Vysokoškolská skripta

VŠE

FAKULTA
INFORMATIKY
A STATISTIKY

UMĚLÉ NEURONOVÉ SÍTĚ

Ing. Miloš Křivan

Název	**Umělé neuronové sítě**
Autor	Ing. Miloš Křivan
Vydavatel	Vysoká škola ekonomická v Praze Nakladatelství Oeconomica
Doporučeno	pro magisterské studium na VŠE v Praze
Vydání	první v elektronické podobě
Návrh obálky	Daniel Hamerník, DiS.
Počet stran	77
DTP	Vysoká škola ekonomická v Praze Nakladatelství Oeconomica
Sazba	autor

Zdarma ke stažení

Tato publikace neprošla redakční úpravou.

ISBN 978-80-245-2420-7

Obsah

1. Definice umělé neuronové sítě

Umělou neuronovou síť (ANN) definujeme jako *orientovaný graf s dynamicky ohodnocenými vrcholy a hranami*, tj. jako uspořádanou pětici $[V, E, \varepsilon, y, w]$, kde je:

V	množina vrcholů (neuronů)
E	množina hran (synapsí)
ε	zobrazení incidence hran s vrcholy ($\varepsilon: E \to V \times V$)
y	dynamické ohodnocení vrcholů ($y: V \times \boldsymbol{t} \to \mathbb{R}$)
w	dynamické ohodnocení hran ($w: \varepsilon(E) \times \boldsymbol{T} \to \mathbb{R}$),

a pro $\forall t \in \boldsymbol{t}$ resp. $\forall T \in \boldsymbol{T}$ položíme $y([i,t]) \equiv y_i(t)$ resp. $w([i,j,T]) \equiv w_{ij}(T)$.

Vektor $\vec{y}(t) = [y_i(t) | i \in V]$ nazveme *stavem sítě* v čase t a analogicky vektor $\vec{w}(T) = [w_{ij}(T) | [i,j] \in V \times V]$ nazveme *konfigurací sítě* v čase T, přičemž pro $[i,j] \notin \varepsilon(E)$ položíme $w_{ij}(T) = 0$. Stav resp. konfiguraci sítě jakožto vektorové funkce času t resp. T nazveme *aktivní dynamikou* resp. *adaptivní dynamikou* umělé neuronové sítě.

Rozlišením časů aktivní a adaptivní dynamiky jsme vyjádřili skutečnost, že neuronová síť pracuje ve dvou časově nezávislých režimech, aktivním a adaptivním. V adaptivním režimu probíhá učení sítě, tj. nastavení konfigurace sítě jako důsledek síti předkládaných vzorů a jim odpovídajících obrazů, a v aktivním režimu probíhá realizace v adaptivním režimu naučené funkce sítě, tj. zaujetí stavu sítě jako důsledek na síť přiloženého stimulu.

Aktivní resp. adaptivní dynamiku umělé neuronové sítě spojitého času určíme jako vektory řešení následujících soustav diferenciálních rovnic (ART):

$$\frac{d}{dt} x_j(t) + x_j(t) = \sum_i f_i(x_i(t - \Delta t))\, w_{ij} - \vartheta_j$$

$$\frac{d}{dT} w_{ij}(T) + \beta\, g_j(x_j(T))\, w_{ij}(T) = \alpha\, f_i(x_i(T))\, g_j(x_j(T))$$

kde pro $i, j \in V$, $\alpha, \beta \in <0,1>$ a $\Delta t > 0$ vyjadřující zpoždění signálu je:

x_i	potenciál i-tého neuronu
f_i	aktivační funkce i-tého neuronu ($y_i = f_i(x_i)$)
g_j	adaptační funkce j-tého neuronu
ϑ_j	práh j-tého neuronu
w_{ij}	synaptická váha vazby i-tého neuronu s j-tým neuronem
α	míra plasticity synapse
β	míra elasticity synapse.

Z definice aktivní resp. adaptivní dynamiky plyne pro $t \to \infty$ resp. $T \to \infty$:

$$x_j(\infty) = \sum_i y_i(\infty)\, w_{ij} - \vartheta_j$$

resp. pro $\beta \neq 0$:

$$w_{ij}(\infty) = \frac{\alpha}{\beta}\, y_i(\infty)$$

a pro $\beta = 0$, $w_{ij}(0) = 0$, $g_j(x_j) = x_j$ za předpokladu zanedbání změny potenciálů neuronů během adaptivní dynamiky:

$$w_{ij}(T) = \alpha\, y_i(T)\, x_j(T)\, T$$

tj. z $w_{ij}(\infty) = \infty$ plyne, že pro $\beta = 0$ nemá smysl nechat čas adaptivní dynamiky běžet do nekonečna, ale necháme-li čas adaptivní dynamiky běžet *N*-krát pouze po dobu jednotkového časového skoku, pak pro $k \in \{1, -, N\}$ dostaneme (*Hebbovské učení*):

$$w_{ij}(N) = \alpha \sum_k y_i(k)\, x_j(k)$$

Elasticita synapse částečně eliminuje nevratnou deformaci synapse způsobenou procházejícím signálem a omezuje tak její schopnost adaptovat se na procházející signál, kterou představuje plasticita synapse. V případě nenulové resp. nulové míry elasticity synapse je kapacita paměti synapse omezená resp. neomezená.

Nahradíme-li v definici aktivní resp. adaptivní dynamiky uvedené derivace analogickými výrazy pro diskrétní čas:

$$\frac{d}{dt}x_j(t) \equiv \frac{x_j(t+1) - x_j(t)}{t + 1 - t} \qquad \frac{d}{dt}w_{ij}(T) \equiv \frac{w_{ij}(T+1) - w_{ij}(T)}{T + 1 - T}$$

a položíme-li $\Delta t = 0$, pak obdržíme následující soustavy diferenčních rovnic a vektory jejich řešení pak určují aktivní a adaptivní dynamiku umělé neuronové sítě diskrétního času:

$$x_j(t+1) = \textstyle\sum_i f_i(x_i(t))\, w_{ij} - \vartheta_j \quad \text{resp.} \quad y_j(t+1) = f_j(\textstyle\sum_i y_i(t)\, w_{ij} - \vartheta_j)$$

$$w_{ij}(T+1) = (1 - \beta\, g_j(x_j(T)))\, w_{ij}(T) + \alpha\, f_i(x_i(T))\, g_j(x_j(T))$$

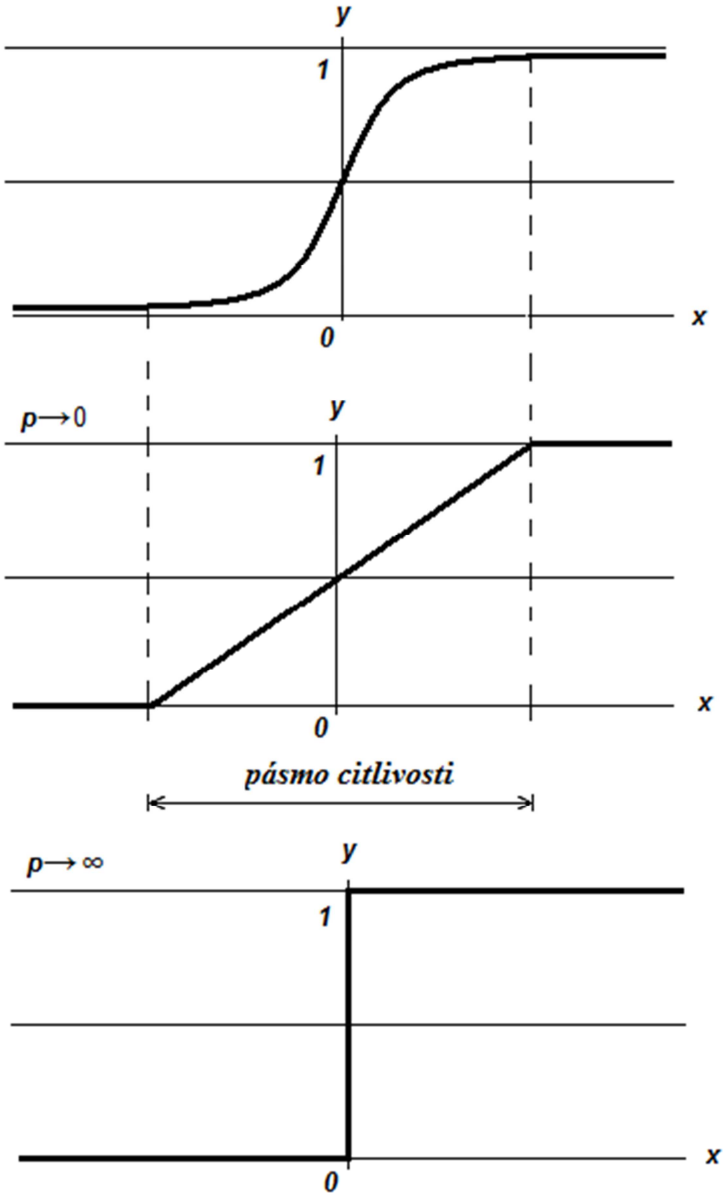
kde $i, j \in V$.

