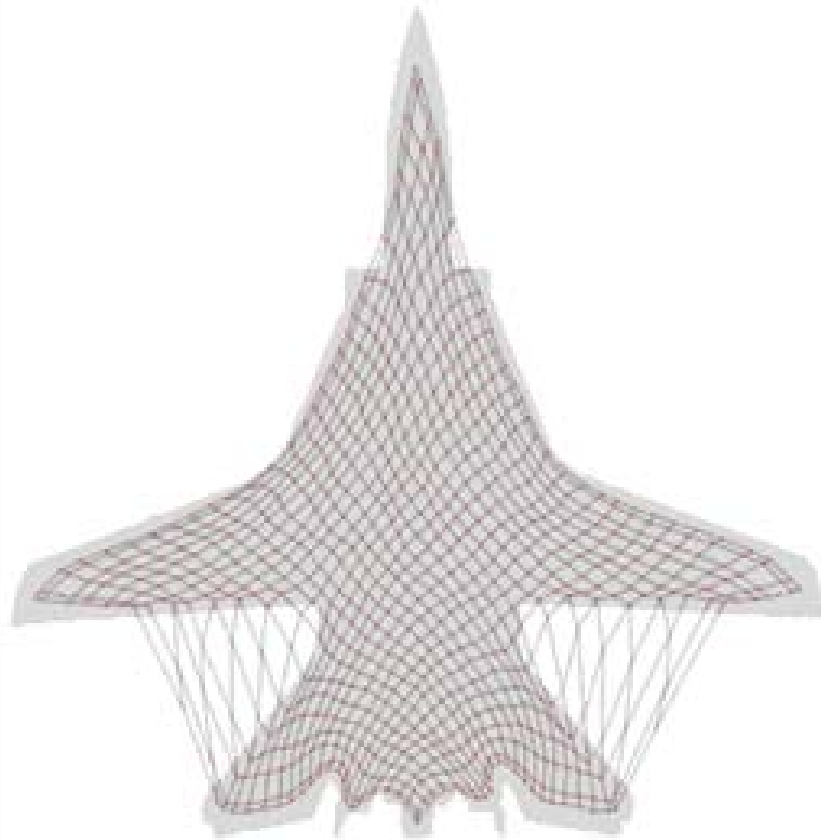


- Používá se
 1. při učení,
 2. někdy k určování vítěze.

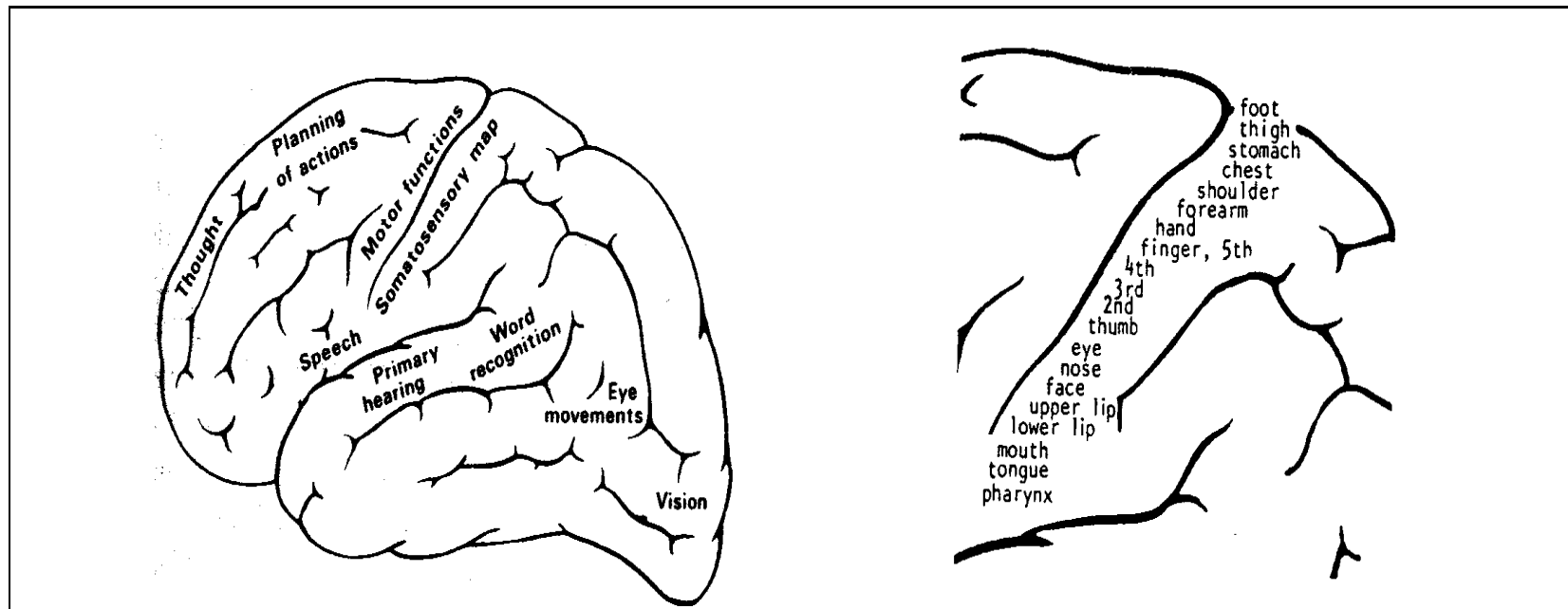
Visualizace pokrytí vstupního prostoru SOMem 1D- a 2D-



Nebo třeba ...



