ANALÝZA A KLASIFIKACE BIOMEDICÍNSKÝCH DAT

prof. Ing. Jiří Holčík, CSc.

V. NEURONOVÉ SÍTĚ III.

UČENÍ VÍCEVRSTVÉHO PERCEPTRONU
ALGORITMUS ZPĚTNÉHO ŠÍŘENÍ CHYBY

(Back Propagation Algorithm – BackPropag – BP)

- snahou je dosáhnout takového nastavení vah, aby odchylka (chyba) mezi aktuálními a požadovanými výstupy sítě byla minimální
- chyba sítě

$$E = \sum_k E_k$$

kde E_k chyba odpovídající k-tému vzoru určená podle vztahu

$$E_k = \frac{1}{2}\sum_j (y_j - d_{kj})$$

j je index pořadí neuronu ve výstupní vrstvě a d_{kj} je j-tý element požadovaného výstupu k-tého trénovacího vzoru

UČENÍ VÍCEVRSTVÉHO PERCEPTRONU
ALGORITMUS ZPĚTNÉHO ŠÍŘENÍ CHYBY

- počáteční nastavení vah na malé hodnoty se střední hodnotou cca nula; heuristicky z intervalu (-2/s, 2/s), kde s je počet vstupů neuronu;
- po předložení všech vzorů učební množiny se zadaptují váhy sítě podle vztahu
  - $w_{ij}(t+1) = w_{ij} + \Delta w_{ij}$,

kde pro Δw_{ij} je

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

($0<\eta\leq 1$ je parametr učení)

(**akumulované učení** – váhy se mění až po vyhodnocení reakce na všechny vzory)

UČENÍ VÍCEVRSTVÉHO PERCEPTRONU
ALGORITMUS ZPĚTNÉHO ŠÍŘENÍ CHYBY

- díky linearitě derivace

$$\frac{\partial E}{\partial w_{ij}} = \sum_k \frac{\partial E_k}{\partial w_{ij}}$$

- $$\frac{\partial E_k}{\partial w_{ij}} = \frac{\partial E_k}{\partial y_j}\frac{\partial y_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}}$$

- $$\frac{\partial net_j}{\partial w_{ij}} = y_i, \text{ protože } net_j = \sum_i w_{ij}.y_i$$

UČENÍ VÍCEVRSTVÉHO PERCEPTRONU

ALGORITMUS ZPĚTNÉHO ŠÍŘENÍ CHYBY

- ☑ parciální derivaci $\partial y_j/\partial net_j$ dostaneme derivací aktivační funkce (sigmoidy) a pomocí její funkční hodnoty

$$y_j = f(net_j) = \frac{1}{1+e^{-\lambda . net_j}}$$

$$\frac{\partial y_j}{\partial net_j} = \frac{\lambda . e^{-\lambda . net_j}}{(1+e^{-\lambda . net_j})^2} = \left| 1 - y_j = \frac{e^{-\lambda . net_j}}{1+e^{-\lambda . net_j}} \right| = \lambda \frac{e^{-\lambda . net_j}}{1+e^{-\lambda . net_j}} \frac{1}{1+e^{-\lambda . net_j}} = \lambda (1 - y_j) . y_j$$

UČENÍ VÍCEVRSTVÉHO PERCEPTRONU

ALGORITMUS ZPĚTNÉHO ŠÍŘENÍ CHYBY

- ☑ hodnotu parciální derivace $\partial E_k/\partial y_j$ získáme metodou „zpětného šíření chyby“ (postup odzadu je dán tím, že primární informace o chybě je pouze ve výstupní vrstvě – tam je $\frac{\partial E_k}{\partial y_j} = y_j - d_{kj}$