Závislost stavu na potenciálu neuronu (aktivační funkci) aproximujme *sigmoidou*:

$$f(x) = \frac{1}{1 + e^{-px}}$$

kde parametr $p > 0$ vyjadřuje strmost sigmoidy, pro strmost blížící se nule resp. nekonečnu dostaneme aktivační funkci ve tvaru *linearity* resp. *ostré nelinearity*:

$$\lim_{p \to 0} f(x) = \frac{1}{2} \qquad \lim_{p \to \infty} f(x) = 0 \quad x < 0 \qquad \lim_{p \to \infty} f(x) = 1 \quad x > 0$$



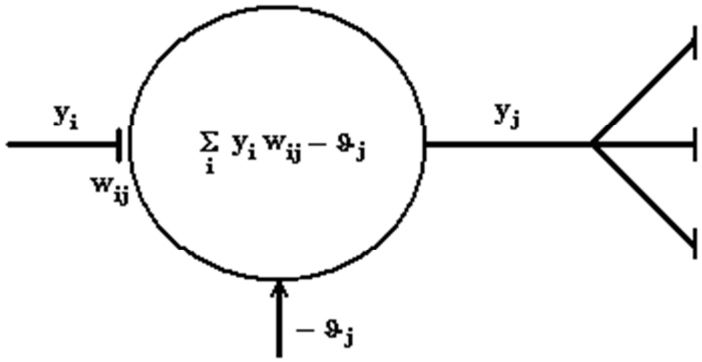
V následující ekvivalentní formulaci potenciálu a aktivační funkce neuronu:

$$x_j = \sum_i y_i w_{ij} \qquad f(x) = \frac{1}{1 + e^{-p(x-\vartheta)}}$$

chápeme parametr ϑ jako práh neuronu, jehož podkročení resp. překročení potenciálem neuronu neuron *inhibuje* resp. *excituje*. V původně uvedené formulaci potenciálu a aktivační funkce neuronu:

$$x_j = \sum_i y_i w_{ij} - \vartheta_j \qquad f(x) = \frac{1}{1 + e^{-px}}$$

chápeme parametry $-\vartheta_j$ jako vnější stimul aktivující síť svou injektáží přímo do neuronů, na jejichž hodnoty se iniciují potenciály neuronů: $x_j(0) = -\vartheta_j$, pak vektor $\vec{x}(0) = [x_j(0)|j \in V]$ označíme jako *vstup sítě*, jako *výstup sítě* chápeme stav sítě.



Vektorovou funkci přiřazující vstupu sítě výstup sítě nazveme *funkcí sítě*:

$$\vec{F}(\vec{x}(0)) = \vec{y}(\Delta t)$$

kde Δt je *doba aktivace* (odezvy) sítě, přičemž konfigurace sítě je parametrem funkce sítě.

Množinu vstupů sítě injektovaných do sítě během adaptivní dynamiky nazveme *trénovací množinou*:

$$\{\vec{x}(T)|T \in \Delta T\}$$

kde ΔT je *doba adaptace* (učení) sítě.

2. Lineární heteroasociativní paměť

Topologie sítě:

Rozdělme populaci neuronů sítě V na dvě disjunktní populace V_1 a V_2 ($V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$, $|V_1| = n$, $|V_2| = m$) a propojme je hranami tak, aby z každého neuronu populace V_1 směřovala hrana do každého neuronu populace V_2 ($\varepsilon(E) = V_1 \times V_2$), síť je tedy orientována směrem od populace V_1 k populaci V_2 a populaci V_1 resp. V_2 pak chápeme jako populaci vstupní resp. výstupní.

Aktivní dynamika:

Definujme aktivační funkci neuronů sítě jako identitu. Nastavíme-li počáteční hodnoty potenciálů neuronů vstupní populace následovně:

$$x_i(0) = v_i$$

pak v následujícím okamžiku dostaneme tyto hodnoty potenciálů neuronů výstupní populace:

$$x_j(1) = \sum_i v_i w_{ij}$$

kde $i \in V_1$ a $j \in V_2$, vektor $\vec{v} = [v_i | i \in V_1]$ pak chápeme jako vstup sítě.

Adaptivní dynamika:

Definujme adaptační funkci neuronů výstupní populace jako identitu. Mějme trénovací množinu ve tvaru $\{[\vec{a}(T), \vec{b}(T)] | T \in \Delta T\}$, kde $\vec{a} = [a_i | i \in V_1]$ a $\vec{b} = [b_j | j \in V_2]$. Položíme-li plasticitu resp. elasticitu rovnu jedné resp. nule, pak obdržíme následující adaptivní dynamiku:

$$w_{ij}(T) = w_{ij}(T-1) + a_i(T)\, b_j(T)$$

Položíme-li dále $\vec{w}(0) = \vec{0}$ a $\Delta T = \{1, \cdots, N\}$, pak po naučení sítě budou váhy nastaveny na následující hodnoty:

$$w_{ij}(N) = \sum_T a_i(T)\, b_j(T)$$

kde $i \in V_1$, $j \in V_2$ a $T \in \Delta T$.

Funkce sítě:

Nechť vektory $\vec{a}$ z trénovací množiny tvoří ortonormální bázi nadroviny ρ dimenze N v prostoru $\mathbb{R}^n$ ($N < n$), pak pro libovolný vektor $\vec{v} \in \rho$ platí:

$$x_j(1) = \sum_i v_i \sum_T a_i(T)\, b_j(T) = \sum_T b_j(T) \sum_i v_i a_i(T) = \sum_T b_j(T)\, \vec{v}\, \vec{a}(T) =$$

$$= \sum_T b_j(T)\, (s_1 \vec{a}(1) + \cdots + s_N \vec{a}(N))\, \vec{a}(T) = \sum_T s_T b_j(T)$$

kde $i \in V_1$, $j \in V_2$, $T \in \Delta T$ a $s_1, \cdots, s_N$ jsou souřadnice vektoru $\vec{v}$ vzhledem k výše uvedené ortonormální bázi nadroviny ρ, pak při době aktivace sítě rovné jedné dostaneme následující funkci sítě:

$$\vec{F}(\vec{v}) = \vec{F}(s_1 \vec{a}(1) + \cdots + s_N \vec{a}(N)) = s_1 \vec{b}(1) + \cdots + s_N \vec{b}(N)$$

Funkce sítě tedy libovolnému vstupu sítě ležícímu v nadrovině ρ, tvořenému lineární kombinací vektorů $\vec{a}$ (vzorů) trénovací množiny, přiřadí identickou lineární kombinaci jim odpovídajících heteroasociací, tj. vektorů $\vec{b}$ (obrazů) trénovací množiny.

3. Lineární autoasociativní paměť (LAM)

Topologie sítě:

Mějme populaci neuronů sítě V a propojme ji hranami tak, aby z každého neuronu populace směřovala hrana do každého neuronu populace ($\varepsilon(E) = V \times V$, $|V| = n$).

Aktivní dynamika:

Definujme aktivační funkci neuronů sítě jako identitu. Nastavíme-li počáteční hodnoty potenciálů neuronů následovně:

$$x_i(0) = v_i$$

pak v následujícím okamžiku dostaneme tyto hodnoty potenciálů neuronů populace:

$$x_j(1) = \sum_i v_i w_{ij}$$

kde $i, j \in V$, vektor $\vec{v} = [v_i | i \in V]$ pak chápeme jako vstup sítě.

Adaptivní dynamika:

Definujme adaptační funkci neuronů sítě jako identitu. Mějme trénovací množinu ve tvaru $\{\vec{c}(T) | T \in \Delta T\}$. Položíme-li plasticitu resp. elasticitu rovnu jedné resp. nule, pak obdržíme následující adaptivní dynamiku:

$$w_{ij}(T) = w_{ij}(T-1) + c_i(T)\, c_j(T)$$

Položíme-li dále $\vec{w}(0) = \vec{0}$ a $\Delta T = \{1, \cdots, N\}$, pak po naučení sítě budou váhy nastaveny na následující hodnoty:

$$w_{ij}(N) = \sum_T c_i(T)\, c_j(T)$$

kde $i, j \in V$ a $T \in \Delta T$.

