učení sítí lze klasifikovat následovně – **s učitelem** (např. BP, učení perceptronu; při učení jsou k dispozici dvojice ⟨vstup, výstup⟩ výstup ~ správná odpověď ~ učitel (angl. supervised learning)

– **bez učitele** (unsupervised learning) info o správném výstupu není k dispozici (nebo nemá smysl)

↓ příkladem: samoorganizující se sítě

charakteristickým rysem samoorganiz. sítí je tzv. kompetiční učení (soutěžní, competitive):

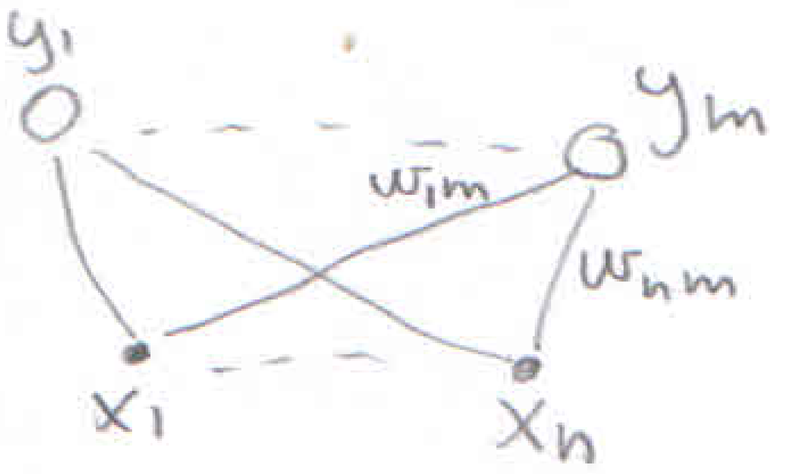
- po předložení vstupu výstupní neurony "soutěží" o to, který bude aktivní (vždy jen jeden), vítězný neuron tzv. rozpoznal vstupní vzor (vstupní signál)
- poprvé popsáno v 60. letech 20. stol.
- existuje řada modelů, nejvýznamnější jsou Kohonenovy mapy
- neurofyziologické výzkumy potvrzují, že některé části mozku pracují na principu soutěžního učení (termín grandmother cell – neuron, který rozpoznává babičku)

Základní úkol: – dána trénovací množina $T = \{ \mathbf{x}^p \in \mathbb{R}^n \mid p \in P \}$ vzorů (obecněji může být známo rozdělení pravděpodobnosti na vstupech) a parametr m (počet reprezentantů ~ výstupních neuronů)

– reprezentant $j = 1, \ldots, m$ je dán vektorem $\mathbf{w}_j = \langle w_{1j}, \ldots, w_{nj} \rangle \in \mathbb{R}^n$

někdy codebook words; v našem pohledu váhy spojení

schéma ~ kompetiční neur. sítě



– jsou-li už nějak $\mathbf{w}_j$ určeny a je-li $\mathbf{x} \in \mathbb{R}$ vstup, je mu přiřazen reprezentant c, který je mu nejblíže:

$$c = \arg \min_{j=1}^{m} \| \mathbf{x} - \mathbf{w}_j \| \quad , \text{ kde např. } \| \mathbf{x} - \mathbf{w}_j \| = \sqrt{\sum (x_i - w_{ij})^2}$$

Eukl. norma

– cílem je najít m reprezentantů, pro které je

$$E = \frac{1}{|P|} \cdot \sum_{p \in P} \| \mathbf{x}^p - \mathbf{w}_c \|^2 \quad \text{minimální.}$$

