Učení bez učitele

Už bylo: Učení bez učitele (unsupervised learning)

- Kompetitivní modely
- Klastrování
- Kohonenovy mapy
- LVQ (Učení vektorové kvantizace)

Zbývá: Hybridní modely (kombinace učení bez učitele a s učitelem

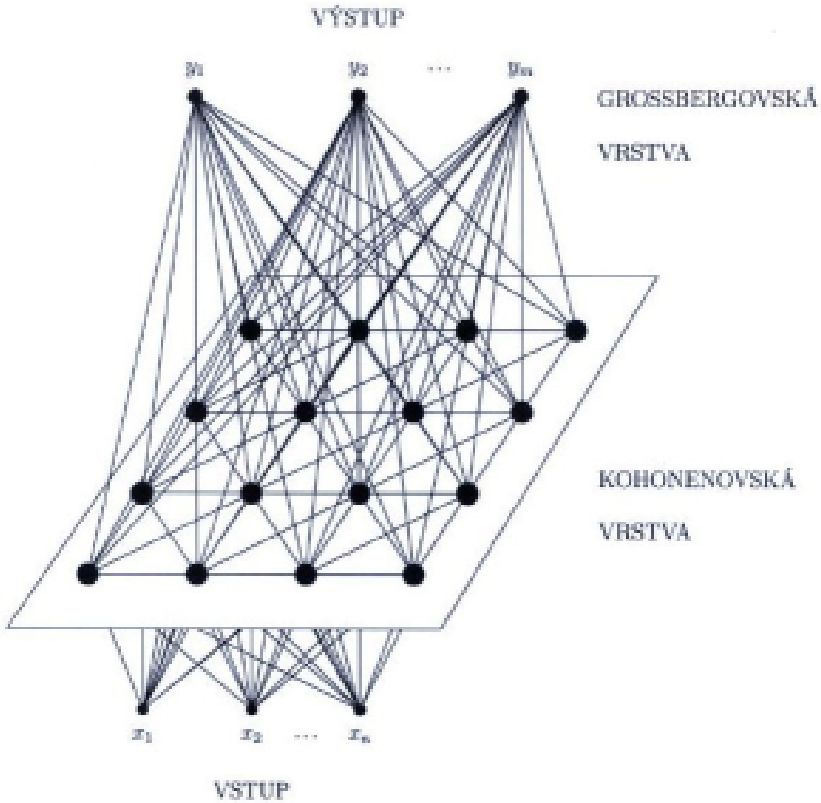
- Sítě se vstřícným šířením (Counterpropagation)
- RBF-sítě
- ART (Adaptive Resonance Theory)

Sítě se vstřícným šířením (Counter-propagation)

(Hecht-Nielsen, 1987)

Architektura

- Tři vrstvy neuronů:
  - Vstupní vrstva
  - Kohonenovská (klastrovací) vrstva
  - Grossbergovská vrstva
- Učení s učitelem
- Rozpoznávání

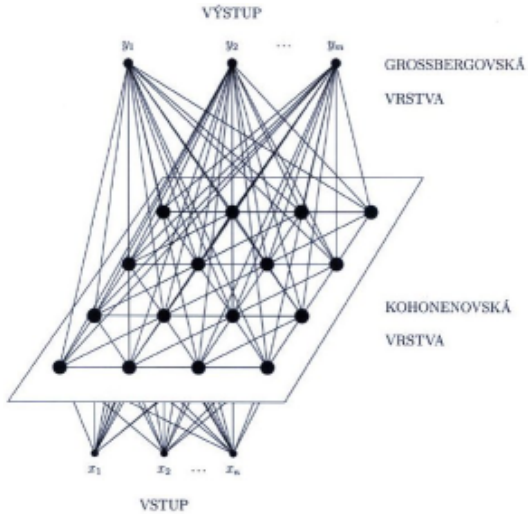


obrázek převzat z I.Mrázová, Neuronové sítě

Sítě se vstřícným šířením (Counter-propagation)

Terminologie

- Vstupní vrstva ... n neuronů
- Kohonenovská vrstva ... mřížka s K neurony
- Grossbergovská vrstva ... m neuronů
- Síť zobrazí vstupní vektor $\vec{x} \in R^n$ na výstupní vektor $\vec{y} \in R^m$