Funkce sítě:

Nechť $\{\vec{a}(1),\cdots,\vec{a}(n)\}$, $\{\vec{b}(1),\cdots,\vec{b}(n)\}$ a $\{\vec{c}(1),\cdots,\vec{c}(n)\}$ jsou báze prostoru $\mathbb{R}^n$, kde druhou z uvedených bází získáme ortogonalizací báze první a třetí z uvedených bází získáme normalizací báze druhé, trénovací množina $\{\vec{c}(T)|T\in\Delta T\}$ je pak za předpokladu $N<n$ částí výše uvedené ortonormální báze, a pak platí (viz ortonormalizační proces):

$$\vec{F}\big(\vec{a}(T)\big)=\big(\vec{c}(1)\cdot\vec{a}(T)\big)\,\vec{c}(1)+\cdots+\big(\vec{c}(T-1)\cdot\vec{a}(T)\big)\,\vec{c}(T-1)\;+\big|\vec{b}(T)\big|\vec{c}(T)=\vec{a}(T)$$
$$\vec{F}\big(\vec{a}(S)\big)=\big(\vec{c}(1)\cdot\vec{a}(S)\big)\,\vec{c}(1)+\cdots+\big(\vec{c}(S-1)\cdot\vec{a}(S)\big)\,\vec{c}(S-1)+\big|\vec{b}(S)\big|\vec{c}(S)-$$
$$-\big(\vec{c}(N+1)\cdot\vec{a}(S)\big)\,\vec{c}(N+1)-\cdots-\big(\vec{c}(S-1)\cdot\vec{a}(S)\big)\,\vec{c}(S-1)-\big|\vec{b}(S)\big|\vec{c}(S)=\vec{a}(S)-\vec{d}(S)$$

kde $T\in\{1,\cdots,N\}$, $S\in\{N+1,\cdots,n\}$, z čehož pro libovolný vektor $\vec{v}\in\mathbb{R}^n$ plyne:

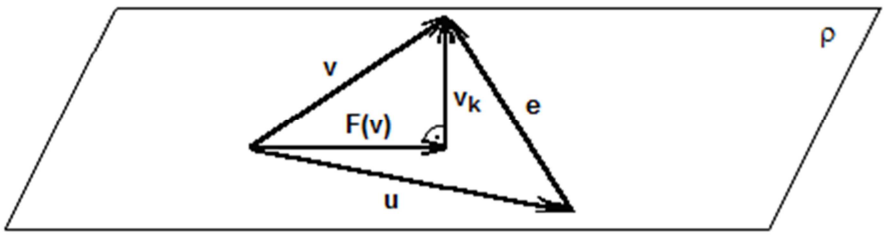
$$\vec{F}(\vec{v})=\vec{F}\big(s_1\vec{a}(1)+\cdots+s_n\vec{a}(n)\big)=s_1\vec{F}\big(\vec{a}(1)\big)+\cdots+s_n\vec{F}\big(\vec{a}(n)\big)=$$
$$=\big(s_1\vec{a}(1)+\cdots+s_n\vec{a}(n)\big)-\big(s_{N+1}\vec{d}(N+1)+\cdots+s_n\vec{d}(n)\big)=\vec{v}-\vec{v}_k=\vec{v}_p$$

kde $s_1,\cdots,s_n$ jsou souřadnice vektoru $\vec{v}$ vzhledem k bázi $\{\vec{a}(1),\cdots,\vec{a}(n)\}$ a $\vec{v}_p$ resp. $\vec{v}_k$ je ortogonální projekce vektoru $\vec{v}$ resp. kolmice spuštěná z vektoru $\vec{v}$ do nadroviny ρ generované vektory množiny $\{\vec{a}(T)|T\in\Delta T\}$.

Funkce sítě tedy libovolnému vstupu sítě při době aktivace sítě rovné jedné přiřadí jeho ortogonální projekci do nadroviny ρ, tj. jemu nejbližší autoasociaci z nadroviny ρ.

Nechť vektor $\vec{u}$ leží v nadrovině ρ a nechť vektor $\vec{v}$ je jeho náhodné zkreslení, pak si vektor $\vec{e}=\vec{v}-\vec{u}$ definujeme jako *šum* zkreslující vektor $\vec{u}$. Funkci sítě lineární autoasociativní paměti pak můžeme chápat jako filtraci šumu, tj. opravu vektoru $\vec{v}$.

Při konstantní velikosti šumu kvalita jeho filtrace závisí také na jeho poloze. Nejpříznivější poloha šumu je, je-li kolmý na nadrovinu ρ, pak oprava vektoru $\vec{v}$ se shoduje s vektorem $\vec{u}$ ($\vec{F}(\vec{v})=\vec{u}$). Nejnepříznivější poloha šumu je, leží-li v nadrovině ρ, pak oprava vektoru $\vec{v}$ se shoduje se zkreslením vektoru $\vec{u}$ ($\vec{F}(\vec{v})=\vec{v}$).



4. Lineární model (RAM)

Topologie sítě: Shodná s lineární autoasociativní pamětí.

Aktivní dynamika:

Definujme aktivační funkci neuronů sítě jako identitu. Nastavme počáteční hodnoty potenciálů neuronů následovně:

$$x_i(0) = v_i$$

a definujme vektorovou funkci:

$$\vec{P}(\vec{x}) = [v_1, \cdots, v_k, x_{k+1}, \cdots, x_n]$$

pro $\vec{x} \in \mathbb{R}^n$, tj. $k < n$ a mějme následující aktivní dynamiku:

$$x_j(t+1) = P_j(\sum_i x_i(t)\, w_{ij})$$

kde $i, j \in V$, vektor $\vec{v} = [v_i | i \in V]$ pak chápeme jako vstup sítě.

Adaptivní dynamika: Shodná s lineární autoasociativní pamětí.

Funkce sítě:

Definujme vektory $\vec{b}(0) = [v_1, \cdots, v_k, 0, \cdots, 0]$ a $\vec{b}(T) = [a_1(T), \cdots, a_k(T), 0, \cdots, 0]$ kde $\vec{b} \in \mathbb{R}^n$ a vektorovou funkci $\vec{Q}(\vec{x}) = [0, \cdots, 0, x_{k+1}, \cdots, x_n]$ pro $\vec{x} \in \mathbb{R}^n$, pak množina $\sigma = \{\vec{P}(\vec{x}) | \vec{x} \in \mathbb{R}^n\}$ je lineární posunutí nadroviny $\varsigma = \{\vec{Q}(\vec{x}) | \vec{x} \in \mathbb{R}^n\}$ o vektor $\vec{b}(0)$.

Předpokládejme, že vektory $\vec{b}(1), \cdots, \vec{b}(N)$ jsou lineárně nezávislé ($N \leq k$), pak existuje právě jeden vektor $\vec{y} \in \mathbb{R}^n$ ležící v průniku nadroviny ρ s množinou σ.

Pro $N > k$ uvažujme nadrovinu ρ_2 jako průnik nadrovin ρ a ς a nadrovinu ρ_1 jako množinu všech vektorů nadroviny ρ kolmých na každý vektor nadroviny ρ_2, pak nadrovina ρ je direktním součtem nadrovin ρ_1 a ρ_2 ($\rho = \rho_1 \oplus \rho_2$).

1.tvrzení: Pro každý vektor $\vec{x}$ nadroviny ρ_1 platí, že vektor $\vec{F}(\vec{P}(\vec{x}))$ leží v nadrovině ρ_1.

Důkaz: $\vec{z} \in \rho_2$

$$\left(\vec{x} - \vec{F}(\vec{P}(\vec{x}))\right)\vec{z} = \left(\vec{P}(\vec{x}) - \vec{F}(\vec{P}(\vec{x}))\right)\vec{z} + \left(\vec{x} - \vec{P}(\vec{x})\right)\vec{z} = 0$$

2.tvrzení: Pro každé dva vektory $\vec{x}_1$ a $\vec{x}_2$ nadroviny ρ_1 platí, že velikost vektoru $\vec{F}(\vec{P}(\vec{x}_1)) - \vec{F}(\vec{P}(\vec{x}_2))$ je menší než velikost vektoru $\vec{x}_1 - \vec{x}_2$.

Důkaz: $\vec{z} = (\vec{P}(\vec{x}_1) - \vec{P}(\vec{x}_2)) - \vec{F}(\vec{P}(\vec{x}_1) - \vec{P}(\vec{x}_2))$

$$|\vec{x}_1 - \vec{x}_2|^2 = \left|\vec{P}(\vec{x}_1) - \vec{P}(\vec{x}_2)\right|^2 + \left|(\vec{x}_1 - \vec{x}_2) - (\vec{P}(\vec{x}_1) - \vec{P}(\vec{x}_2))\right|^2$$

$$\left|\vec{P}(\vec{x}_1) - \vec{P}(\vec{x}_2)\right|^2 = \left|\vec{F}(\vec{P}(\vec{x}_1)) - \vec{F}(\vec{P}(\vec{x}_2))\right|^2 + |\vec{z}|^2$$

Z výše uvedených tvrzení podle věty o pevném bodě plyne, že existuje právě jeden vektor $\vec{y} \in \mathbb{R}^n$ ležící v nadrovině ρ_1 tak, že platí:

$$\lim_{t\to\infty} \vec{x}(t) = \vec{F}(\vec{P}(\vec{y})) = \vec{y}$$

který současně leží v průniku nadroviny ρ s množinou σ, neboť z

$$\lim_{t\to\infty} \left|\vec{x}(t) - \vec{P}(\vec{x}(t))\right| \sin\varphi = \lim_{t\to\infty} |\vec{x}(t) - \vec{x}(t+1)| = 0$$

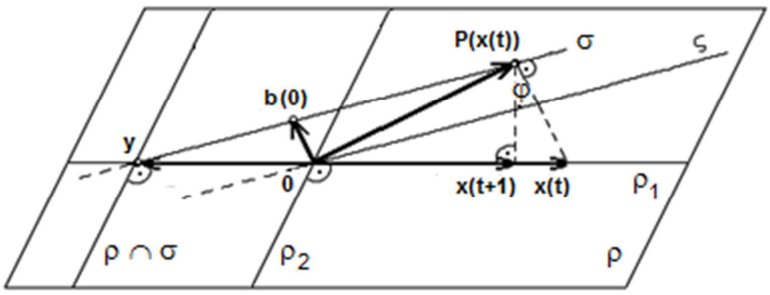
plyne

$$\vec{P}(\vec{y}) = \vec{y}$$

pak průnik nadroviny ρ s množinou σ je lineární posunutí nadroviny ρ_2 o vektor $\vec{y}$.