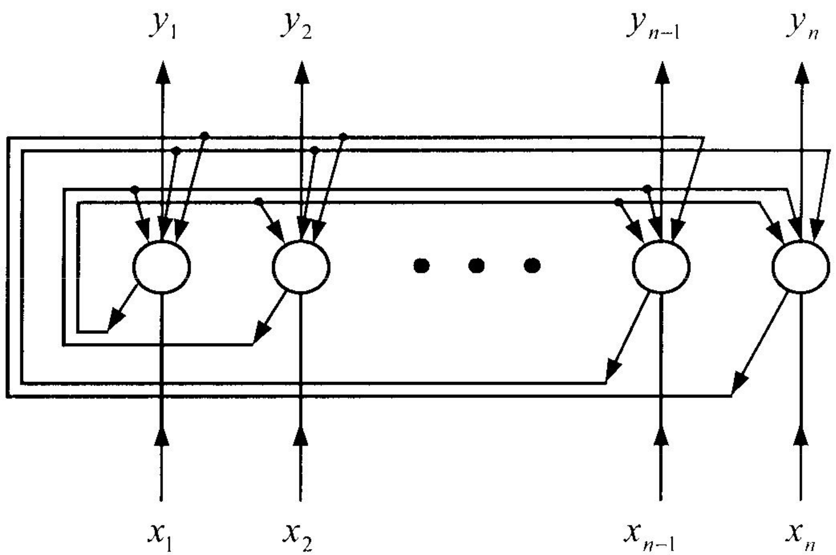
  j se mění přes neurony ve výstupní vrstvě. Derivace pro skryté vrstvy určujeme podle vztahu

$$\frac{\partial E_k}{\partial y_j} = \sum_r \frac{\partial E_k}{\partial y_r} \frac{\partial y_r}{\partial net_r} \frac{\partial net_r}{\partial y_j} = \sum_r \frac{\partial E_k}{\partial y_r} \lambda y_r (1 - y_r) w_{rj},$$

  kde index r značí všechny neurony, do nichž vede výstupu j-tého neuronu a $\partial E_k/\partial y_r$ jsou hodnoty známé z předchozího kroku výpočtu.

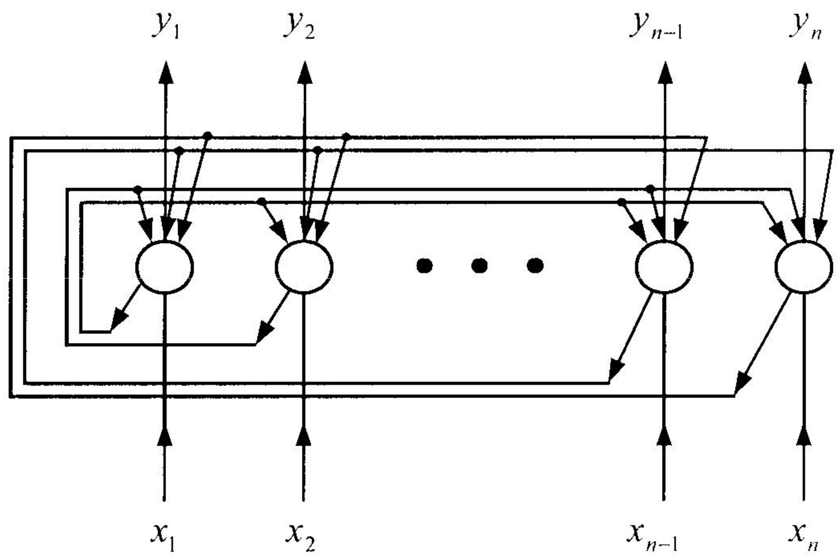
HOPFIELDOVA SÍŤ
USPOŘÁDÁNÍ

- má tolik vstupů, kolik vstupů i výstupů (každý neuron je současně vstupní i výstupní);
- výstup každého neuronu je veden přes váhy w_{ij}, i,j=1,2,…,n zpět na vstupy zbylých neuronů, tak se vytváří uzavřené smyčky



HOPFIELDOVA SÍŤ
USPOŘÁDÁNÍ

- ☑ na vstupy neuronů nejsou přiváděny vlastní výstupy
- ☑ symetrická cyklická síť s diagonálně symetrickou maticí vah s nulovou hlavní diagonálou



HOPFIELDOVA SÍŤ
USPOŘÁDÁNÍ

- hodnoty vstupů, stavů a tím i výstupů jsou bipolární (-1, +1), každý neuron počítá klasicky (váhovaný součet vstupů – váhy jsou celočíselné) svůj vlastní potenciál a aktivuje svůj výstup aktivační funkcí nejčastěji podle ostré prahové nelinearity

HOPFIELDOVA SÍŤ
UČENÍ

- ☑ jednorázový neiterační proces, při kterém pro každý vkládaný vzor vytvoříme dílčí matici n x n (n je počet neuronů) – prvky matice vzniknou vynásobením i-tého vstupu j-tým výstupem

$$w_{ij} = \begin{cases} \sum_{k=1}^{s} x_{ki} x_{kj}, & \text{pro } i \neq j \\ 0, & \text{pro } i \neq j \end{cases}$$

$$\Downarrow$$

symetrická matice s -1 a +1 s výjimkou nulové diagonály

- ☑ matice vah vznikne součtem dílčích matic všech vzorů z učební množiny

HOPFIELDOVA SÍŤ
UČENÍ

- ☑ Hebbův zákon:

  Změna váhy mezi dvěma neurony je úměrná jejich souhlasné aktivitě (lze vyjádřit součinem jejich stavů).