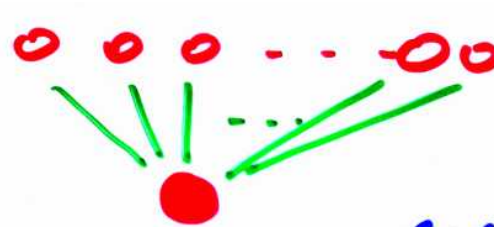
  - z_i ... výstupy (aktivity) neuronů v Kohonenovské vrstvě
  - y_l ... výstupy (aktivity) neuronů v Grossbergovské vrstvě
  - w_{ji} ... váhy hran mezi vstupní a Kohonenovskou vrstvou
  - v_{il} ... váhy hran mezi Kohonenovskou a Grossbergovskou vrstvou

Sítě se vstřícným šířením (Counter-propagation)

Režim vybavování

- zobrazení $f : R^n \rightarrow R^m$:
- Vstupní vektor $\vec{x}$ vybudí jeden neuron v Kohonenovské vrstvě (vítězný).. k-tý
- Grossbergovská (výstupní) vrstva:
  - Provádí standardní skalární součin:

$$y_l = \sum_{i=1}^{K} v_{il} z_i = v_{kl}$$



Grossbergova (výstupní) hvězda

$\rightarrow$ výstup sítě:

$$\vec{y} = \vec{v_k}$$

- Grossbergovská vrstva provádí výběr jednoho vektoru z K vektorů ($\sim$ váhy hran od k-tého neuronu v Kohonenově mřížce)

Sítě se vstřícným šířením (Counter-propagation)

Algoritmus

1. **Inicializace:** Zvolíme náhodné hodnoty synaptických vah
2. Předložíme nový trénovací vzor ve tvaru $(\vec{x}, \vec{t}) =$ (vstup, požadovaný výstup).
3. Spočítáme vzdálenosti d_i mezi $\vec{x}$ a $\vec{w}_i$ pro každý neuron i v Kohonenovské vrstvě. Použijeme např. Euklidovskou metriku:

$$d_i = \sqrt{\sum_j (x_j - w_{ji})^2}$$

4. Vyber neuron k s minimální vzdáleností d_k jako ,,vítěze"

$$k = argmin_i d_i$$

Sítě se vstřícným šířením (Counter-propagation)

Algoritmus - pokračování

5. Aktualizujeme váhy w_{ji} mezi vstupním neuronem j a neurony i Kohonenovské vrstvy, které se nacházejí v okolí vítězného neuronu k tak, aby lépe odpovídaly předloženému vzoru:

$$\vec{w}_i(t+1) = \vec{w}_i(t) = \alpha(t)\Lambda(i,k)(t)(\vec{x} - \vec{w}_i(t)),$$

- $\Lambda(i,k)$ ... funkce okolí
- $0 < \alpha(t) < 1$ ... parametr učení pro váhy mezi vstupní a Kohonenovskou vrstvou, klesá v čase.
- t představuje současný a (t + 1) následující krok učení.

Sítě se vstřícným šířením (Counter-propagation)

Algoritmus - dokončení

6. Aktualizujte váhy v_{kl} mezi ,,vítězným" neuronem k z Kohonenovské vrstvy a neurony l Grossbergovské vrstvy tak, aby výstupní vektor lépe odpovídal požadované odezvě:

$$v_{kl}(t+1) = (1-\gamma)v_{kl}(t) + \gamma z_k t_l,$$

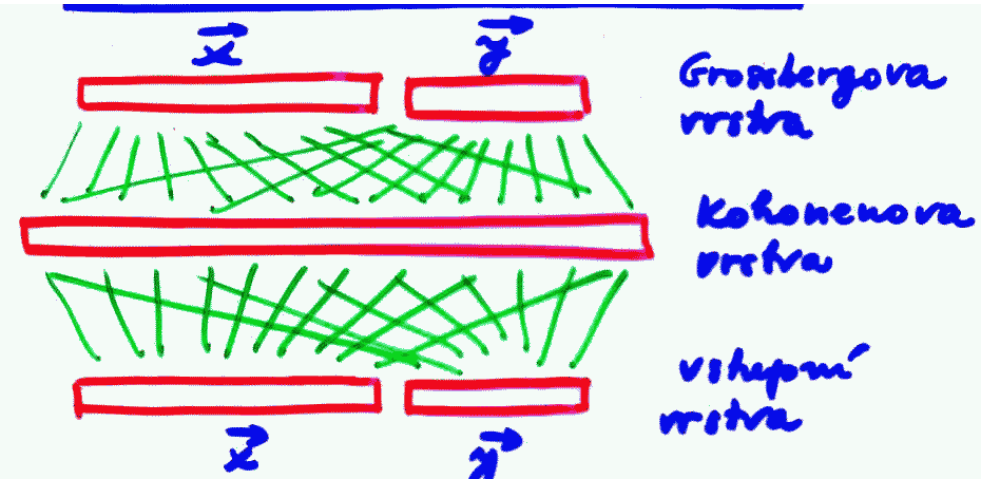
   - $0 < \gamma < 1$ ... parametr učení pro váhy mezi Kohonenovskou a Grossbergovskou vrstvou,
   - z_k ... označuje aktivitu ,,vítězného" neuronu Kohonenovské vrstvy.
   - t_l ... označuje požadovanou aktivitu neuronu l Grossbergovské vrstvy