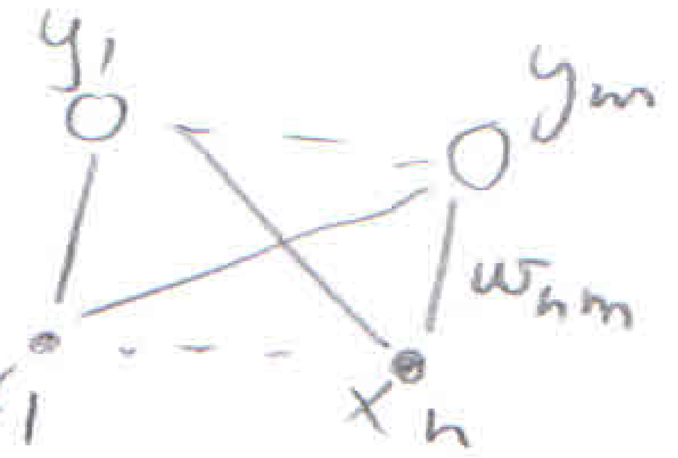
Lloydův algoritmus učení

- inicializuj $\mathbb{W}_j$ náhodně
- opakuj následující, dokud chyba neklesne pod zadanou ε (přesnost)
  - projdi T a pro každé $x_p \in T$ urči příslušný vítězný neuron $\mathbb{W}_c$
  - pro $j = 1, \dots, m$ (výst. neurony) urči T_j = množina vstupů z T, pro které byl j vítězem, tj. $T_j = \{ p \in P \mid j = \arg\min_j \| x_p - w_j \| \}$
  - $t_j = \frac{1}{|T_j|} \sum_{x \in T_j} x$ ... těžiště T_j
  - $\mathbb{W}_j \leftarrow t_j$

ke změně reprezentantů dojde až po průchodu celou T na pomalé, náhodné; osvědčila se následující on-line verze, která mj. odpovídá představě neurofyziologie

Kohonenovo učení

tedy, v konceptech neur. sítí:



aktivační dynamika: - předložen vstup $x \in \mathbb{R}^n$

- výstupy $y_j = \begin{cases} 1 & \text{pokud } j = \arg\min_{e=1}^{m} \| x - \mathbb{W}_e \| \text{ ... zahoří} \\ 0 & \text{jinak ... nezahoří} \end{cases}$

činnost toho celé, realizováno mechanismem laterální inhibice

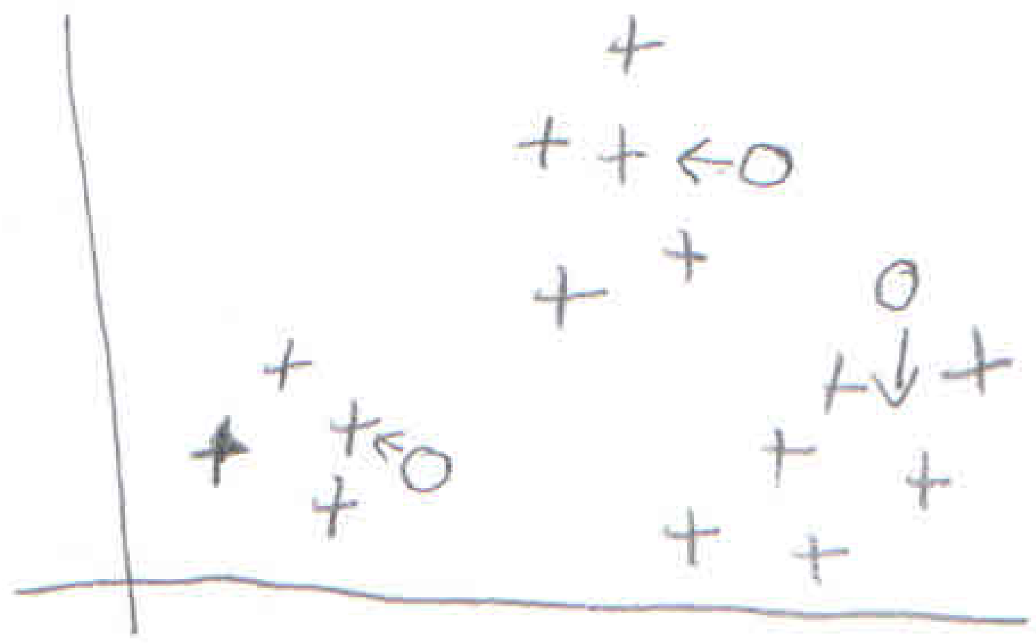
učení (adaptivní dynamika)