Stav sítě tedy během aktivní dynamiky alternujícími ortogonálními projekcemi $\vec{F}$ a $\vec{P}$ konverguje s libovolnou přesností ke stavu ležícímu v průniku nadroviny ρ s množinou σ v na něm kolmém směru, tj. ke stavu nejbližšímu ortogonální projekci vstupu sítě do nadroviny ρ. Doba odezvy sítě je závislá na volbě výše uvedené přesnosti. Pro $N \leq k$ zřejmě platí, že nadrovina ρ_1 je totožná s nadrovinou ρ a nadrovina ρ_2 obsahuje pouze nulový vektor.

Funkci sítě rekurentní autoasociativní paměti pak můžeme chápat podobně jako u lineární autoasociativní paměti jako filtraci šumu s tím, že můžeme zafixovat některé složky vstupu sítě $v_1, \cdots, v_k$ tak, že funkce sítě opraví pouze zbývající složky vstupu sítě $v_{k+1}, \cdots, v_n$. Nezafixujeme-li žádnou složku vstupu sítě, pak funkce sítě rekurentní autoasociativní paměti je totožná s funkcí sítě lineární autoasociativní paměti.



5. Hopfieldova autoasociativní paměť (HAM)

Topologie sítě:

Mějme populaci neuronů sítě V a propojme ji hranami tak, aby z každého neuronu populace směřovala hrana do každého neuronu populace kromě do sebe sama ($\varepsilon(E) = V \times V - \{[i,i]|i \in V\}, |V| = n$).

Aktivní dynamika: (synchronní)

Definujme aktivační funkci neuronů sítě jako modifikovanou ostrou nelinearitu ($f(x) = -1$ pro $x < 0$ a $f(x) = +1$ pro $x > 0$). Nastavme počáteční hodnoty potenciálů neuronů následovně:

$$x_i(0) = v_i$$

a mějme následující aktivní dynamiku:

$$y_j(t+1) = f_j(\sum_i y_i(t)\, w_{ij})$$

kde $i,j \in V$, vektor $\vec{v} = [v_i|i \in V]$ pak chápeme jako vstup sítě.

Adaptivní dynamika:

Definujme adaptační funkci neuronů sítě jako identitu. Mějme trénovací množinu ve tvaru $\{\vec{a}(T)|T \in \Delta T\}$. Položíme-li plasticitu resp. elasticitu rovnu jedné resp. nule, pak pro $i \neq j$ obdržíme následující adaptivní dynamiku:

$$w_{ij}(T) = w_{ij}(T-1) + a_i(T)\, a_j(T)$$

Položíme-li dále $\vec{w}(0) = \vec{0}$ a $\Delta T = \{1, \cdots, N\}$, pak pro $i \neq j$ po naučení sítě dostaneme:

$$w_{ij}(N) = \sum_T a_i(T)\, a_j(T)$$

kde $i,j \in V$ a $T \in \Delta T$.

Funkce sítě:

Definujme si *energetickou funkci* stavu sítě:

$$E(\vec{y}) = -\,½ \sum_j \sum_i y_i w_{ij} y_j$$

a určeme její následující parciální derivace:

$$-\frac{\partial E(\vec{y})}{\partial y_j} = \sum_i y_i w_{ij}$$

kde $i, j \in V$.

Dále si definujme *stabilní stav* sítě jako stav, pro který platí $\vec{y}(t) = \vec{y}(t+1)$ a *kvazistabilní stav* sítě jako stav, pro který platí $\vec{y}(t) = -\vec{y}(t+1)$.

1.tvrzení: Pro každý nestabilní stav sítě za předpokladu $y_i(t) = y_i(t+1)$ pro $i \in V - \{j\}$ (asynchronní aktivní dynamika) při $x_j(t+1) \neq 0$ platí, že hodnota energetické funkce stavu sítě v čase t je větší než hodnota energetické funkce stavu sítě v čase $t+1$.

Důkaz:

$$(\sum_i y_i(t)\, w_{ij})\, y_j(t) < (\sum_i y_i(t+1)\, w_{ij})\, y_j(t+1)$$

kde $i, j \in V$.

Z výše uvedeného tvrzení a z faktu, že energetická funkce je zdola omezená, vyplývá, že stav sítě během aktivní dynamiky kvazigradientním sestupem po energetické funkci konverguje ke stabilnímu stavu s lokálně minimální hodnotou energetické funkce.

Nechť trénovací množina obsahuje bipolární vektory tvořící ortogonální bázi nadroviny ρ v prostoru $\mathbb{R}^n$ ($N < n$) a definujme nadrovinu σ všech vektorů kolmých na každý vektor nadroviny ρ, pak prostor $\mathbb{R}^n$ je direktním součtem nadrovin ρ a σ ($\mathbb{R}^n = \rho \oplus \sigma$) a platí:

$$y_j(t+1) = f_j(y_{pj}(t) - \frac{N}{n} y_j(t)) \qquad E(\vec{y}) = -\,½\, n\, (\vec{y}\, \vec{y}_p - N)$$

kde $\vec{y}_p$ je ortogonální projekce stavu sítě $\vec{y}$ do nadroviny ρ, viz:

$$\sum_i y_i \sum_T a_i(T)\, a_j(T) = \sum_T a_j(T) \sum_i y_i a_i(T) = \sum_T a_j(T)(\vec{y}\ \vec{a}(T) - y_j a_j(T)) =$$

$$= \sum_T s_T a_j(T)\ \vec{a}(T)\vec{a}(T) - y_j \sum_T a_j(T)^2 = n\, y_{pj} - N\, y_j$$

$$\sum_j y_j \sum_i y_i \sum_T a_i(T)\, a_j(T) = \sum_j y_j \left(n\, y_{pj} - N\, y_j\right) =$$

$$= n \sum_j y_j y_{pj} - N \sum_j {y_j}^2 = n\, \vec{y}\, \vec{y}_p - N\, n$$

kde $j \in V$, $i \in V - \{j\}$, $T \in \Delta T$ a $s_1, \cdots, s_n$ jsou souřadnice vektoru $\vec{y}$ vzhledem k ortogonální bázi prostoru $\mathbb{R}^n$ obsahující vektory báze nadroviny ρ, viz $\vec{y} = s_1\vec{a}(1) + \cdots + s_n\vec{a}(n)$.

2.tvrzení: Každý stav sítě ležící v nadrovině ρ je stabilní.

Důkaz: pro každé $j \in V$ platí:

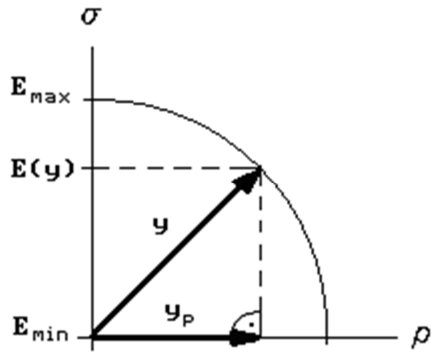
$$y_j(t+1) = f_j((1 - \frac{N}{n})y_j(t)) = y_j(t)$$

3.tvrzení: Každý stav sítě ležící v nadrovině σ je kvazistabilní.

Důkaz: pro každé $j \in V$ platí:

$$y_j(t+1) = f_j(-\frac{N}{n}\ y_j(t)) = -y_j(t)$$

Hodnota energetické funkce stavu sítě je tedy nepřímo úměrná skalárnímu součinu stavu sítě s jeho ortogonální projekcí do nadroviny ρ a bude minimální resp. maximální, právě když stav sítě bude ležet v nadrovině ρ resp. σ.



Z výše uvedených tvrzení plyne, že každý stav sítě, jenž je prvkem trénovací množiny, je stabilní a energetická funkce sítě v něm nabývá svého lokálního minima. Počáteční stav sítě během aktivní dynamiky tedy konverguje k prvku trénovací množiny, tzv. *atraktoru*, jehož hodnota energetické funkce leží na dně gradientního spádu, na jehož úbočí se hodnota energetické funkce počátečního stavu sítě nalézá, tj. leží-li počáteční stav sítě ve sféře vlivu daného atraktoru. Kvazistabilní stavy se nalézají na hranicích sfér vlivů různých atraktorů.