7. Pokračujeme krokem (2)

Sítě se vstřícným šířením (Counter-propagation)

Příklady použití

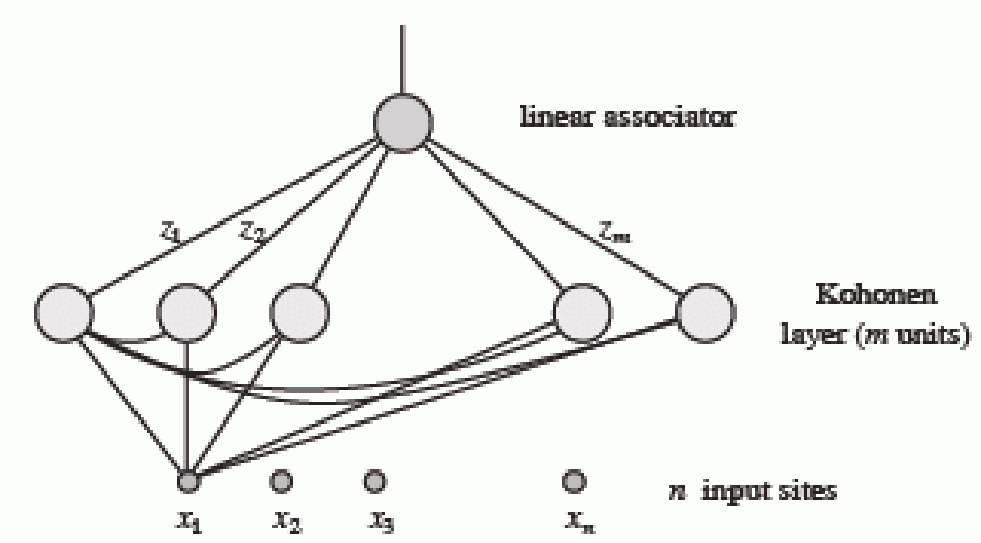
- Heteroasociativní paměť
- Komprese dat
  - např. přenos obrazů, videa
- Podobně jako BP-síť
  - efektivnější výpočet, rychlejší adaptace
  - nižší přesnost
- Původní využití: reprezentace zobrazení f a f^{-1} zároveň:



RBF-sítě (Sítě s lokálními jednotkami)

Radial basis functions
(Moody, Darken, 1989)

- Hybridní architektura
- Učení s učitelem
- Rozdíl od counter-propagation: Gaussovské jednotky v Kohonenovské vrstvě



RBF-sítě (Sítě s lokálními jednotkami)

Neurony v Kohonenovské vrstvě

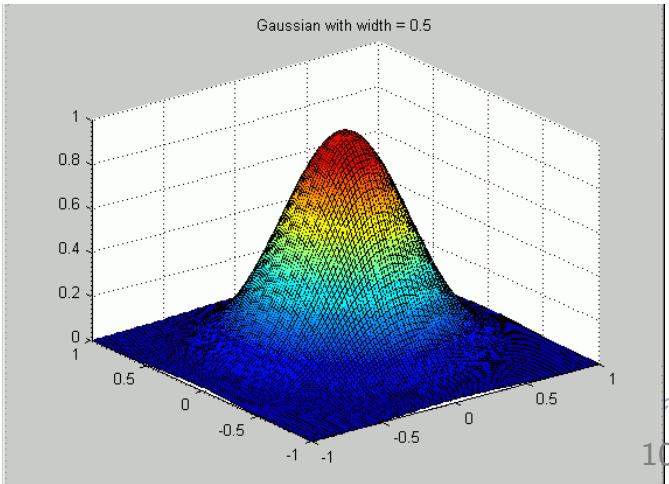
- Lokální výpočetní jednotky (RBF-jednotky)
- Neuron spočte svůj vnitřní potenciál ξ a výstup y podle:

$$\xi = \frac{\| \vec{x} - \vec{w} \|}{h}$$

- Gaussovská (radiální) přenosová funkce:

$$z = f(\xi) = e^{-\frac{\xi^2}{\alpha}} = e^{-\frac{\| \vec{x} - \vec{w} \|^2}{\alpha h^2}}$$