SOM inspirace

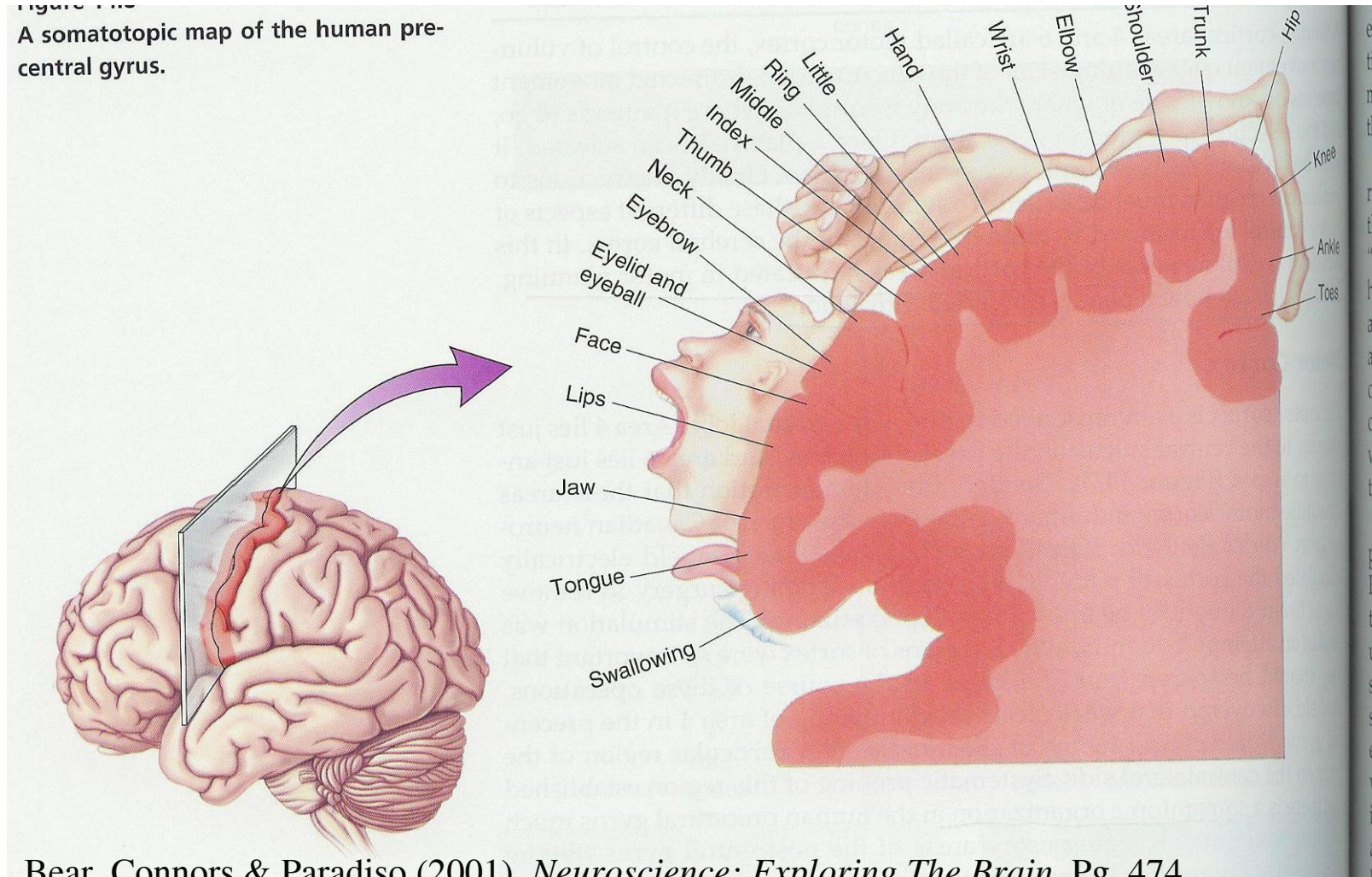


Ne bezdomovci, ale mozek.

Řídící centra souvisejících orgánů spolu sousedí.

Jiný pohled

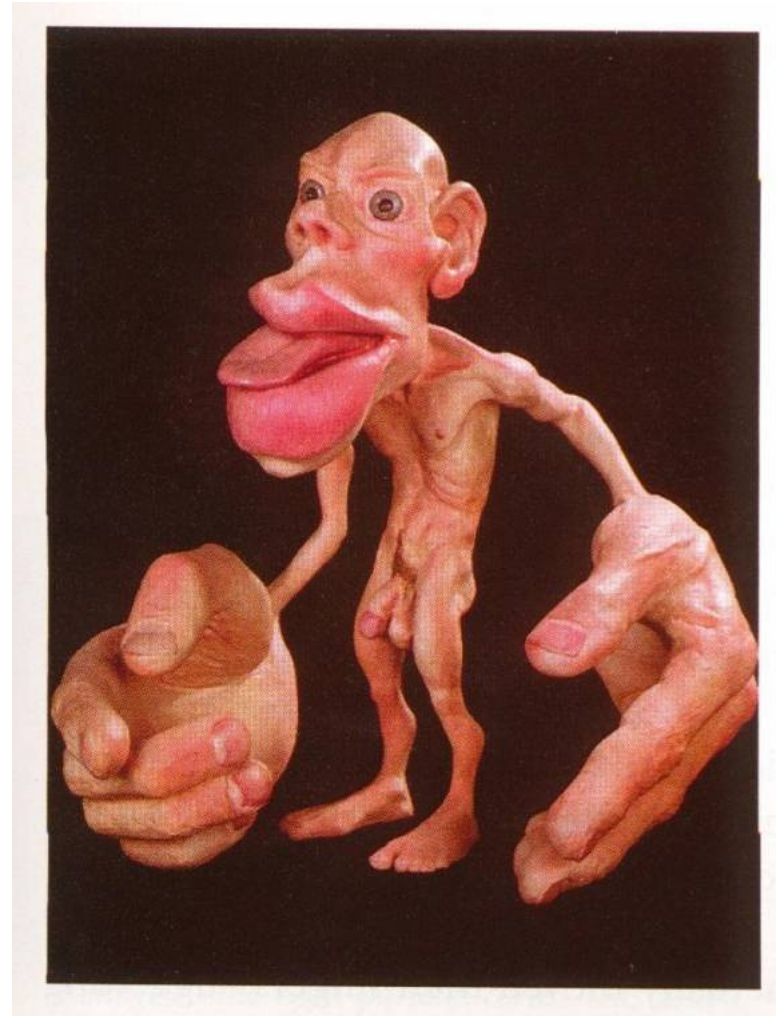
Figure 11.12
A somatotopic map of the human pre-central gyrus.



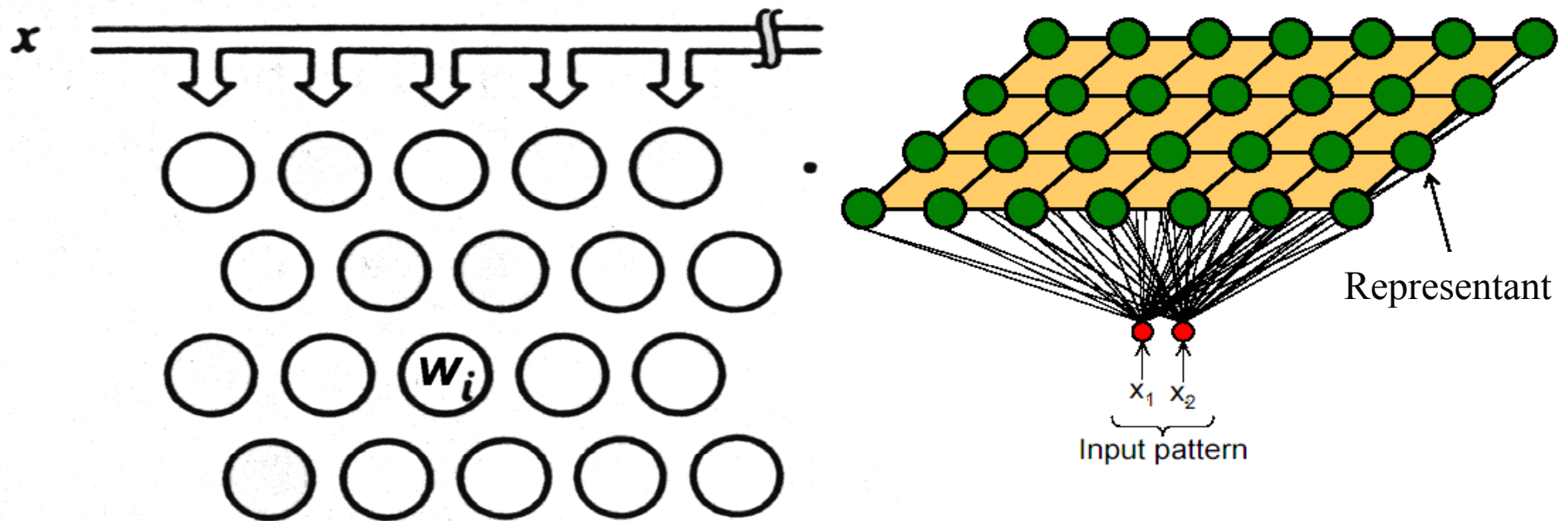
Bear, Connors & Paradiso (2001). *Neuroscience: Exploring The Brain*. Pg. 474.