4.tvrzení: Stav sítě je stabilní právě když pro každé $j \in V$ platí:

$$1 \geq y_j \, y_{pj} > \frac{N}{n}$$

tj.

$$n \geq \vec{y} \, \vec{y}_p > N$$

Důkaz:

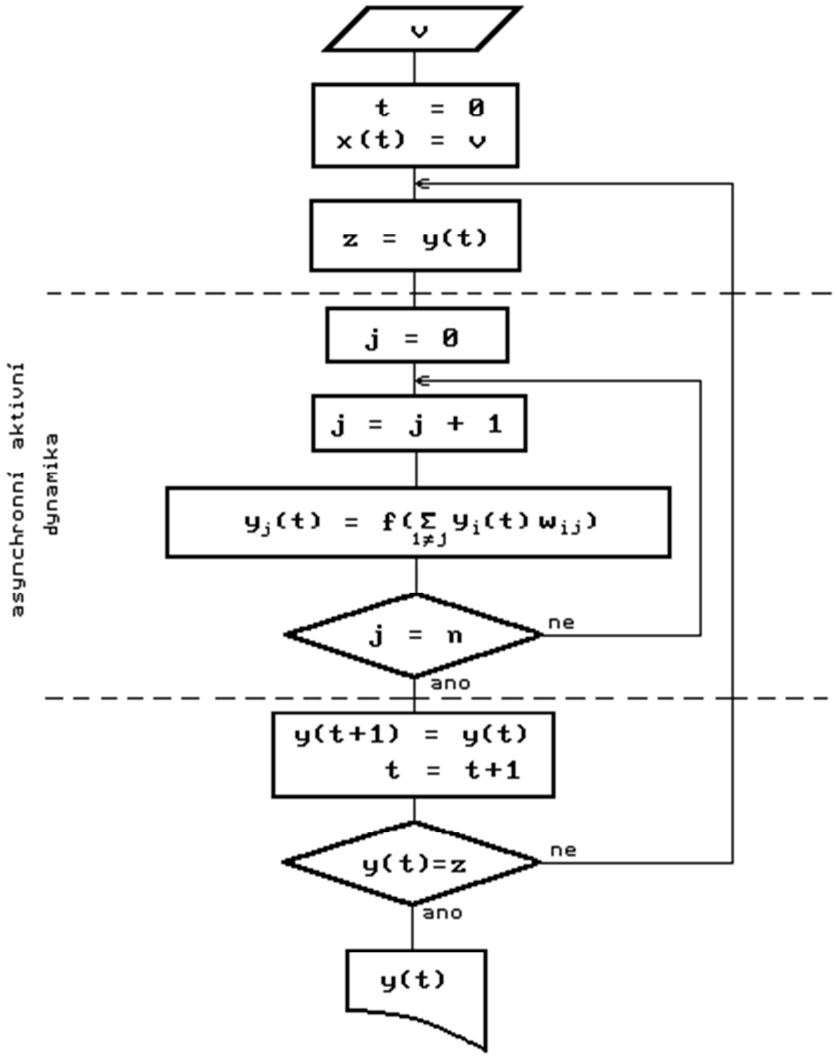
$$y_j(t+1) = f_j(y_{pj}(t) - \frac{N}{n} \, y_j(t)) = y_j(t)$$

kde $\vec{y}_p$ je ortogonální projekce stavu sítě $\vec{y}$ do nadroviny ρ.

Z výše uvedeného tvrzení plyne existence i takových stabilních stavů sítě, které nejsou prvky trénovací množiny ($\vec{y} \neq \vec{y}_p$), tzv. *falešných atraktorů*, k nimž počáteční stav sítě konverguje, nalézá-li se ve sféře jejich vlivu.

Funkce sítě pak libovolnému bipolárnímu vstupu sítě přiřadí buďto jemu nejbližší autoasociaci prvku trénovací množiny, leží-li k ní vstup sítě dostatečně blízko, anebo odezvu sítě nezávislou na prvcích trénovací množiny, leží-li vstup sítě dostatečně daleko od každého prvku trénovací množiny. Doba aktivace sítě závisí na vzdálenosti vstupu sítě od atraktoru resp. falešného atraktoru.

Funkci sítě Hopfieldovy autoasociativní paměti pak můžeme chápat podobně jako u lineární autoasociativní paměti jako filtraci šumu, přičemž z hlediska správné funkce sítě přípustné zkreslení atraktoru je dáno oblastí vymezenou sférou vlivu tohoto atraktoru.


v
t = 0
x(t) = v
z = y(t)
asynchronní aktivní dynamika
j = 0
j = j + 1
$y_j(t) = f(\sum_{i \neq j} y_i(t) w_{ij})$
j = n
ne
ano
y(t+1) = y(t)
t = t+1
y(t)=z
ne
ano
y(t)

6. Bidirektní heteroasociativní paměť (BAM)

Topologie sítě:

Rozdělme populaci neuronů sítě V na dvě disjunktní populace V_1 a V_2 ($V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$, $|V_1| = n$, $|V_2| = m$) a propojme je hranami tak, aby z každého neuronu populace V_1 směřovala hrana do každého neuronu populace V_2 ($\varepsilon(E_1) = V_1 \times V_2$) a z každého neuronu populace V_2 směřovala hrana do každého neuronu populace V_1 ($\varepsilon(E_2) = V_2 \times V_1$), síť je tedy orientována směrem od populace V_1 k populaci V_2 a zpět. Populaci V_1 pak chápeme jako populaci vstupní.

Aktivní dynamika: (synchronní)

Definujme aktivační funkci neuronů sítě jako modifikovanou ostrou nelinearitu ($f(x) = -1$ pro $x < 0$ a $f(x) = +1$ pro $x > 0$). Nastavme počáteční hodnoty potenciálů a stavů neuronů následovně:

$$x_i(0) = v_i \qquad y_j(0) = 0$$

a mějme následující aktivní dynamiku:

$$y_j(t+1) = f_j\left(\sum_i y_i(t)\, w_{ij}\right) \qquad y_i(t+2) = f_i\left(\sum_j y_j(t+1)\, w_{ji}\right)$$

kde $i \in V_1$, $j \in V_2$ a vektor $\vec{v} = [v_i | i \in V_1]$ pak chápeme jako vstup sítě.

Adaptivní dynamika:

Definujme adaptační funkci neuronů sítě jako identitu. Mějme trénovací množinu ve tvaru $\{[\vec{a}(T), \vec{b}(T)] | T \in \Delta T\}$, kde $\vec{a} = [a_i | i \in V_1]$ a $\vec{b} = [b_j | j \in V_2]$. Položíme-li plasticitu resp. elasticitu rovnu jedné resp. nule, pak obdržíme následující adaptivní dynamiku:

$$w_{ij}(T) = w_{ij}(T-1) + a_i(T)\, b_j(T) \qquad w_{ji}(T) = w_{ji}(T-1) + b_j(T)\, a_i(T)$$

Položíme-li dále $\vec{w}(0) = \vec{0}$ a $\Delta T = \{1, \cdots, N\}$, pak po naučení sítě dostaneme:

$$w_{ij}(N) = \sum_T a_i(T)\, b_j(T) \qquad w_{ji}(N) = \sum_T b_j(T)\, a_i(T)$$

kde $i \in V_1$, $j \in V_2$ a $T \in \Delta T$.

Funkce sítě:

Definujme si *energetickou funkci* stavu sítě:

$$E(\vec{y}) = -\sum_j \sum_i y_i w_{ij} y_j$$

a určeme její následující parciální derivace:

$$-\frac{\partial E(\vec{y})}{\partial y_j} = \sum_i y_i w_{ij} \qquad -\frac{\partial E(\vec{y})}{\partial y_i} = \sum_j y_j w_{ji}$$

kde $i \in V_1, j \in V_2$.

Dále si definujme *stabilní stav* sítě jako stav, pro který platí $\vec{y}(t) = \vec{y}(t+2)$.

1.tvrzení: Pro každý nestabilní stav sítě za předpokladu $y_j(t) = y_j(t+1)$ pro $j \in V_2 - \{k\}$ (asynchronní aktivní dynamika) při $x_k(t+1) \neq 0$ platí, že hodnota energetické funkce stavu sítě v čase t je větší než hodnota energetické funkce stavu sítě v čase $t+1$.

Důkaz:

$$(\sum_i y_i(t)\, w_{ij})\, y_k(t) < (\sum_i y_i(t+1)\, w_{ij})\, y_k(t+1)$$

kde $i \in V_1$ a $j, k \in V_2$.

2.tvrzení: Pro každý nestabilní stav sítě za předpokladu $y_i(t+1) = y_i(t+2)$ pro $i \in V_1 - \{k\}$ (asynchronní aktivní dynamika) při $x_k(t+2) \neq 0$ platí, že hodnota energetické funkce stavu sítě v čase $t+1$ je větší než hodnota energetické funkce stavu sítě v čase $t+2$.

Důkaz:

$$(\sum_j y_j(t+1)\, w_{ji})\, y_k(t+1) < (\sum_j y_j(t+2)\, w_{ji})\, y_k(t+2)$$

kde $i, k \in V_1$ a $j \in V_2$.

Z výše uvedených tvrzení a z faktu, že energetická funkce je zdola omezená, vyplývá, že stav sítě během aktivní dynamiky kvazigradientním sestupem po energetické funkci konverguje ke stabilnímu stavu s lokálně minimální hodnotou energetické funkce.