- $\vec{x} \in R^n$ ... vstupní vektor
- $\vec{w} \in R^n$ ... váhový vektor neuronu
- h ... konstanta (pro daný neuron)
  ... šířka okolí
- α ... konstanta



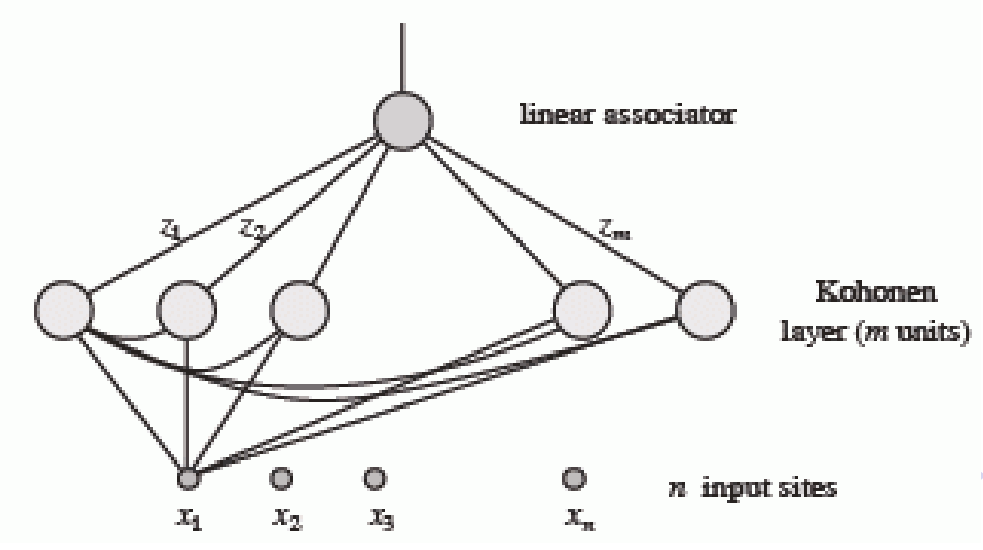
RBF-sítě (Sítě s lokálními jednotkami)

Celková funkce sítě

- $f : R^n \rightarrow R^m$:

$$f_l(x_1, ..., x_n) = \sum_{i=1}^{K} v_{il} z_i = \sum_{i=1}^{K} v_{il} e^{-\frac{\|\vec{x} - \vec{w}_i\|^2}{\alpha h_i^2}}$$

- $\vec{v}_l \in R^K$ ... váhový vektor ze skrytých neuronů do výstupního neuronu l
- Výstupní neurony jsou lineární jednotky



RBF-sítě (Sítě s lokálními jednotkami)

Algoritmus učení

- **Vstup:** trénovací množina s N vzory ve tvaru $(\vec{x}_p, \vec{d}_p) =$ (vstup, požadovaný výstup).
- **Výstup:** parametry sítě - váhy hran a parametry neuronů

Algoritmus učení má tři fáze:

1. Spočítáme středy centroidů (RBF-jednotek) ... váhy w_{ji} ze vstupní do Kohonenovské vrstvy
2. Spočítáme šířky okolí centroidů h_i a další parametry
3. Spočítáme váhy do výstupní vrstvy ... v_{il}

RBF-sítě (Sítě s lokálními jednotkami)

Algoritmus učení - má tři fáze

1. Spočítáme středy centroidů ... váhy w_{ji} ze vstupní do Kohonenovské vrstvy
   - samoorganizace (učení bez učitele)
   - viz. counter-propagation
2. Spočítáme šířky okolí centroidů h_i a další parametry
   - např. podle vzdálenosti nejbližších sousedů (není třeba znova předkládat trénovací vzory)
3. Spočítáme váhy do výstupní vrstvy ... v_{il}
   - např. pomocí algoritmu zpětného šíření (učení s učitelem)

RBF-sítě (Sítě s lokálními jednotkami)

Algoritmus učení - výpočet vah do výstupní vrstvy... v_{il}

- Pomocí algoritmu zpětného šíření (učení s učitelem)
- N trénovacích vzorů ve tvaru $(\vec{x}_p, \vec{d}_p)$ = (vstup, požadovaný výstup)
- Chybová funkce:

$$E = \frac{1}{2}\sum_{p=1}^{N}\sum_{l=1}^{m}(\sum_{i=1}^{K} v_{il}z_i - d_p)^2 = \frac{1}{2}\sum_{p=1}^{N}\sum_{l=1}^{m}(\sum_{i=1}^{K} v_{il}e^{-\frac{\|\vec{x}_p-\vec{w}_i\|^2}{\alpha h_i^2}} - d_p)^2$$