  Podporovány jsou souhlasné aktivity a rozdílné aktivity váhy naopak potlačují.

HOPFIELDOVA SÍŤ
VYBAVOVÁNÍ

- založeno na porovnávání vzorů pomocí Hammingovy vzdálenosti a za správnou odpověď se bere ten vzor, který má tuto vzdálenost nejmenší.
- na rozdíl od učení je vybavování iterační děj

HOPFIELDOVA SÍŤ
VYBAVOVÁNÍ

- ☑ Nejprve jsou nastaveny počáteční stavy jednotlivých neuronů podle předloženého vzoru podle vztahu

$$y_i(0)=x_i,\ i=1,2,\dots,n$$

a dále jsou tyto stavy iteračně aktualizovány podle vztahu

$$y_j(t+1)=f\left(\sum_{i=1}^{n} w_{ij} y_i(t)\right),\quad j=1,2,\dots,n$$

(f je aktivační funkce).

HOPFIELDOVA SÍŤ
VYBAVOVÁNÍ

- ☑ po každé iteraci jsou výstupy poopraveny a slouží jako vstupy do sítě.
- ☑ opakujeme tak dlouho, až se stavy (výstupy) všech neuronů během dvou po sobě následujících cyklů nezmění.

HOPFIELDOVA SÍŤ
VYBAVOVÁNÍ

Dva způsoby aktualizace stavů:

- ☑ synchronní
  - ➔ Napřed jsou všechny aktuální hodnoty stavů uschovány a pak následuje výpočet nových stavů s pomocí uložených hodnot

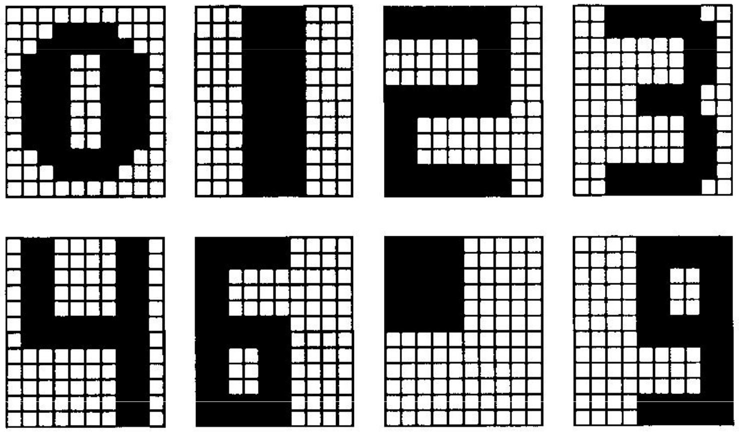
    (k aktualizaci potřebujeme znát hodnoty ostatních stavů, které by byli při výpočtu změněny)
- ☑ asynchronní
  - ➔ Systematicky či náhodně jsou vybírány neurony a nově vypočtená hodnota původní okamžitě nahradí $\Rightarrow$ ovlivní to (i pořadí) navazující výpočty a tím se i změní chování sítě.

HOPFIELDOVA SÍŤ
POUŽITÍ

- ☑ jako autoasociativní paměť;
- ☑ pro řešení optimalizačních problémů;

HOPFIELDOVA SÍŤ
PŘÍKLAD

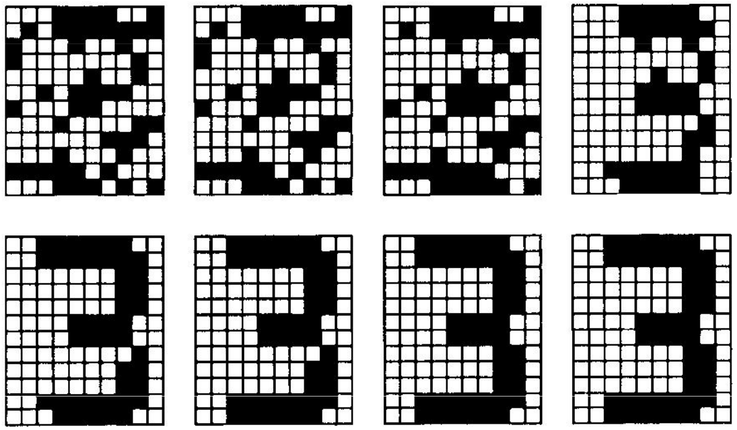
- ☑ Učení a rekonstrukce osmi obrazů s následujícími vzory