- procházíme T, pro každý $x \in T$ proveď pro každý $j = 1, \dots, m$

$$\mathbb{W}_j(t+1) = \begin{cases} \mathbb{W}_j(t) + \theta(x - \mathbb{W}_j(t)) & \text{pokud } j = \arg\min_e \| x - \mathbb{W}_e \| \\ \mathbb{W}_j(t) & \text{jinak} \end{cases}$$

tedy $w_{ij}(t+1) = \begin{cases} w_{ij}(t) + \theta(x_i - w_{ij}(t)) \\ w_{ij}(t) \end{cases}$

θ ... rychlost učení, $0 < \theta < 1$, v čase se zmenšuje geometrický pohled $n=2$



$+ \dots$ vzor $x^p \in T$

$o \dots$ vektory vah w_j

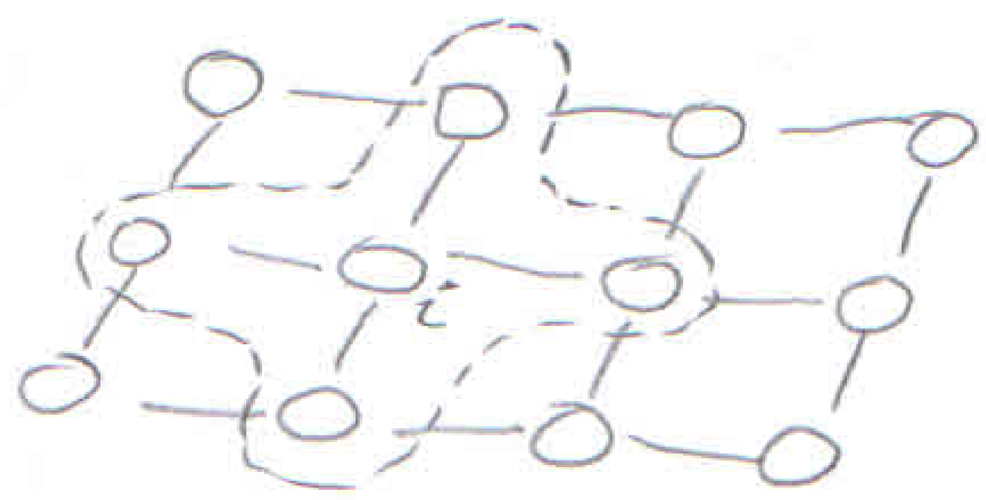
tento mechanismus je známý v literatuře o shlukování jako k-means clustering

Kohonenovy (samoorganizační) mapy

- Teuvo Kohonen 1982
- podobná výše uvedené jednoduché samoorganizační síti, neurony jsou navíc uspořádány do topologické struktury, např. 2D mřížka
- topol. struktura dána vhodnou vzdáleností neuronů; k neuronu i je dáno jeho okolí: $N_\sigma(i) = \{ j \mid d(i,j) \le \sigma \}$

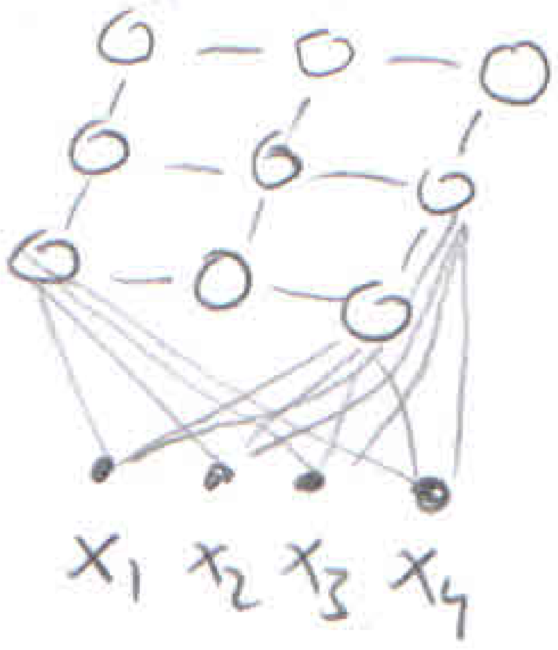
  velikost okolí — neuron — vzdálenost neuronů i a j — zá