- Adaptační pravidlo pro jeden trénovací vzor

$$\Delta v_{il} \sim -\frac{\partial E}{\partial v_{il}} \sim \gamma e^{-\frac{\|\vec{x}_p-\vec{w}_i\|^2}{\alpha h_i^2}}(d_p - \sum_{i=1}^{K} v_{il}e^{-\frac{\|\vec{x}_p-\vec{w}_i\|^2}{\alpha h_i^2}})$$

$$= \gamma z_i(d_p - \sum_{i=1}^{K} v_{il}z_i)$$

RBF-sítě (Sítě s lokálními jednotkami)

Analýza modelu

- Univerzální aproximátor (narozdíl od BP-sítí stačí jedna skrytá vrstva)
  ... ale potřebný počet lokálních jednotek roste exponencielně
- Alternativa BP-sítí, pro některé typy problémů se hodí lépe, pro některé hůře než BP-sítě
- Rychlé učení (až o dva řády rychlejší než BP-sítě)
- Neumí si poradit s irelevantními vstupy.
- Obtížně se hledá učící algoritmus

RBF - Jak je to v Matlabu

- *newrbe* ... vytvoření modelu
  - počet výpočetních jednotek je roven počtu trénovacích vzorů
  - net = newrbe(P,T,SC)
  - P ... vstupní vzory
  - T ... výstupní vzory
  - SC ... šířka okolí
- *newrb* ... vytvoření modelu
  - přidává výpočetní jednotky, dokud MSE není menší než daná mez (EG)
  - net = newrb(P,T,EG,SC)
  - P ... vstupní vzory
  - T ... výstupní vzory
  - EG ... požadovaná MSE
  - SC ... šířka okolí
- *sim* ... rozpoznávání
  - Y = sim(net,P).

ART-sítě (Adaptive resonance theory)

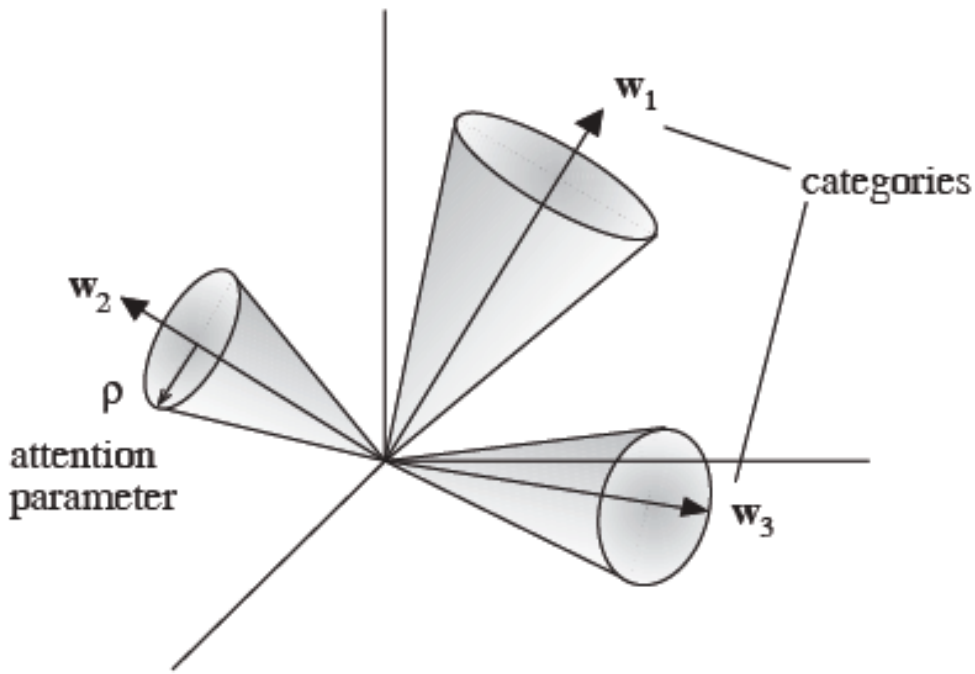
(Grossberg, Carpenter, 1986)

- Hybridní architektura - částečně modulární
- Učení bez učitele
- Online učení

Použití

- Shlukování - plasticita a stabilita
- Rozpoznávání znaků, řečových segmentů apod.