Somatosensory Man

Velikost orgánů škálována
v poměru velikostí
příslušných řídicích
center v
somatosensorickém
kortexu



SOM architektura 1/3



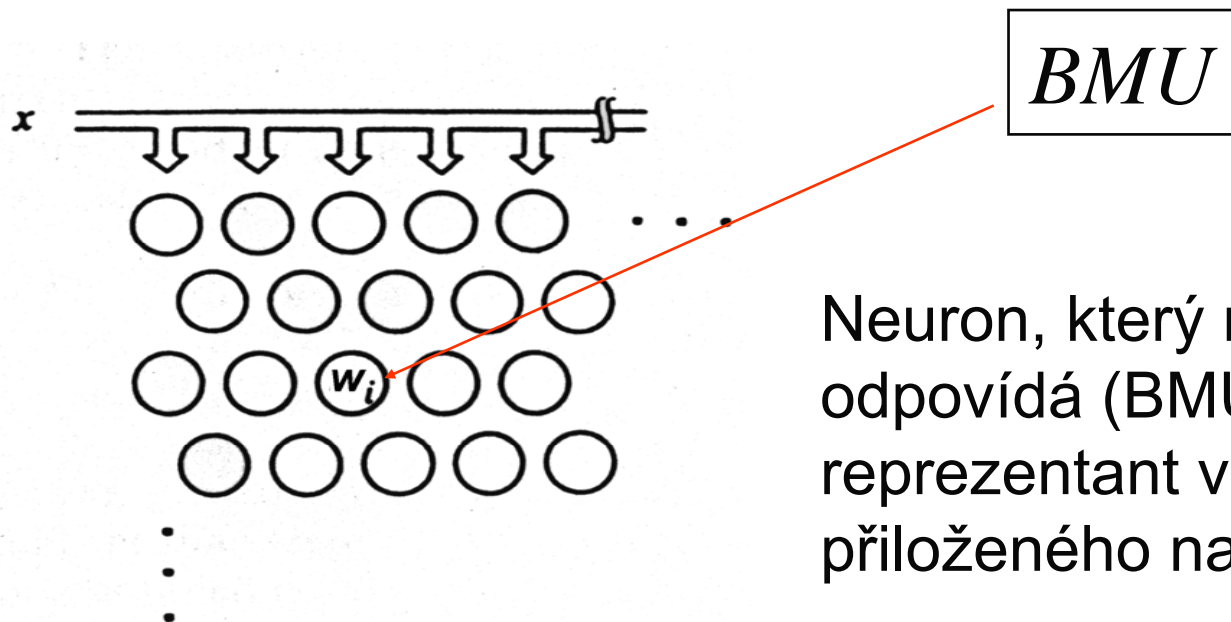
-
- Typicky: 2D pole reprezentantů
- (také se jim nepřesně říká neurony)

SOM architektura 2/3

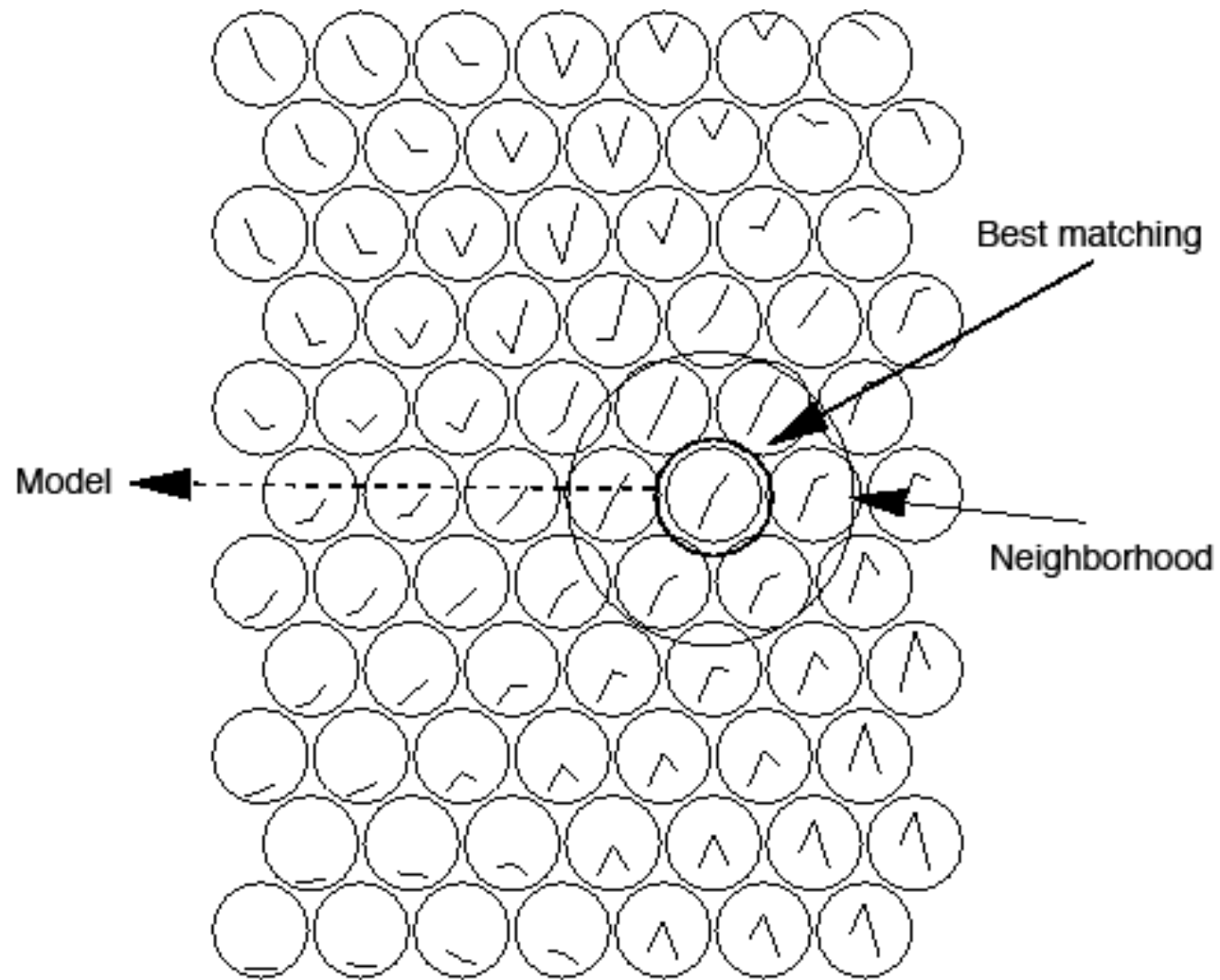
- 2D- uspořádání (do mřížky) je nejtypičtější.
- Neurony lze ale uspořádat lineárně (1D- dost často), nebo prostorově (3D- velmi výjimečně).
- Uspořádání slouží k tomu, aby měl neuron definované **sousedy** ve svém okolí.
- Kohonenovo doporučení: **obdélníková SOM!**

SOM architektura 3/3

Každý neuron má vektor vah w , vektory se porovnávají se vstupním vektorem x , vybírá ten se nejpodobnější



Neuron, který nejlépe odpovídá (BMU) je reprezentant vektoru přiloženého na vstup



Timo Honkela (Description of Kohonen's Self-Organizing Map)

SOM neuron 1/2

- Vyhodnocuje podobnost předloženého vstupního vektoru x od (ve vahách w_i) zapamatovaného, reprezentanta, referenčního vektoru.
- Podobnost = např. Eukleidovská vzdálenost:

$$j^* = \arg \min_i \{ \|x - w_i\| \},$$

- SOM neuron je tedy reprezentantem shluku.

Učení Kohonenovy sítě 1/3

- Nezapomeňte: učicí algoritmus uspořádává neurony v mřížce tak, aby reprezentovaly předložená vstupní data.
- Otázka k přemýšlení: co se děje s vahami neuronů v průběhu času?

Učení Kohonenovy sítě 2/3

1. Inicializace,
2. předložení vzoru,
3. výpočet vzdálenosti,
4. výběr nejpodobnějšího neuronu,
5. přizpůsobení vah,

$$w_{ij}(t + 1) = w_{ij}(t) + \eta(t)[x_i(t) - w_{ij}(t)]$$

6. goto 2.
- Rozumíte vzorci? Váhy jakých neuronů se přizpůsobují?

Příklad

$$\mathbf{X} = \begin{bmatrix} 0.52 \\ 0.12 \end{bmatrix}$$

$$\mathbf{W}_1 = \begin{bmatrix} 0.27 \\ 0.81 \end{bmatrix}$$

$$\mathbf{W}_2 = \begin{bmatrix} 0.42 \\ 0.70 \end{bmatrix}$$

$$\mathbf{W}_3 = \begin{bmatrix} 0.43 \\ 0.21 \end{bmatrix}$$

$$d_1 = \sqrt{(x_1 - w_{11})^2 + (x_2 - w_{21})^2} = \sqrt{(0.52 - 0.27)^2 + (0.12 - 0.81)^2} = 0.73$$

$$d_2 = \sqrt{(x_1 - w_{12})^2 + (x_2 - w_{22})^2} = \sqrt{(0.52 - 0.42)^2 + (0.12 - 0.70)^2} = 0.59$$

$$d_3 = \sqrt{(x_1 - w_{13})^2 + (x_2 - w_{23})^2} = \sqrt{(0.52 - 0.43)^2 + (0.12 - 0.21)^2} = 0.13$$

Vyhrál třetí neuron – je nejbliže

Příklad ...