- ☑ 120 neuronů (obrázky jsou 10 x 12 = 120 bodů) a tedy 120^2 = 14 400 vah
- ☑ je vhodné aby vzory měly co největší Hammingovu vzdálenost

HOPFIELDOVA SÍŤ
PŘÍKLAD

- ☑ Fáze vybavování (rekonstrukce)



HOPFIELDOVA SÍŤ
VLASTNOSTI

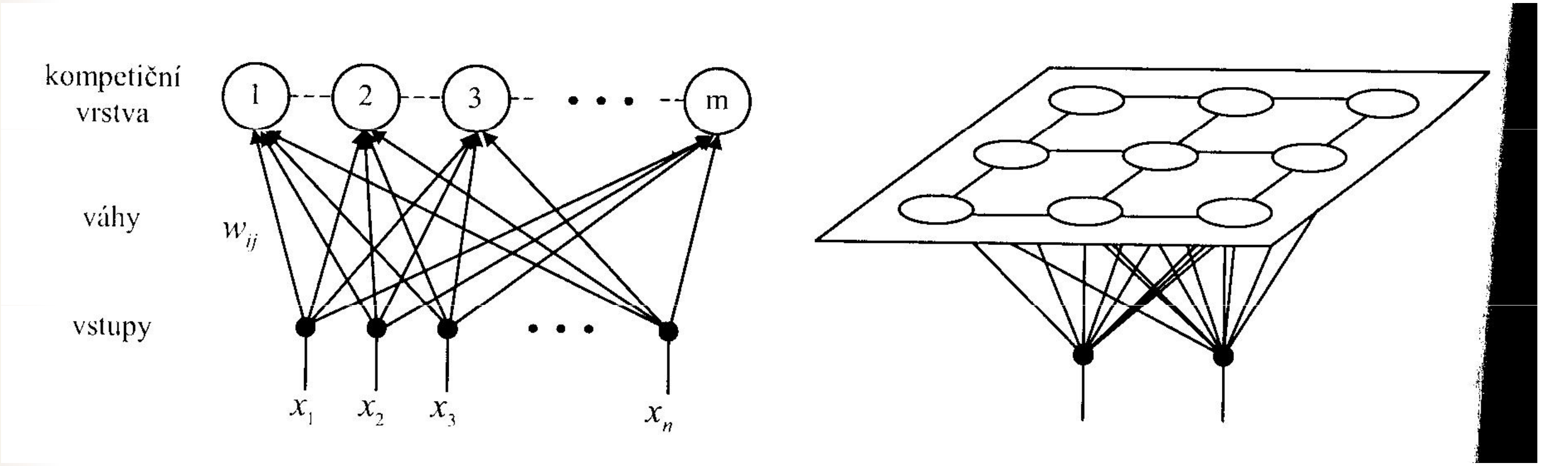
- ☑ omezení při použití jako asociativní paměť:
  - počet uložených obrazců (vzorů) (kapacita paměti) může být relativně malý – je určen poměrem počtu trénovacích vzorů a počtu neuronů (naučíme-li síť příliš mnoho vzorů, může síť konvergovat k nějakému zvláštnímu obrazci, kterému nebyla naučena); je třeba, aby bylo

$$\text{počet vzorů} \leq 0{,}138 . \text{počet neuronů}$$

  - nutnost volit co nejvzdálenější vzory, tj. maximalizovat Hammingovu vzdálenost

KOHONENOVA SÍŤ
(SAMOORGANIZUJÍCÍ SE MAPA – SOM)
STRUKTURA

- ☑ obsahuje jedinou vrstvu v tzv. Kohonenově (kompetiční) vrstvě;
- ☑ vstupy jsou plně propojeny s neurony



- ☑ neurony mají mezi sebou postranní vazby, které definují topologickou mřížku sítě – nejčastěji čtvercová
- ☑ váhy neuronů lze vnímat jako souřadnice neuronu v prostoru;