Úloha

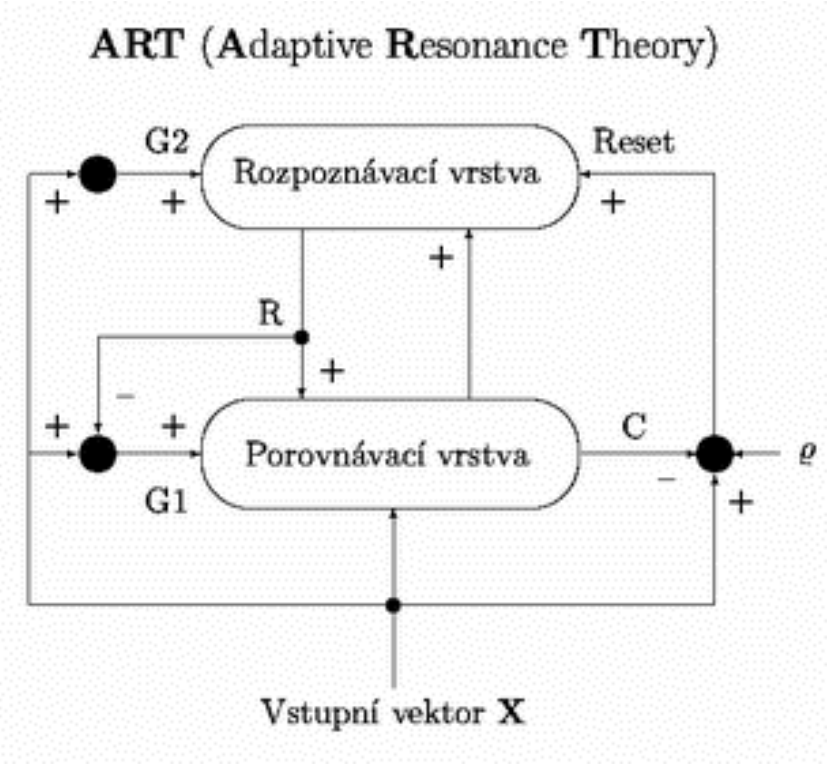


ρ ... parametr bdělosti

ART-sítě (Adaptive resonance theory)

Architektura ART-1

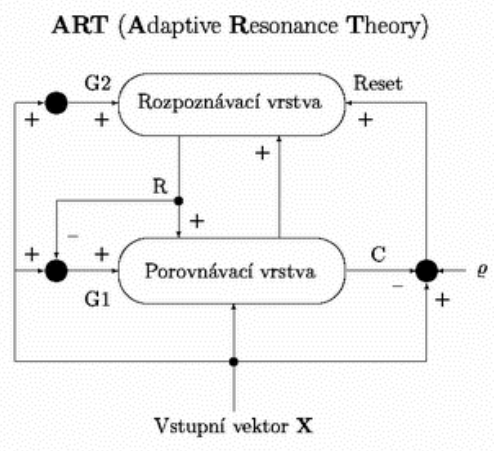
- Dvouvrstvá rekurentní síť
  - Porovnávací (vstupní) vrstva ... n neuronů
  - Rozpoznávací (výstupní) vrstva ... m neuronů
- ART-1 ... binární vstupy
- ART-2 ... reálné vstupy



ART-sítě (Adaptive resonance theory)

Vazby mezi neurony:

- ve výstupní vrstvě ... laterální inhibice
- ze vstupní do výstupní vrstvy
  (váhy $w_{ij}, i = 1, ..., n, j = 1, ..., m,$)
- z výstupních neuronů ke vstupním
  (váhy $t_{ij}, i = 1, ..., n, j = 1, ..., m,$)
  ... pro porovnání skutečné podobnosti s předloženým vzorem (založena na skalárním součinu)
- Řídící signály ... G1, G2, Reset



ART-sítě (Adaptive resonance theory)

Test bdělosti

- práh bdělosti ρ ... určuje, jak blízko musí být předložený vzor k uloženému, aby mohly patřit do stejné kategorie
- Mechanismus vypnutí (zablokování) neuronu s maximální odezvou
  $\rightarrow$ stabilita $\times$ plasticita sítě
  $\rightarrow$ síť má velké problémy i při jen trochu zašumněných vzorech (příliš narůstá počet uložených vzorů)

ART-sítě (Adaptive resonance theory)

Algoritmus učení – má 5 fází:

1. inicializační - nastavení počátečního stavu sítě
2. rozpoznávací - dopředný výpočet - naleznu vítězný neuron v rozpoznávací vrstvě
3. porovnávací - zpětný výpočet - provedu test bdělosti
4. vyhledávací - hledám jiný vítězný neuron
5. adaptační - adaptace vah u vítězného neuronu

ART-sítě (Adaptive resonance theory)