Přiblížím ho ke vzoru

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)[x_i(t) - w_{ij}(t)]$$

$$\Delta w_{13} = \eta(t)(x_1 - w_{13}) = 0.1(0.52 - 0.43) = 0.01$$

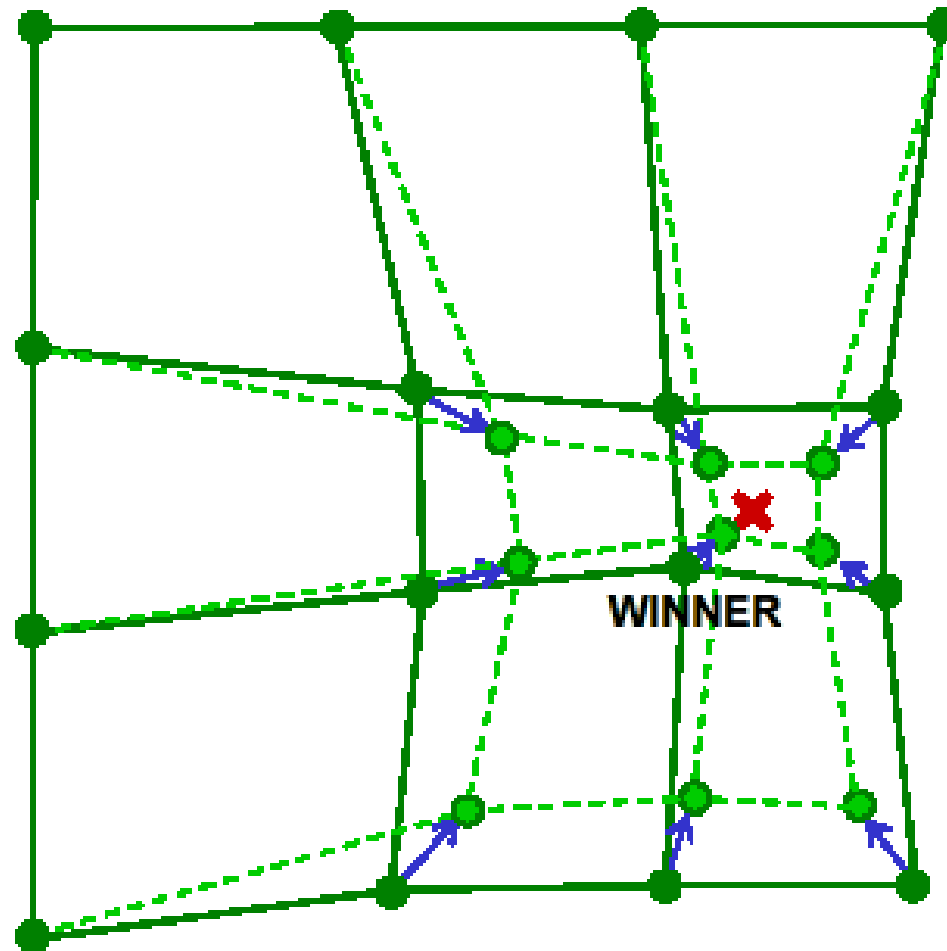
$$\Delta w_{23} = \eta(t)(x_2 - w_{23}) = 0.1(0.12 - 0.21) = -0.01$$

$$\mathbf{W}_3(p+1) = \mathbf{W}_3(p) + \Delta \mathbf{W}_3(p) = \begin{bmatrix} 0.43 \\ 0.21 \end{bmatrix} + \begin{bmatrix} 0.01 \\ -0.01 \end{bmatrix} = \begin{bmatrix} 0.44 \\ 0.20 \end{bmatrix}$$

Upravil jsem váhy pouze BMU – vítěznému neuronu

Zde tedy vítěz bere vše!

Takhle to vypadá, když updatují také okolní neurony



Učení Kohonenovy sítě 3/3

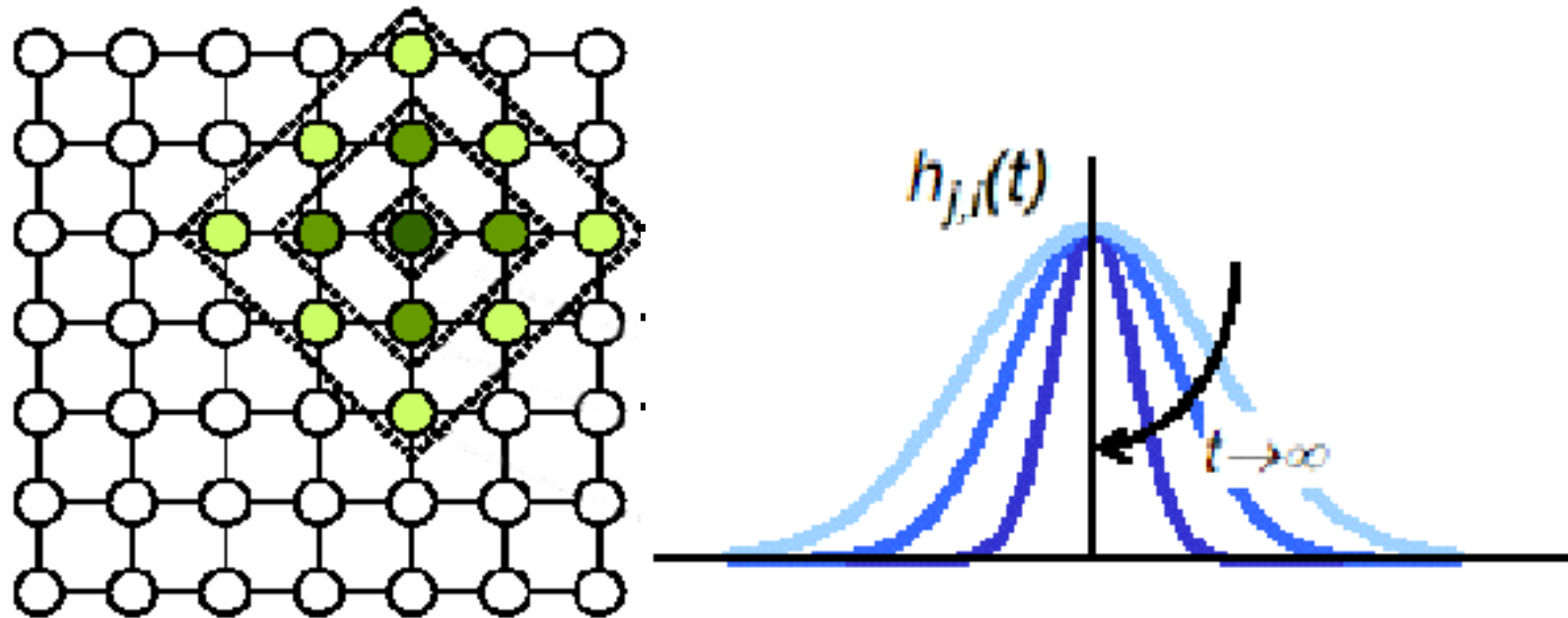
- Velkou roli při učení hraje okolí:
 - topologické uspořádání,
 - vzdálenost susedů.
- Okolí se v čase mění:
 - jeho „průměr“ s časem klesá (až k nulovému).
- změna se realizuje sdruženým učícím parametrem $\eta(t)$.