Nechť vektory trénovací množiny $\vec{a}(T)$ resp. $\vec{b}(T)$ jsou bipolární vektory tvořící ortogonální bázi nadroviny ρ_a resp. ρ_b v prostoru $\mathbb{R}^n$ resp. $\mathbb{R}^m$ ($N < n, m$), pak prvky trénovací množiny tvoří ortogonální bázi nadroviny ρ v prostoru $\mathbb{R}^{n+m}$ a platí:

$$y_j(t+1) = f_j(y_{pj}(t)) \qquad y_i(t+2) = f_i(y_{pi}(t+1))$$

$$E(\vec{y}) = -\,½\,(n\,\vec{y}_b\vec{y}_{bp} + m\,\vec{y}_a\vec{y}_{ap})$$

tj. pro $n = m$:

$$E(\vec{y}) = -\,½\,n\,\vec{y}\,\vec{y}_p$$

kde $\vec{y}_{ap}$, $\vec{y}_{bp}$, $\vec{y}_p$ jsou ortogonální projekce vektorů $\vec{y}_a$, $\vec{y}_b$, $\vec{y}$ do nadrovin ρ_a, ρ_b, ρ, viz:

$$\sum_i y_i \sum_T a_i(T)\,b_j(T) = \sum_T b_j(T) \sum_i y_i a_i(T) =$$

$$\sum_T b_j(T)\,\vec{y}_a\,\vec{a}(T) = \sum_T s_T b_j(T)\,\vec{a}(T)\vec{a}(T) = n\,y_{pj}$$

$$\sum_j y_j \sum_T b_j(T)\,a_i(T) = \sum_T a_i(T) \sum_j y_j b_j(T) =$$

$$\sum_T a_i(T)\,\vec{y}_b\,\vec{b}(T) = \sum_T s_{n+T} a_i(T)\;\vec{b}(T)\vec{b}(T) = m\,y_{pi}$$

$$\sum_j y_j \sum_i y_i \sum_T a_i(T)\,b_j(T) \;= n \sum_j y_j y_{pj} = n\,\vec{y}_b\,\vec{y}_{bp}$$

$$\sum_i y_i \sum_j y_j \sum_T b_j(T)\,a_i(T) \;= m \sum_i y_i y_{pi} = m\,\vec{y}_a\,\vec{y}_{ap}$$

kde $i \in V_1$, $j \in V_2$, $T \in \Delta T$, $\vec{y}_a = [y_i | i \in V_1]$, $\vec{y}_b = \left[y_j | j \in V_2\right]$ a $s_1, \cdots, s_{n+m}$ jsou souřadnice vektoru $\vec{y} = [\vec{y}_a, \vec{y}_b]$ vzhledem k ortogonální bázi prostoru $\mathbb{R}^{n+m}$ obsahující vektory báze nadroviny ρ, viz $\vec{y}_a = s_1\vec{a}(1) + \cdots + s_n\vec{a}(n)$, $\vec{y}_b = s_{n+1}\vec{b}(1) + \cdots + s_{n+m}\vec{b}(m)$.

Hodnota energetické funkce stavu sítě je tedy nepřímo úměrná skalárnímu součinu stavu sítě s jeho projekcí do nadroviny ρ a bude minimální, právě když stav sítě bude ležet v nadrovině ρ.

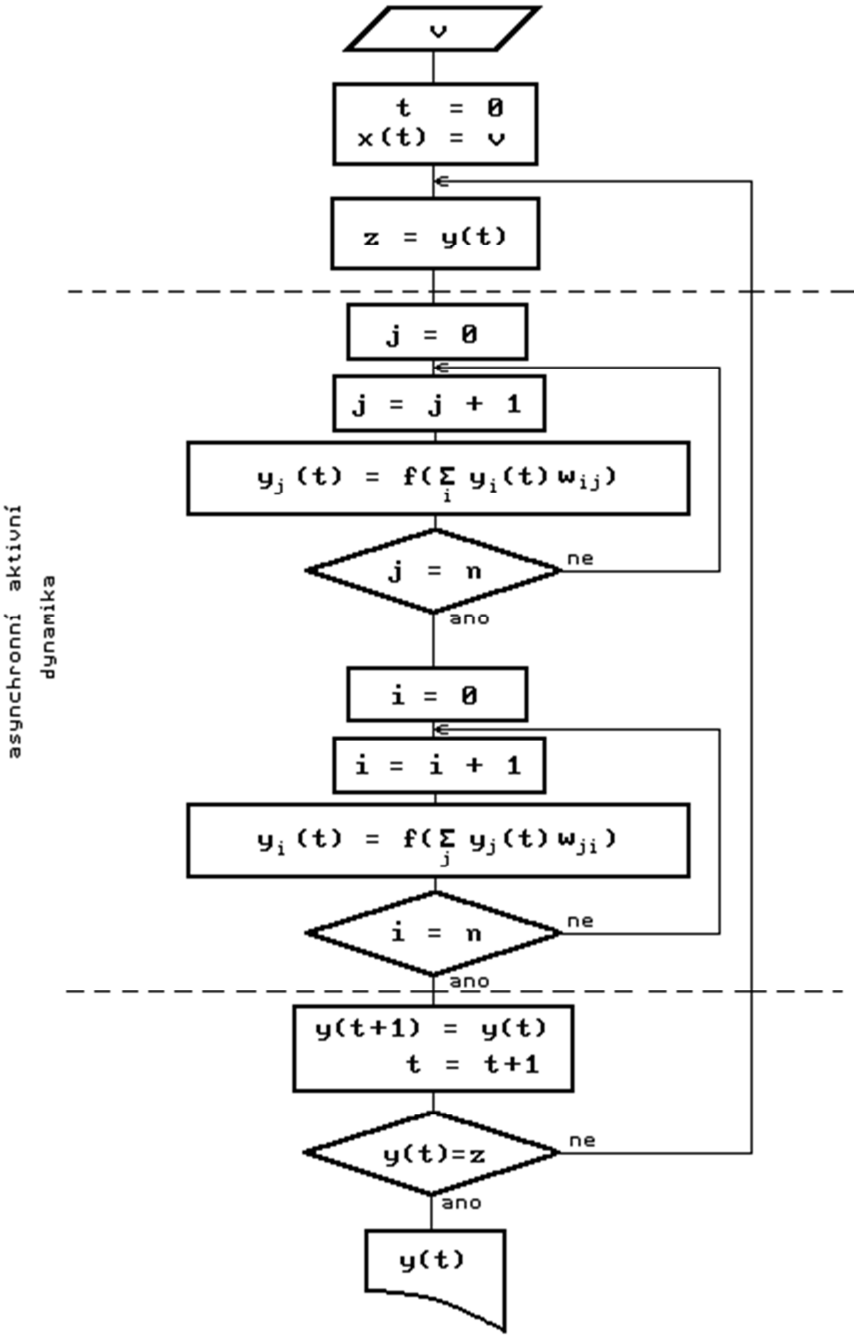
3.tvrzení : Každý stav sítě ležící v nadrovině ρ je stabilní.

Důkaz: pro každé $j \in V_2$ a $i \in V_1$ platí:

$$y_j(t+1) = f_j(y_j(t)) = y_j(t) \qquad y_i(t+2) = f_i(y_i(t+1)) = y_i(t+1)$$

Z výše uvedeného tvrzení plyne, že každý stav sítě, jenž je prvkem trénovací množiny, tzv. *atraktor*, je stabilní a energetická funkce sítě v něm nabývá svého lokálního minima. Analogicky s Hopfieldovou autoasociativní pamětí však mohou vznikat i *falešné atraktory*, tj. stabilní stavy neobsažené v trénovací množině.

Funkce sítě pak bipolárnímu vstupu sítě, ležícímu dostatečně blízko nějakému vzoru $\vec{a}$, nalezením stability mezi oscilujícími stavy vstupní a výstupní populace, tzv. *rezonancí*, přiřadí jeho heteroasociaci, tj. jeho obraz $\vec{b}$.


v
t = 0
x(t) = v
z = y(t)
asynchronní aktivní dynamika
j = 0
j = j + 1
$y_j(t) = f(\sum_i y_i(t) w_{ij})$
j = n
ne
ano
i = 0
i = i + 1
$y_i(t) = f(\sum_j y_j(t) w_{ji})$
i = n
ne
ano
y(t+1) = y(t)
t = t+1
y(t)=z
ne
ano
y(t)

7. Optimalizační model (CHN)

Topologie sítě:

Mějme populaci neuronů sítě V a propojme ji hranami tak, aby z každého neuronu populace směřovala hrana do každého neuronu populace kromě do sebe sama ($\varepsilon(E) = V \times V - \{[i,i]|i \in V\}, |V| = n$).

Aktivní dynamika: (synchronní)

Definujme aktivační funkci neuronů sítě jako sigmoidu. Nastavme počáteční hodnoty stavů neuronů následovně:

$$y_i(0) = 0$$

a mějme následující aktivní dynamiku:

$$y_j(t+1) = f_j(\sum_i y_i(t)\, w_{ij} - \vartheta_j)$$

kde $i, j \in V$.

Adaptivní dynamika:

Konfigurace sítě se určuje extrakcí z objektivní funkce.

Funkce sítě:

Definujme si *energetickou funkci* stavu sítě:

$$E(\vec{y}) = -\,(½ \sum_j \sum_i y_i w_{ij} y_j - \sum_j y_j \vartheta_j)$$

a určeme její následující parciální derivace:

$$-\frac{\partial E(\vec{y})}{\partial y_j} = \sum_i y_i w_{ij} - \vartheta_j \quad -\frac{\partial E(\vec{0})}{\partial y_j} = -\vartheta_j \quad -\frac{\partial^2 E(\vec{y})}{\partial y_i \partial y_j} = w_{ij}$$

kde $i, j \in V$.