KOHONENOVA SÍŤ
STRUKTURA

- ☑ neurony vycházejí z formálních neuronů, které nemají práh – neurony radiálního typu;
- ☑ jejich výstup je zpravidla dvouhodnotový (aktivní, neaktivní);
- ☑ počet neuronů je volitelný (parametr sítě) – v praxi desítky až stovky;

KOHONENOVA SÍŤ
VYBAVOVÁNÍ

- ☑ nejdříve vypočítány vzdálenosti d_j mezi předloženým vzorem a vahami všech neuronů v kompetiční vrstvě, např. podle vztahu (Kohonen)

$$d_j = \sum_{i=1}^{m} (x_i - w_{ij})^2,$$

- ☑ kde index j prochází přes všechny neurony kompetiční vrstvy, kterých je m, x_i jsou elementy předloženého vzoru a w_{ij} jsou váhy neuronů. Vybere se ten neuron j* s minimální vzdáleností od předloženého vzoru

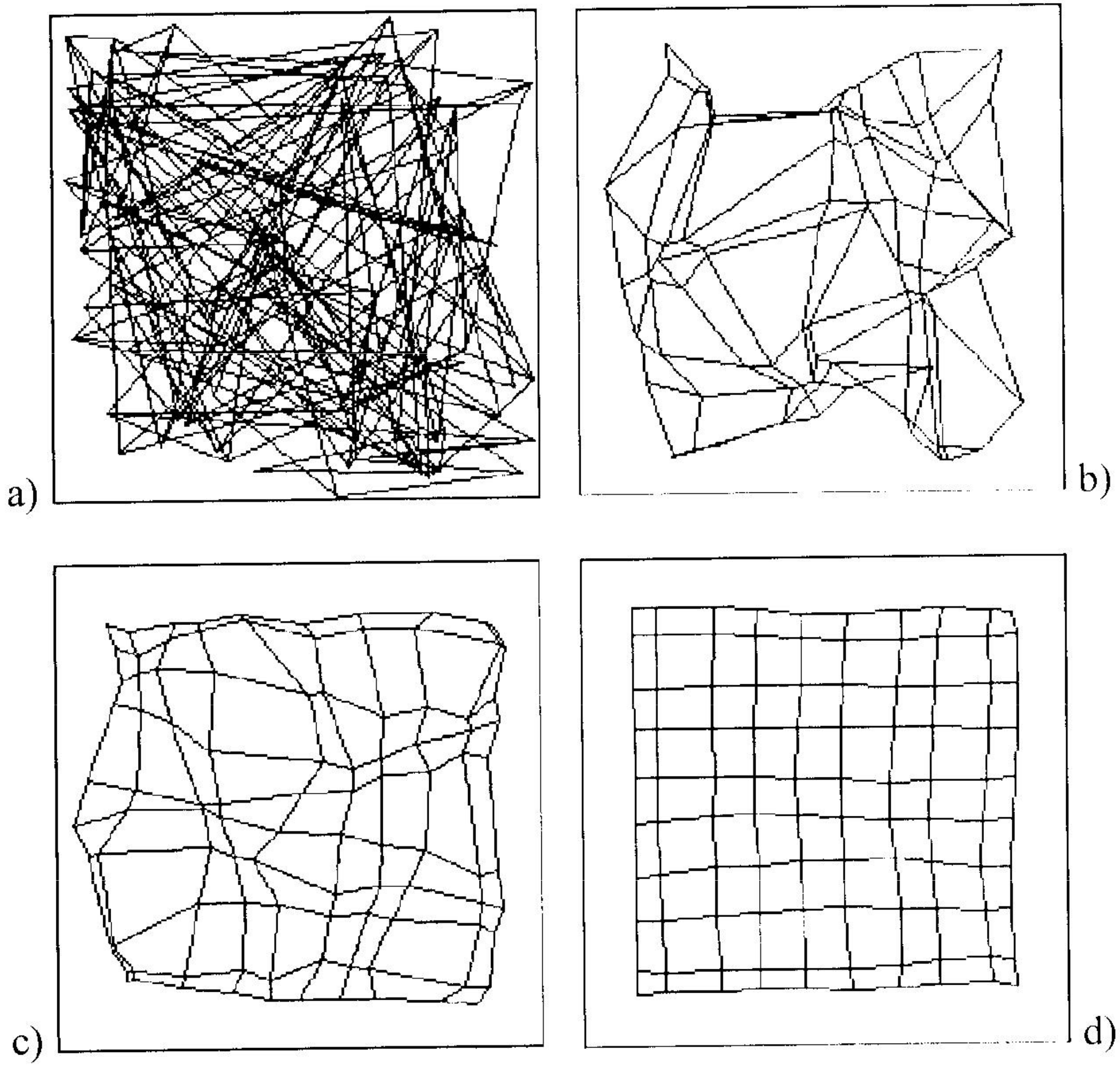
$$d_{j^*} = \min_j d_j$$

- ☑ výstup tohoto neuronu je aktivní, výstupy ostatních neuronů jsou neaktivní

KOHONENOVA SÍŤ
UČENÍ

- ☑ učení je založeno na porovnávání vstupních vzorů a váhových vektorů uložených v každém neuronu;
- ☑ učící algoritmus se snaží rozmístit neurony v mřížce tak, aby jejich rozdělení aproximovalo hustotu rozdělení trénovacích vzorů;
- ☑ jakmile je nalezen neuron nejbližší předloženému trénovacímu vzoru, jsou upraveny váhy tohoto neuronu a dále váhy neuronu v jeho okolí; protože je počáteční nastavení vah náhodné, jsou i neurony v prostoru umístěny náhodně a teprve vlivem učení se jejich rozmístění přibližuje rozdělení trénovacích vzorů

KOHONENOVA SÍŤ
UČENÍ



Obr. B.15. Počáteční stav, postupná adaptace mřížky a výsledný stav po 10 000 trénovacích krocích.

KOHONENOVA SÍŤ
UČENÍ