Příklad okolí: Gaussovské

$$\eta_{ij^*}(t) = \alpha(t) \exp\left(\frac{\|r_{j^*} - r_i\|^2}{2\sigma^2(t)}\right),$$

- Člen $\alpha(t)$ představuje učicí krok,
- druhý člen pak tvar okolí (v tomto případě Gaussova křivka s proměnným tvarem v čase).

Příklad okolí: Gaussovské



Distance related learning

Visualizace: klasická SOM

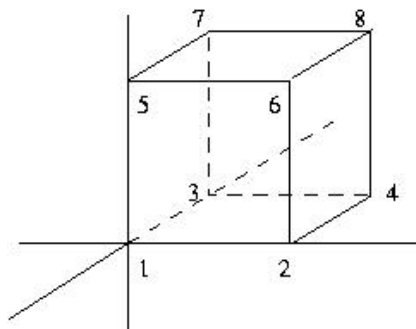
- Problém jak zobrazit pozici neuronů (reprezentantů)
- Dimenze vah = dimenze vstupního vektoru
- Potřebuji zobrazit ve 2D, jak?
 - U-matice,
 - analýza hlavní komponenty (PCA),
 - Sammonova nelineární projekce.

U-matrice (Unified distance)

- Matice vzdáleností mezi váhovými vektory jednotlivých neuronů, typicky se vizualizuje, vzdáleností vyjádřeny barvou – světlá barva = malá vzdálenost.
- Zobrazuje strukturu vzdáleností v prostoru dat.
- Poloha BMU odráží topologii dat.
- Barva neuronu je vzdálenost je váhového vektoru od všech ostatních váhových vektorů
- Tmavé váhové vektory jsou vzdáleny od ostatních datových vektorů ve vstupním prostoru.
- Světlé váhové vektory jsou obklopeny cizími vektory ve vstupním prostoru.
- Kopce oddělují clustery (údolí).

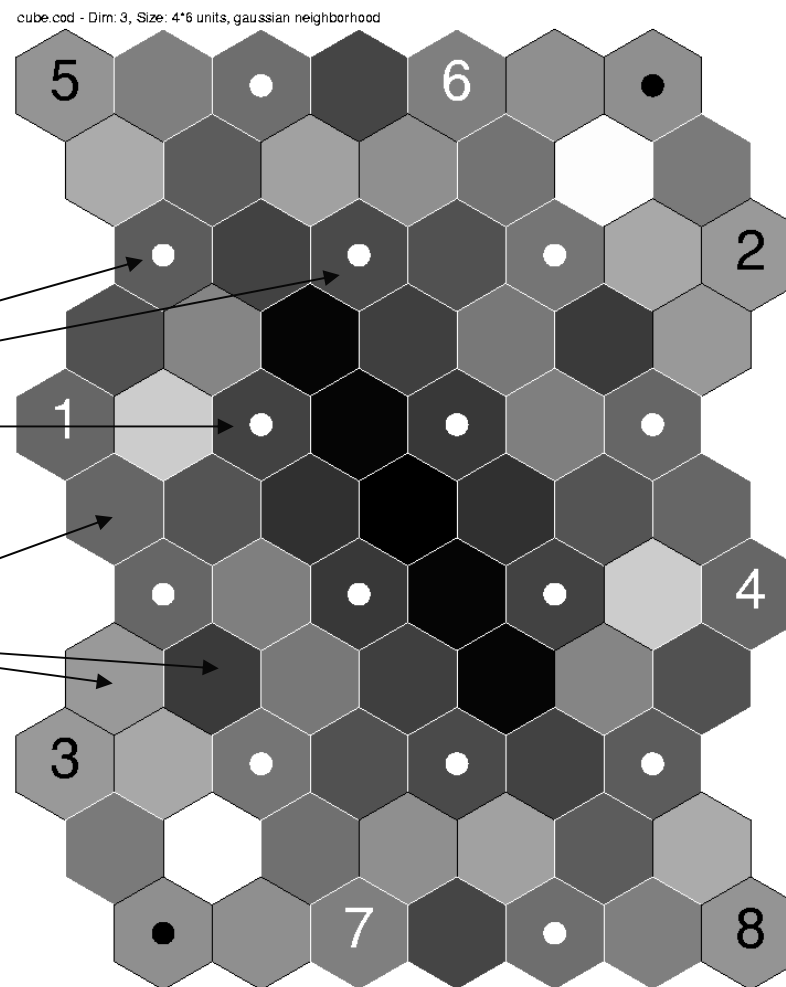
Příklad U-matice

■ Data:



■ Neurony

■ Vzdálenosti mezi
sousedními neurony



P-matrix (Pareto density estimation)

- Zobrazuje počet datových vektorů ze vstupního prostoru, které patří do koule kolem jeho váhového vektoru (s poloměrem nastaveným podle Paretova pravidla).
- Odráží hustotu dat.
- Neurony s velkou hodnotou jsou umístěny do hustých oblastí vstupního prostoru.
- Neurony s malou hodnotou jsou "osamělé" ve vstupním prostoru.
- "údolí" oddělují clustery ("náhorní plošiny").
- Doplnuje informace získané z U-matice.

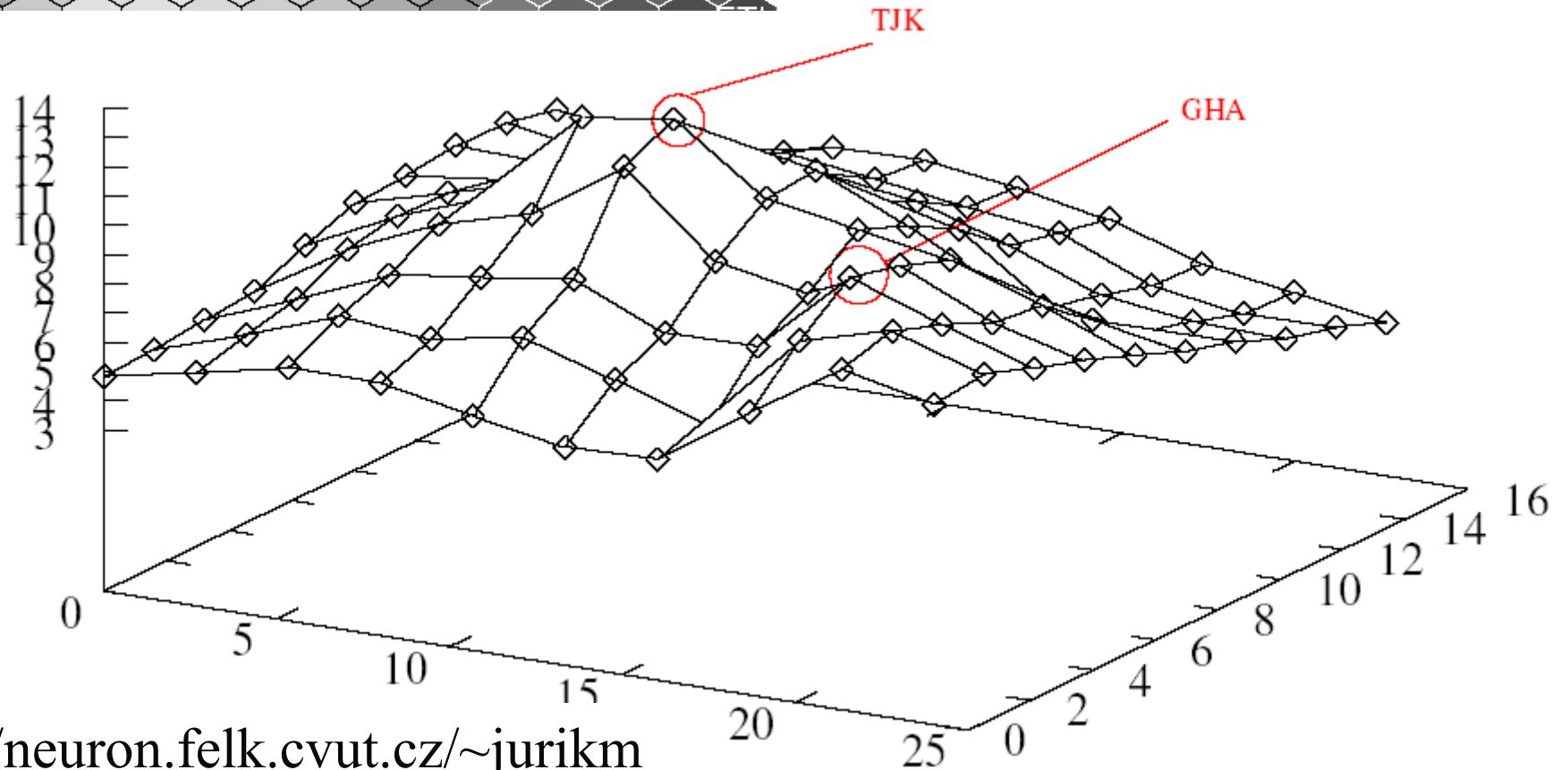
U*-Matrix

- Kombinace U-Matice a P-Matice
- Je to U-matice, korigovaná hodnotami v P-matici.
- Vzdálenosti mezi sousedními neurony (neurony a a b v mřížce) jsou vypočítány z U-matice a jsou váženy hustotou vektorů kolem neuronu a .