Algoritmus učení – inicializační fáze

1. **Počáteční inicializace vah:**

$$\begin{aligned} t_{ij}(0) &= 1, \\ w_{ij}(0) &= \frac{1}{1+n}, \\ i = 1,...,n \quad &, \quad j = 1,...,m, \\ 0 \leq \quad \rho \quad &\leq 1 \end{aligned}$$

- $w_{ij}(t)$ ... váha mezi vstupním neuronem i a výstupním neuronem j v čase t
- $t_{ij}(t)$ ... váha mezi výstupním neuronem j a vstupním neuronem i v čase t (vzor specifikovaný výstupním neuronem j)
- ρ ... práh bdělosti

ART-sítě (Adaptive resonance theory)

Algoritmus učení – inicializační a rozpoznávací fáze

2. Předlož nový vstupní vzor: $\vec{x}(t) = \{x_1, ..., x_n\}$
3. Spočti odezvu (aktivitu) neuronů ve výstupní (rozpoznávací) vrstvě:

$$y_j(t) = \sum_{i=1}^{n} w_{ij}(t) x_i,\ j = 1, ..., m$$

   - $y_j(t)$ ... aktivita výstupního neuronu j v čase t

4. Vyber neuron k, který nejlépe odpovídá předloženému vzoru (např. pomocí laterální interakce):

$$k = argmax\{y_j\}$$

ART-sítě (Adaptive resonance theory)

Algoritmus učení – porovnávací a vyhledávací fáze

5. **Test bdělosti:**
   - Výpočet bdělosti μ vítězného neuronu k podle:

$$\mu = \frac{\|T.\vec{x}\|}{\|\vec{x}\|},$$

$$\|T.\vec{x}\| = \sum_{i=1}^{n} t_{ik}(t)x_i, \ \|\vec{x}\| = \sum_{i=1}^{n} x_i,$$

   - Pokud platí $\mu > \rho$, pokračuj krokem 7, jinak pokračuj krokem 6.
6. **Zmraz (zablokuj) neuron k s největší odezvou:**
   - Nastav výstup neuronu k dočasně na nulu.
   - Opakuj krok 3 (neuron k se neúčastní maximalizace).

ART-sítě (Adaptive resonance theory)

Algoritmus učení – adaptační fáze

7. Pokud nebyl nalezen vhodný neuron, přidej do sítě nový neuron jako ,,vítězný".
8. **Adaptace vah u ,,vítězného" neuronu k:**

$$\begin{aligned} t_{ik}(t+1) &= t_{ik}(t)x_i, \\ w_{ik}(t+1) &= \frac{t_{ik}(t)x_i}{0.5 + \sum_{l=1}^{n} t_{lk}(t)x_l} \end{aligned}$$

9. Odblokuj všechny zmražené neurony a opakuj krok 2.

ART-sítě (Adaptive resonance theory)

Analýza modelu

- Hlavní výhody: Stabilita a plasticita sítě
- Síť sama určí správný počet neuronů
- Velká citlivost na počáteční volbu parametrů:
  - práh bdělosti
  - pořadí předkládání vzorů
- Velká citlivost na šum v datech

Příklady aplikací

- Shlukování
- Rozpoznávání znaků, řečových segmentů apod.

Kaskádová korelace

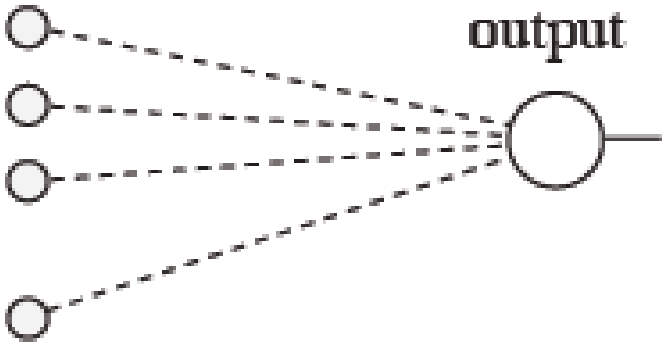
(Fahlman, Labiere, 1990)

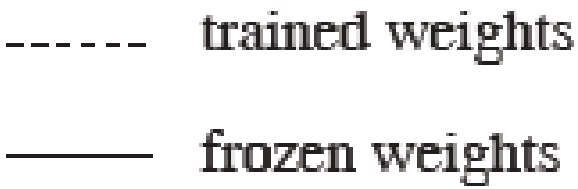
- robustní rostoucí architektura BP-sítě