Definujme *stabilní stav* sítě jako stav, pro který platí $\vec{y}(t) = \vec{y}(t+1)$ a parametr T jako převrácenou hodnotu strmosti sigmoidy.

Tvrzení: Bude-li se blížit hodnota parametru T k nule ($T \to 0$), pak pro každý nestabilní stav sítě za předpokladu $y_i(t) = y_i(t+1)$ pro $i \in V - \{j\}$ (asynchronní aktivní dynamika) při $x_j(t+1) \neq 0$ a platnosti $w_{ij} = w_{ji}$ pro $i, j \in V$, bude platit, že hodnota energetické funkce stavu sítě v čase t bude větší než hodnota energetické funkce stavu sítě v čase $t+1$, tj. $E(\vec{y}(t)) > E(\vec{y}(t+1))$.

Důkaz:

$$(\sum_i y_i(t)\, w_{ij} - \vartheta_j)\, y_j(t) < (\sum_i y_i(t+1)\, w_{ij} - \vartheta_j)\, y_j(t+1)$$

neboť:

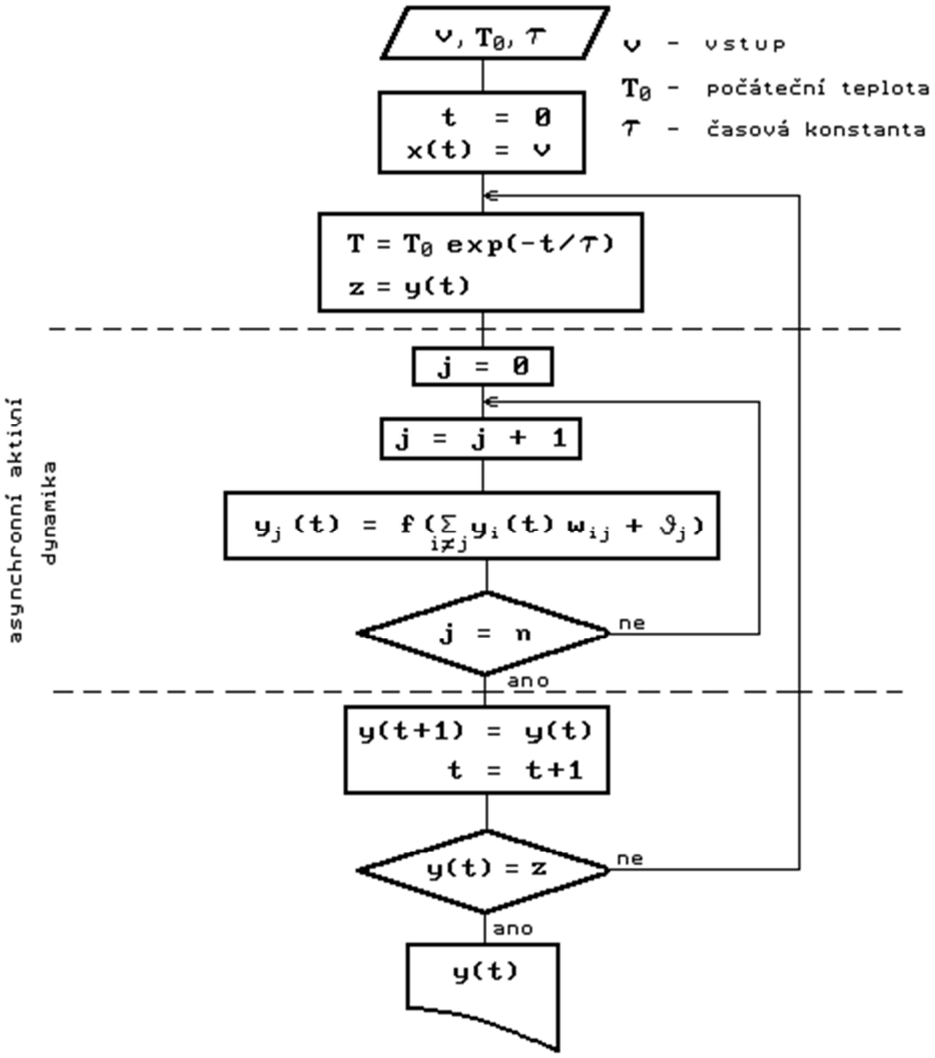
$$x_j(t+1) > 0 \Rightarrow y_j(t+1) = 1 \Rightarrow y_j(t) = 0$$
$$x_j(t+1) < 0 \Rightarrow y_j(t+1) = 0 \Rightarrow y_j(t) = 1$$

kde $i, j \in V$.

Z výše uvedeného tvrzení a z faktu, že energetická funkce je zdola omezená, vyplývá, že stav sítě během aktivní dynamiky kvazigradientním sestupem po energetické funkci konverguje ke stabilnímu stavu s lokálně minimální hodnotou energetické funkce.

Při vyšších hodnotách parametru T může být konvergence k lokálnímu minimu energetické funkce na dně menšího gradientního spádu narušena, čehož lze využít k jeho přeskočení. Budeme-li tedy hodnotu parametru T z dostatečně vysoké počáteční hodnoty během aktivní dynamiky postupně snižovat, pak po přeskočení mělkých lokálních minim při dostatečně nízké hodnotě parametru T nám stav sítě uvízne v oblasti hlubokého minima energetické funkce na dně většího gradientního spádu a zkonverguje ke stabilnímu stavu s přibližně globálně minimální hodnotou energetické funkce, tj. *zamrzne* v globálním minimu. Parametr T tedy hraje roli *teploty* z optimalizační metody simulovaného žíhání.

Funkce sítě pak vstupu sítě přiřadí binární stav sítě s minimální hodnotou energetické funkce, čehož lze využít k hledání minim funkcí ve tvaru kvadratické formy na oblasti vymezené binárními hodnotami nezávisle proměnných. Doba aktivace sítě závisí na vzdálenosti vstupu sítě od výše uvedeného stavu sítě a volbě počáteční teploty a strmosti jejího poklesu. Konfigurace sítě se extrahuje z objektivní funkce určením jejích výše uvedených parciálních derivací.


v, T0, τ
v - vstup
T0 - počáteční teplota
τ - časová konstanta
t = 0
x(t) = v
T = T0 exp(-t/τ)
z = y(t)
asynchronní aktivní dynamika
j = 0
j = j + 1
yj (t) = f (Σ i≠j yi(t) wij + ϑj)
j = n
ne
ano
y(t+1) = y(t)
t = t+1
y(t) = z
ne
ano
y(t)

7.1. Problém obchodního cestujícího řešený optimalizačním modelem

Nechť mohutnost populace neuronů optimalizačního modelu je rovna hodnotě m^2 ($m \in \mathbb{N}$), pak můžeme neurony zmíněné populace interpretovat jako prvky čtvercové matice dimenze $m \times m$.

Definujme si následující funkce stavu sítě:

$$G_1(\vec{y}) = ½ \sum_i (\sum_j y_{ij} - 1)^2 \qquad G_2(\vec{y}) = ½ \sum_j (\sum_i y_{ij} - 1)^2$$

$$G_3(\vec{y}) = \sum_i \sum_j \rho_{ij} \sum_k y_{ik} y_{jk+1}$$

kde $i, j \in \{1, \cdots, m\}$, $k \in \{1, \cdots, m-1\}$ a ρ_{ij} je příslušná metrika.

Určeme jejich parciální derivace:

$$-\frac{\partial G_1(\vec{0})}{\partial y_{ij}} = 1 \qquad -\frac{\partial^2 G_1(\vec{y})}{\partial y_{ij} \partial y_{ik}} = -1$$

$$-\frac{\partial G_2(\vec{0})}{\partial y_{ij}} = 1 \qquad -\frac{\partial^2 G_2(\vec{y})}{\partial y_{ij} \partial y_{kj}} = -1$$

kde $i, j, k \in \{1, \cdots, m\}$ a

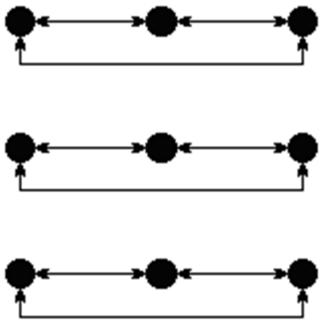
$$-\frac{\partial G_3(\vec{0})}{\partial y_{ij}} = 0 \qquad -\frac{\partial^2 G_3(\vec{y})}{\partial y_{ik} \partial y_{jk+1}} = -\rho_{ij}$$

kde $i, j \in \{1, \cdots, m\}$ a $k \in \{1, \cdots, m-1\}$.

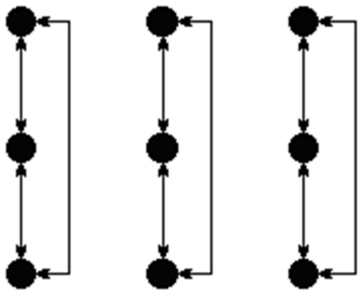
Definujeme-li objektivní funkci G jako součet funkcí G_1, G_2, G_3 a určíme-li konfiguraci a vstup sítě optimalizačního modelu extrakcí z funkce G užitím výše uvedených parciálních derivací, pak se objektivní funkce G ztotožní s energetickou funkcí E optimalizačního modelu, jejíž minimalizace proběhne během aktivní dynamiky.