DP – Milan Juřík



Excitace neuronů při předložení TJK na vstup – UMAT a 3D mapa

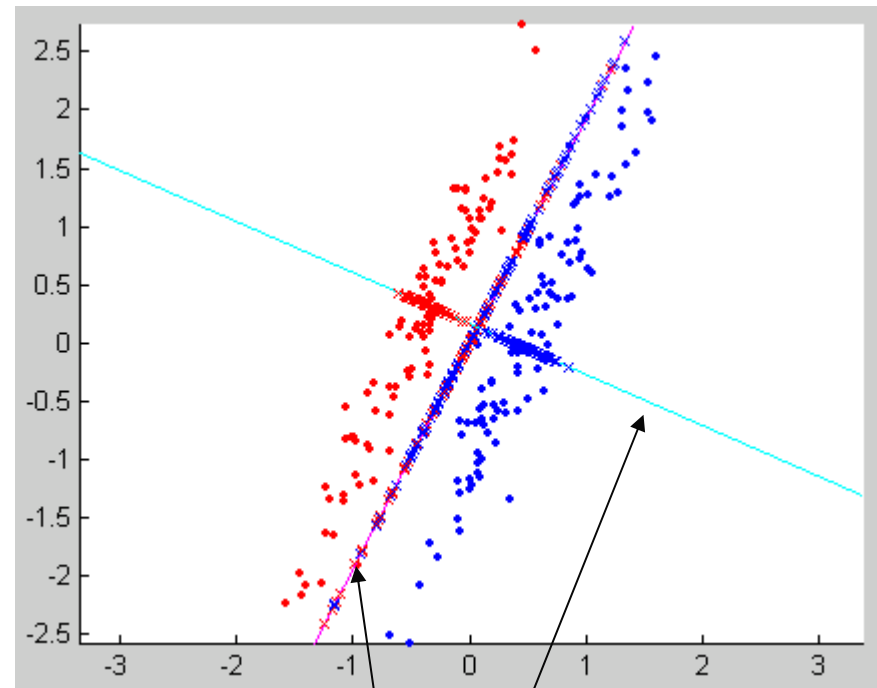


Nevýhody UMAT, PMAT, ...

- Zobrazují jen vzdálenosti mezi sousedy
- Při novém naučení sítě na stejných datech můžou vypadat jinak (můžou být např otočeny o 90 stupňů)
- Nejsou intuitivně interpretovatelné, pokud nevíte co přesně je barvou kódováno.
- Jak ale zobrazit n-rozměrná data ve 2D, abychom pokud možno zachovali originální vzdálenosti?

PCA nebo LDA

- Nové souřadnice vzniknou jako lineární kombinace původních dimenzí
- Algoritmus se jmenuje Analýza hlavní komponenty (Principal Component Analysis)



špatná

dobrá

Sammonova projekce

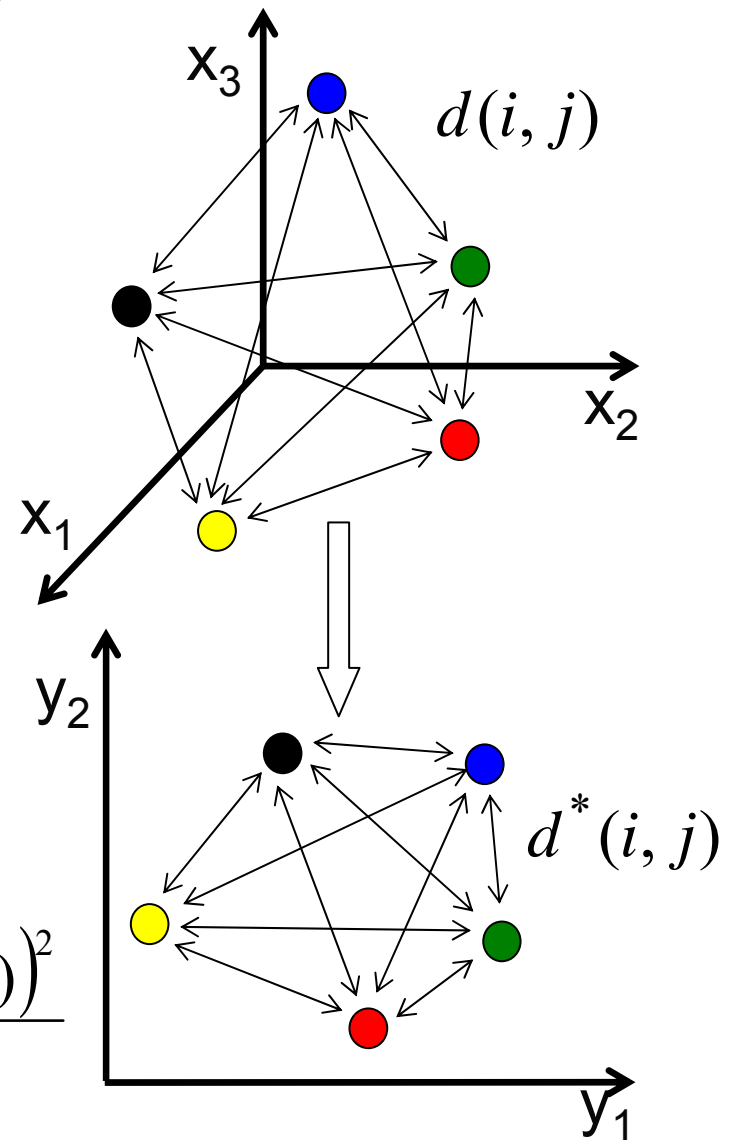
- Mějme N vektorů v L -dimenzionálním prostoru, které označme x_i , $i = 1, \dots, N$. K nim necht' patří N dvoudimenzionálních vektorů označených y_i , $i = 1, \dots, N$. Označme dále vzdálenost mezi vektory x_i a x_j v L -dimenzionálním prostoru D_{ij} a vzdálenost odpovídajících si vektorů y_i a y_j symbolem d_{ij} . Potom Sammonova projekce mapuje vstupní prostor na výstupní na základě minimalizace této chybové funkce:

$$E_{sam} = \frac{1}{\sum_{i < j} D_{ij}} \sum_{i < j}^N \frac{(d_{ij} - D_{ij})^2}{D_{ij}} \dots$$