- adaptace vah podle vztahu

$$w_{ij}(t+1) = w_{ij}(t)+\eta(t)h(\nu,t)(x_i(t)-w_{ij}(t)),$$

$$i=1,\ldots,n,\ j=1,\ldots,m;$$

parametr učení η, $0<\eta<1$; nejdříve blízký 1, posléze se zmenšuje; kolem každého neuronu je definováno okolí, ve kterém jsou prováděny změny vah, pokud tento neuron bude vybrán v kompetici – velikost, tvar a míra vlivu tohoto okolí jsou parametry sítě a mění se během učení;

- délku (dobu) učení určuje počet kroků, nikoliv chyba, která se v KS obtížně stanoví

LOKÁLNÍ OKOLÍ NEURONU

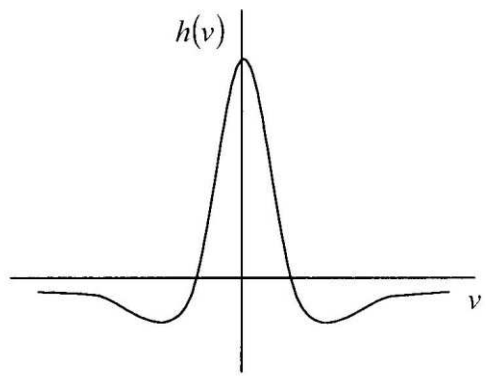
- ☑ slouží k tomu, aby neurony, které jsou v blízkosti vybraného neuronu, byly adaptovány podobným způsobem – cílem učení KS je kromě aproximace hustoty rozdělení vzorů v trénovací množině i rozmístění topologicky podobných neuronů do stejné oblasti prostoru;
- ☑ na počátku se velikost okolí volí velké a postupem výpočtu se monotónně zmenšuje;

LOKÁLNÍ OKOLÍ NEURONU

- $h(\nu,t)$ - adaptační funkce okolí
  - ostře ohraničená funkce
  - lineárně klesající
  - Gaussova funkce (aproximace vlivu neuronu na okolí v biologických sítích

$$h(\nu) = h_0.\exp(-\nu^2/\sigma^2),$$

kde ν je topologická vzdálenost v mřížce od středového neuronu



Obr. B.14. Biologická adaptační funkce.

KOHONENOVA SÍŤ
VLASTNOSTI

- základní varianta bez učitele; pro klasifikaci byla vytvořena modifikace s učitelem
- výhody:
  - odolnost vůči náhodnému šumu v trénovacích datech;
- nevýhody:
  - značný počet trénovacích iterací;
  - potřeba upravovat váhy v každém kroku výpočtu, včetně parametrů okolí vítězného neuronu;