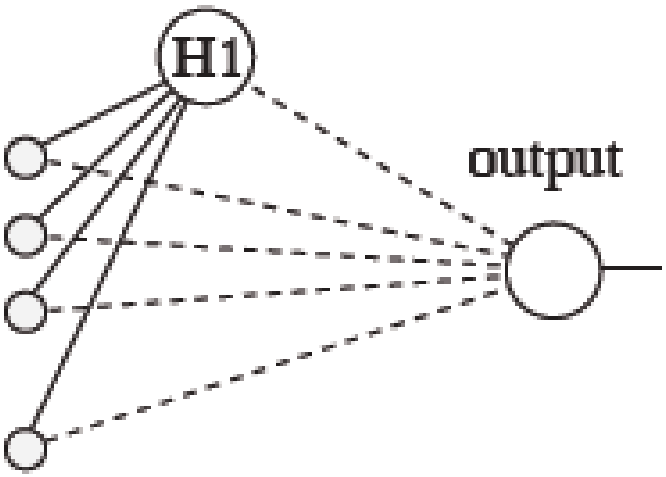
Princip

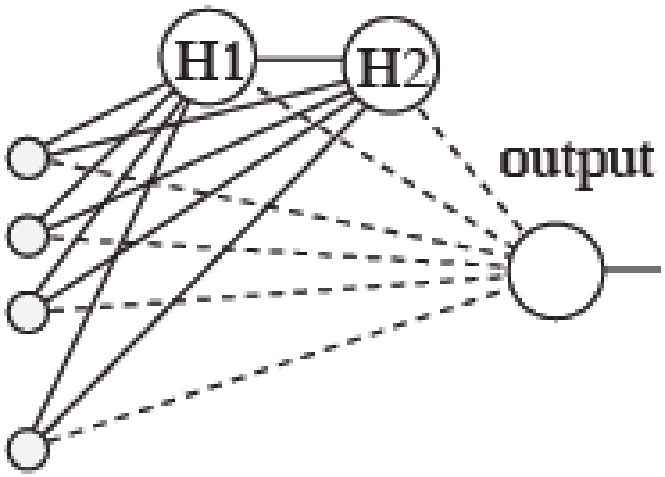
- Systém začíná proces učení s přímým propojením vstupů na výstup
- Postupně jsou přidávány další skryté neurony
- Vstupy každého nového neuronu jsou propojeny se všemi původními vstupy i se všemi dříve vytvořenými neurony

Kaskádová korelace









Kaskádová korelace

Algoritmus učení

- Minimalizace MSE na výstupu sítě

Učení probíhá ve dvou fázích:

- **První fáze:** Adaptace sítě pomocí algoritmu Quickprop
  - pokud je MSE na výstupu dostatečně nízká, KONEC
  - jinak přidáme nový neuron
- **Druhá fáze:** Přidání nového neuronu
  - nový neuron je adaptován tak, aby maximalizoval korelaci mezi svým výstupem a chybou na výstupu sítě
    $\rightarrow$ přidávaný neuron se ,,naučí" nějaký příznak, který vysoce koreluje s aktuální (zbývající) chybou
  - Váhy do nově přidaného neuronu jsou zmrazeny a v dalších fázích se doučují jen váhy na výstup

Kaskádová korelace

Algoritmus učení

- Cílem učení skrytých neuronů je maximalizace S:

$$S = |\sum_{i=1}^{p}(V_i - \overline{V})(E_i - \overline{E})|$$

  - p ... počet trénovacích vzorů
  - V_i ... výstup přidávaného neuronu pro i-tý vzor
  - $\overline{V}$ ... průměrný výstup přidávaného neuronu
  - E_i ... MSE pro i-tý vzor
  - $\overline{E}$ ... průměrná chyba

Kaskádová korelace

Algoritmus učení

- Cílem učení skrytých neuronů je maximalizace S:

$$S = |\sum_{i=1}^{p}(V_i - \overline{V})(E_i - \overline{E})|$$

$$\frac{\partial S}{\partial w_k} = \sum_{i=1}^{p} \sigma(E_i - \overline{E}) f_i' I_{ik}$$

  - σ ... znaménko korelace mezi výstupem a přidávaným neuronem
  - f_i' ... derivace přenosové funkce pro i-tý vzor
  - I_{ik} ... k-tý vstup přidávanho neuronu pro i-tý vzor

Kaskádová korelace

Analýza algoritmu

- Snadné rozšíření na více výstupů
- Síť sama určí správný počet neuronů ... uživatel ho nemusí specifikovat
- Rychlé učení ... v každém kroku se adaptuje jen jeden neuron, váhy do stávajících neuronů už se neadaptují $\rightarrow$ stabilita
- Nebezpečí přeučení ... saturace neuronů
- Vytvářejí se zbytečně hluboké sítě