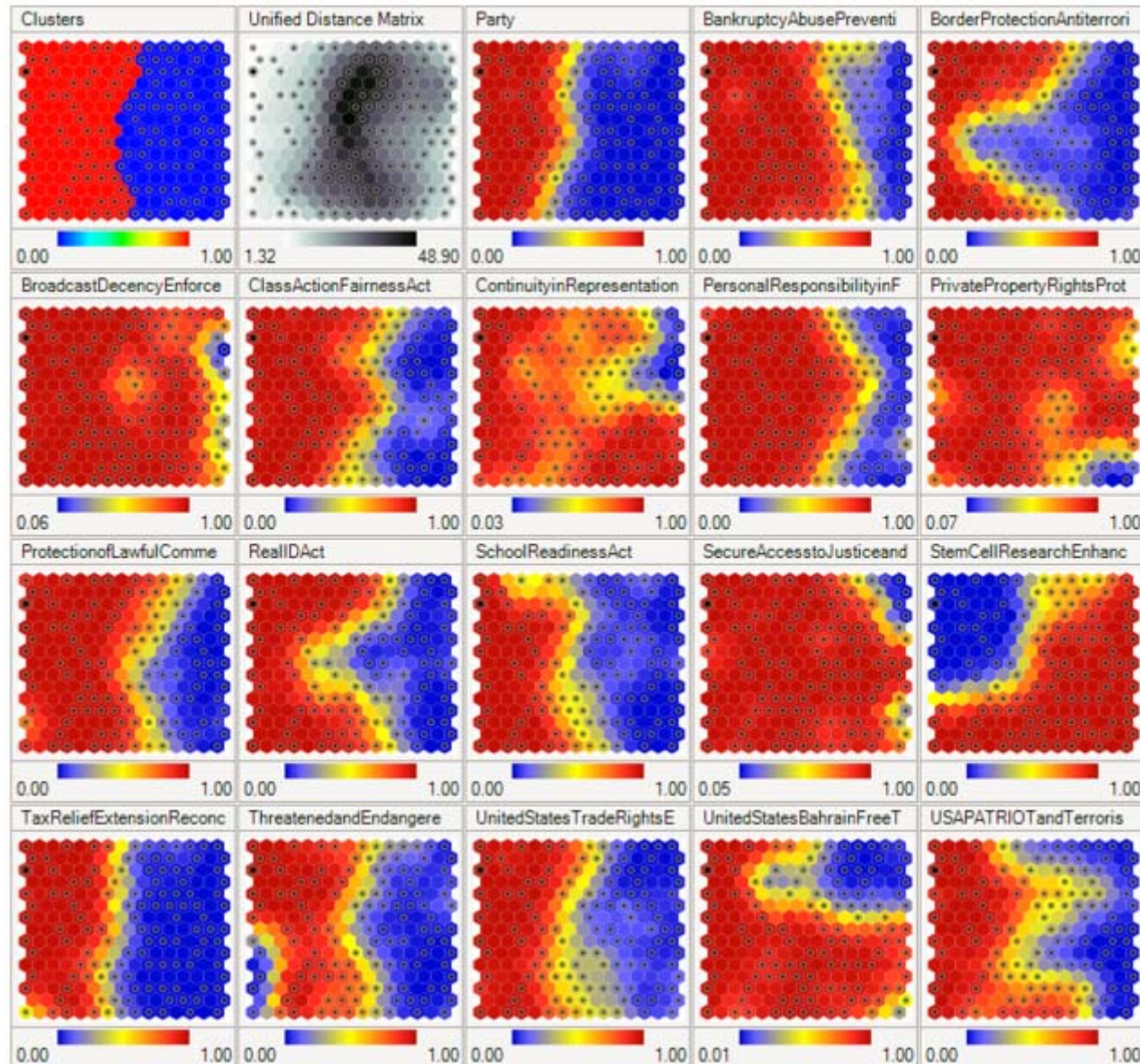
Sammon leképezés

- Távolságtartó leképezés
 - adatpontok közötti távolságok
- A Sammon stress célfüggvényt minimalizálja
 - Nemlineáris optimalizálási feladat
- $N(N-1)/2$ távolság kiszámítása minden iterációs lépésben

$$E = \frac{1}{\sum_{i=1}^{N-1} \sum_{j=i+1}^N d(i, j)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{(d(i, j) - d^*(i, j))^2}{d(i, j)}$$



Příznakové grafy (feature plots)



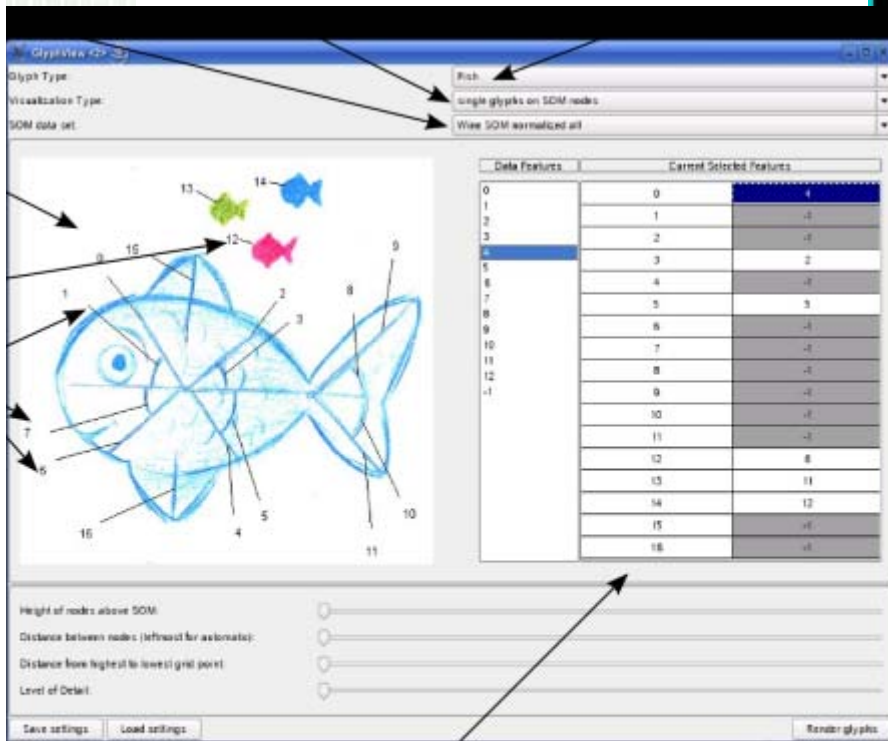
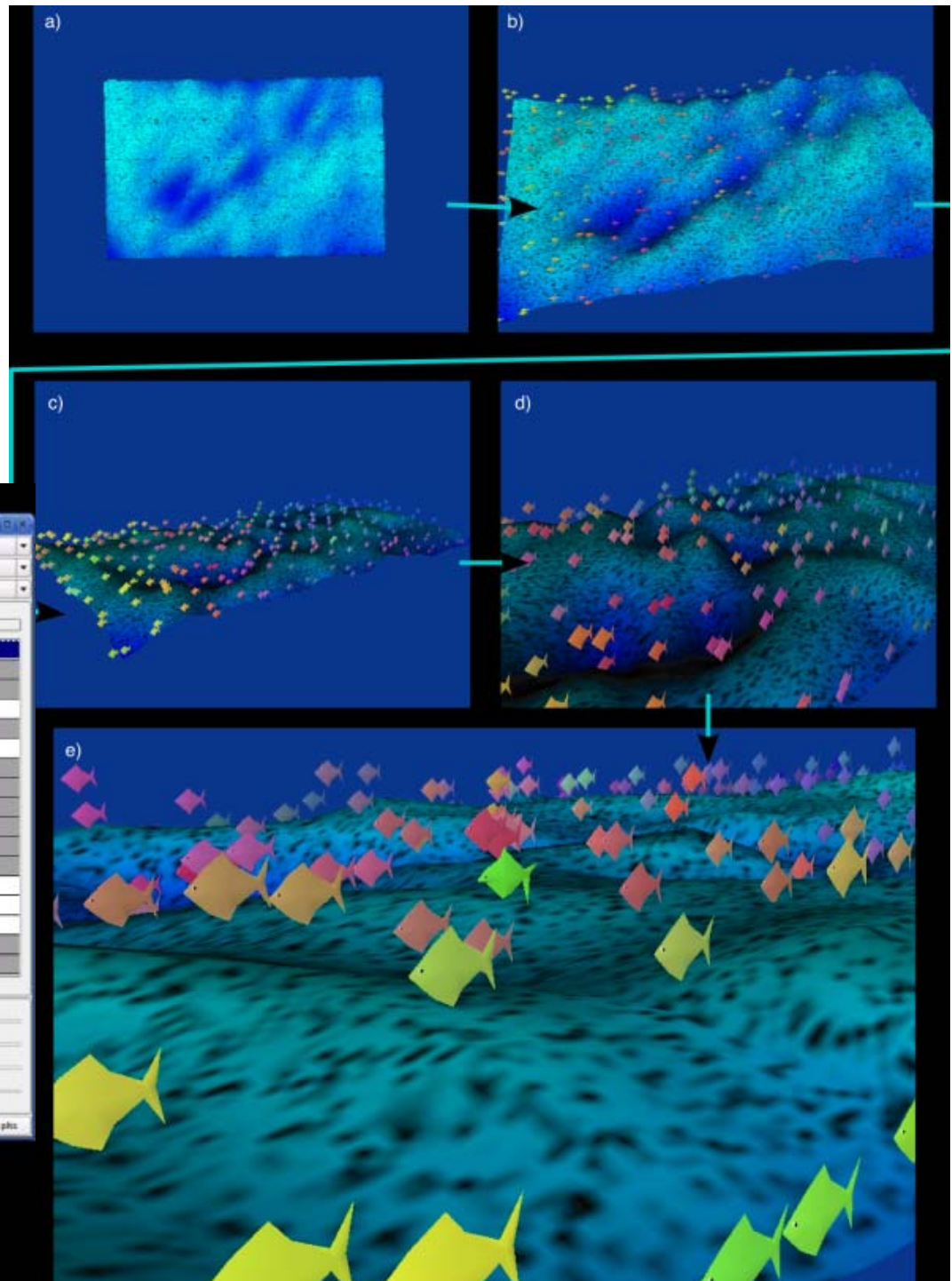
Aplikace SOMU