- př.: 2D mřížka



$d(i,j)$ = délka cesty mezi i a j

- síť



každý neuron j je připojen ke všem vstupům $x_1, \dots, x_4$ váhy $w_{1j}, w_{2j}, w_{3j}, w_{4j}$

- adaptace : - jako u jednoduché samoorg. sítě, ale adaptují se váhy jak vítězného neuronu, též neuronů v jeho okolí:

$$w_j(t+1) = \begin{cases} w_j(t) + \eta(x - w_j(t)) & \text{pro } j \in N_\sigma(c) \\ w_j(t) & \end{cases}$$

kde $c = \arg\min_e \|x - w_e\|$

σ na začátku velké, postupně klesá

η na začátku ≈ 1, postupně klesá

  - popř. η závisí i na vzdálenosti j od c
  - doporučení : - aby se vytvořila rozumná struktura, doporučuje se 10 – 100 tisíc iterací
    - 2 fáze
      - hrubé učení (do 1000 iterací), větší η a σ, vytvoření globální struktury
      - jemné učení, malé hodnoty η a σ

η klesá z ≈ 1 až na 0.01

σ klesá, na zač. okolí = celá síť

- Kohonenova síť umožňuje "mapovat" body ze vstupního prostoru $\mathbb{R}^n$ (Kohonenova mapa) do 2D struktury (struktury neuronů), nebo jiné nízkodimenzionální struktury

- jiné topologie : 1D :

Samorganizující se sítě a učení s učitelem

SOS lze využít pro učení s učitelem, konkrétně pro klasifikaci dat následovně.

Je dána $T = \{\langle x^p, c^p \rangle \mid p \in P, x^p \in \mathbb{R}^n, c^p \in \{1, \dots, k\}\}$

vstup x^p má být přiřazen do třídy c^p (jedné z k tříd)

(př.: x^p popisuje parametry výrobku, $c^p = 1 \Leftrightarrow$ výrobek je ok, $c^p = 0$ výrobek ...)

Postup:

a) nejdříve uvažujeme $T' = \{x^p \mid p \in P\}$, tj. ignorujeme c^p, a vytvoříme kompetitivním učením samoorganizující se síť s n vstupy a m neurony ve výstupní vrstvě ($k \leq m$)

b) označíme výstupní neurony třídami $1, \dots, k$, do kterých se klasifikuje: každý výstupní neuron j označíme tou ~~tří~~ třídou $c(j) \in \{1, \dots, k\}$ s tou vlastností, že ze všech x^p, pro které je j vítězem, jich nejvíce patří do $c(j)$

tedy: projdeme celou T a pro každý $\langle x^p, c^p \rangle$ určíme vítězný neuron j pro vstup x^p, zapamatujeme; ...

c) doučíme síť pro T algoritmem LVQ (learning vector quantization) např. následujícím (známý jako LVQ1):

pro $\forall \langle x^p, c^p \rangle \in T$:

urči vítěze $j = \arg\min_{e} \| x^p - w_e \|$

uprav váhy: $w_j(t+1) = \begin{cases} w_j(t) + \alpha(x^p - w_j(t)) & \text{pokud } c(j) = c^p \\ w_j(t) - \alpha(x^p - w_j(t)) & \text{pokud } c(j) \neq c^p \end{cases}$

opakuj, přitom snižuj α (např. tisíce - desetitisíce iterací, během kterých α klesne z cca 0.01-0.02 k 0