- <http://www.generation5.org/content/2007/kohonenImage.asp>



ReefSOM

- <http://www.brains-minds-media.org/archive/305>



Whizzo's PicSOM

Training settings

Texture and Color settings

Texture Histogram Color Histogram

Texture Area Color Area

Color Kinds

RGB Grey

HSL


Initialization

Training iterations: 100 Map rows: 10 Map cols: 10

Gradient initialization

Input dir: D:\Other\A.I.\fobpro


Preview Thumb



Controls

Select input dir Train SOM Provide image

Results



D:\Other\A.I.\fobpro\vela07... D:\Other\A.I.\fobpro\vela07...

D:\Other\A.I.\fobpro\vela07... D:\Other\A.I.\fobpro\vela07...

Node distribution

3	1	3	1	3	3	1	4	0	4
0	1	2	1	0	1	0	1	0	2
3	1	0	0	3	5	0	4	0	2
1	0	5	2	0	0	1	1	0	1
5	0	0	0	4	2	0	5	0	3
2	0	4	0	1	1	0	1	0	1
4	0	1	0	4	1	3	0	1	5
1	1	5	2	2	0	2	0	1	1
2	1	0	1	0	2	0	2	0	3
4	1	3	3	2	3	1	3	0	3

Neural Network is trained.

SOM vlastnosti

- VQ – vektorová kvantizace, více vektorů se mapuje do jednoho neuronu (jeho váhového vektoru), jak přesně? -> **kvantizační chyba**.
- Komprese dimenze vstupního prostoru.
- **Zachování topologie dat** – sousední (ve vstupním prostoru) vektory se mapují do sousedních (v mřížce) neuronů, jak kvalitně? -> **topografická chyba**.
- SOM má energetickou funkci, kterou minimalizuje -> **zkreslení**.

Zkreslení SOM

- Průměrná vzdálenost mezi každým datovým vektorem a jeho BMU.
- Určuje přesnost mapování (vektorové kvantizace) – už známe
- \mathbf{c}_i je váhový vektor neuronu
- \mathbf{m}_j je vektor dat
- h_{bij} je funkce okolí

$$E_d = \sum_i^{\mathcal{N}} \sum_j^{\mathcal{M}} h_{bij} \|\mathbf{c}_i - \mathbf{m}_j\|$$

Topografická chyba SOM

- Počet vstupních vektorů, pro které vítězný neuron a druhý vítězný neuron nejsou sousedi v mřížce.
- $u(c_i)$ je 1, když sousedi nejsou, jinak 0
- V procentech počet vzorků, u nichž nebyla zachována

$$\epsilon_t = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} u(\mathbf{c}_i)$$

Software:

- Zajímavý a hlavně použitelný SW
SOM_PAK:

- http://www.cis.hut.fi/research/som_pak
- <http://service.felk.cvut.cz/courses/36NAN>
+ český návod na ovládání

- **Matlab SOM toolbox**

- <http://www.cis.hut.fi/projects/somtoolbox/>

- **SOMPAK addon**

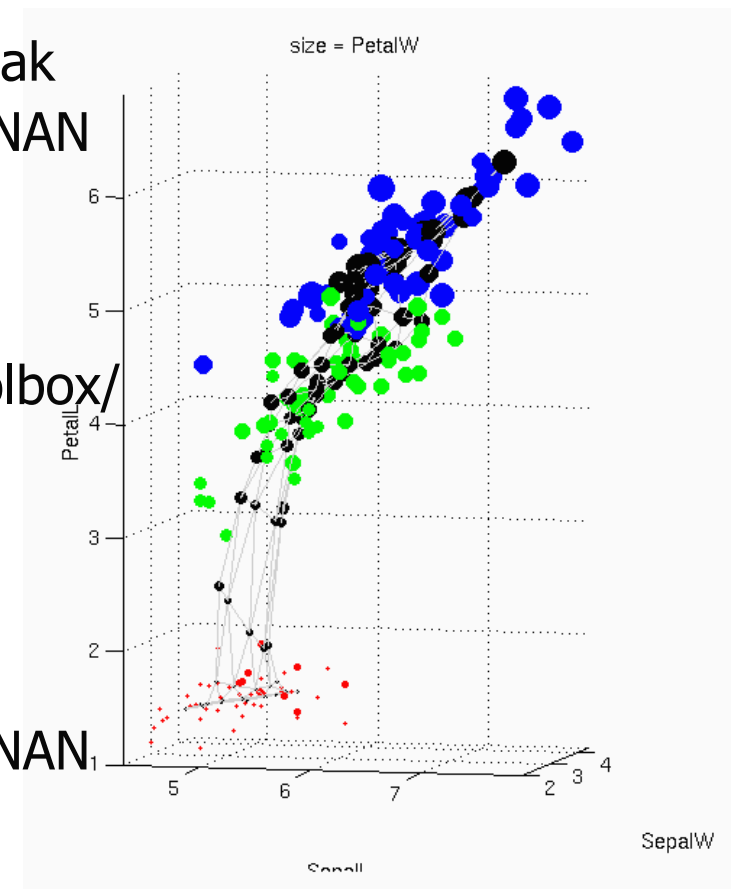
- <http://neuron.felk.cvut.cz/~jurikm>

- **Zooming SOM**

- <http://service.felk.cvut.cz/courses/36NAN>

- **TKM, RSOM**

- <http://service.felk.cvut.cz/courses/36NAN>



Další slajdy pro zájemce

- Náš výzkum – THSOM (Jan Koutník)

<http://cig.felk.cvut.cz/people/koutnik/>

- Jak se pozná, že je síť správně naučena?

http://www.cis.hut.fi/somtoolbox/package/docs2/som_quality.html

Viz další slajdy ...

Literatura

- Ultsch, A. (2003), *Maps for the Visualization of high-dimensional Data Spaces*, <http://www.mathematik.uni-marburg.de/~databionics/downloads/papers/ultsch03maps.pdf>
- Kohonen, T. (1995), *Self-Organizing Maps*, 2nd ed., Springer-Verlag, Berlin, 1995, pp. 113
- SOM Toolbox Manual, <http://www.cis.hut.fi/somtoolbox/package/docs2/somtoolbox.html>