Minimalizací funkce G_1 resp. G_2 zajistíme excitaci právě jednoho neuronu v každém řádku resp. sloupci zmíněné matice, čímž obdržíme tzv. *přípustný stav* sítě, interpretujeme-li si totiž řádky resp. sloupce zmíněné matice jako města resp. pořadí jejich návštěvy obchodním cestujícím, pak každý přípustný stav sítě představuje jisté řešení problému obchodního cestujícího, ovšem ne nutně optimální.

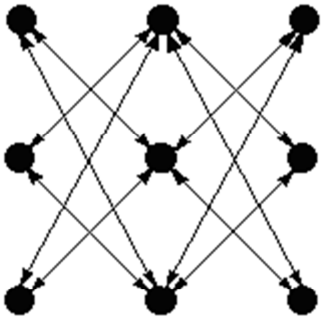
Minimalizací funkce G_3 zajistíme nalezení optimálního přípustného stavu sítě, tj. optimálního řešení problému obchodního cestujícího, neboť hodnota funkce G_3 udává délku cesty obchodního cestujícího v závislosti na přípustném stavu.



vazby zajišťující minimalizaci G_1



vazby zajišťující minimalizaci G_2



vazby zajišťující minimalizaci G_3

8. Kompetiční model

Topologie sítě:

Rozdělme populaci neuronů sítě V na dvě disjunktní populace V_1 a V_2 ($V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$, $|V_1| = n$, $|V_2| = m$) a propojme je hranami tak, aby z každého neuronu populace V_1 směřovala hrana do každého neuronu populace V_2 ($\varepsilon(E_1) = V_1 \times V_2$), síť je tedy orientována směrem od populace V_1 k populaci V_2 a populaci V_1 resp. V_2 pak chápeme jako populaci vstupní resp. výstupní. Dále propojme neurony populace V_2 hranami tak, aby z každého neuronu populace směřovala hrana do každého neuronu populace kromě do sebe sama ($\varepsilon(E_2) = V_2 \times V_2 - \{[j,j]|j \in V_2\}$).

Aktivní dynamika:

Definujme aktivační funkci neuronů populace V_1 resp. V_2 jako identitu resp. ostrou nelinearitu. Nastavme počáteční hodnoty potenciálů a stavů neuronů následovně:

$$x_i(0) = v_i \qquad\qquad y_j(0) = 0$$

a mějme následující aktivní dynamiku:

$$y_j(t+1) = f_j\left(\sum_k y_k(t)\, w_{kj} - \vartheta_j\right) \qquad -\vartheta_j = \sum_i v_i\, w_{ij}$$

kde $i \in V_1$, $j,k \in V_2$ a parametr $-\vartheta_j$ pak představuje zisk potenciálu j-tého neuronu výstupní populace od neuronů vstupní populace. Vektor $\vec{v} = [v_i|i \in V_1]$ pak chápeme jako vstup sítě.

Adaptivní dynamika:

Nastavme počáteční hodnoty vah následovně:

$$w_{ij}(0) = r_{ij} \qquad\qquad w_{kj}(0) = -2$$

a definujme adaptační funkci neuronů výstupní populace jako ostrou nelinearitu. Mějme trénovací množinu ve tvaru $\{\vec{a}(T)|T \in \Delta T\}$, kde $\vec{a} = [a_i|i \in V_1]$. Necháme-li adaptovat pouze vazby mezi neurony populace V_1 a V_2 a položíme-li plasticitu rovnu elasticitě ($\alpha = \beta$), pak obdržíme následující adaptivní dynamiku:

$$w_{ij}(T) = w_{ij}(T-1) + \alpha y_j(T)(a_i(T) - w_{ij}(T-1))$$

$$w_{kj}(T) = w_{kj}(T-1)$$

kde $i \in V_1, j, k \in V_2, T \in \Delta T = \{1, -, N\}$ a r_{ij} je náhodně generovaná hodnota.

V každém kroku adaptivní dynamiky je nutné určit stavy neuronů populace V_2, kroky adaptivní dynamiky jsou tedy podmíněny dynamikou aktivní, která z pohledu adaptivní dynamiky běží nekonečně rychle, takže stav populace V_2 se během adaptivní dynamiky určí synchronně se stavem populace V_1.

Stav populace V_2 v každém adaptačním kroku určíme pomocí aktivní dynamiky s následujícím počátečním nastavením hodnot potenciálů neuronů populace V_1:

$$x_i(0) = a_i(T)$$

kde $i \in V_1$.

Přiřaďme každému neuronu populace V_2 váhový vektor $\vec{w}_j = [w_{ij} | i \in V_1]$, potom neurony populace V_2 spolu s hranami E_2 a aktivní dynamikou tvoří spojitou Hopfieldovu síť s energetickou funkcí:

$$E(\vec{y}) = \sum_j \sum_k y_k y_j + \sum_j y_j \vartheta_j \qquad -\vartheta_j = \sum_i a_i(T)\, w_{ij} = \vec{a}(T)\, \vec{w}_j$$

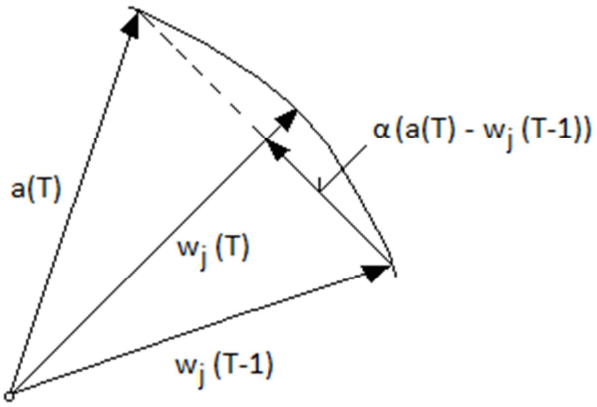
kde $i \in V_1, j \in V_2, k \in V_2 - \{j\}$.

Budou-li vektory trénovací množiny resp. váhové vektory normální, pak vzdálenost mezi uvedenými vektory můžeme definovat jako jimi sevřený úhel $\varphi \in < 0, \pi >$ (neeuklidovská metrika) a pro zisk potenciálu pak bude platit $-\vartheta_j = \cos\varphi\ (\vec{a}(T), \vec{w}_j)$. Výše uvedená energetická funkce pak nabude svého minima, právě když bude excitován právě jeden neuron populace V_2 a to neuron s maximálním ziskem potenciálu, tzv. *gain* neuron.

Proces minimalizace energie stavu populace V_2, realizovaný aktivní dynamikou, kdy excitovaný neuron populace s maximálním ziskem potenciálu inhibuje prostřednictvím záporných interních vazeb ostatní neurony populace, označíme jako *laterální inhibici*. Laterální inhibice, určující na základě předloženého vzoru jemu odpovídající stav populace V_2, nahrazuje v trénovací množině chybějící asociaci vzoru, tj. jeho obraz neboli výrok učitele, a mluvíme tedy o učení bez učitele.

Laterální inhibice v každém adaptačním kroku zabezpečí adaptaci pouze váhového vektoru odpovídajícího *j*-tému gain neuronu, tj. váhového vektoru ve smyslu výše uvedené neeuklidovské metriky nejbližšího předloženému vzoru, k němuž se po povrchu *n*-rozměrné hyperkoule jednotkového poloměru přiblíží o adaptační krok úměrný míře plasticity synapse:

$$\vec{w}_j(T) = \vec{w}_j(T-1) + \alpha\,(\vec{a}(T) - \vec{w}_j(T-1))$$



pak řekneme, že *j*-tý gain neuron vyhrál *kompetici* o předložený vzor trénovací množiny. Normalitu vektoru zadaptovaných vah zajistíme jeho normalizací po adaptaci.

Definujme objektivní funkci:

$$G(\vec{w}_j) = \tfrac{1}{2} \sum_T y_j(T) \sum_i (a_i(T) - w_{ij})^2$$

a určeme její následující parciální derivace:

$$-\frac{\partial G(\vec{w}_j)}{\partial w_{ij}} = \sum_T y_j(T)\big(a_i(T) - w_{ij}\big)$$

kde $i \in V_1$, $j \in V_2$ a $T \in \Delta T$.

Objektivní funkce nabude svého minima, právě když váhový vektor zaujme polohu s minimálním součtem jeho vzdáleností od všech vektorů trénovací množiny excitujících příslušný neuron, tj. ve smyslu euklidovské metriky ve středu uvedených vektorů.

Jelikož výše uvedená adaptivní dynamika je kvazigradientním sestupem po zdola omezené objektivní funkci, tak za předpokladu, že vektory trénovací množiny tvoří ve vstupním prostoru shluky, jejichž počet odpovídá mohutnosti populace V_2, budou náhodně nastavené váhové vektory během adaptivní dynamiky konvergovat ke středům těchto shluků.

Funkce sítě:

Definujme si následující kategorie normálních vektorů, do kterých byl během adaptivní dynamiky proveden rozklad oblasti normálních vektorů obsahující trénovací množinu ($N \gg n, m$):

$$C_k = \{\vec{x} \in \Omega | \varphi(\vec{x}, \vec{w}_k) \leq \varphi(\vec{x}, \vec{w}_j)\} \quad \Omega = \{\vec{x} \in \mathbb{R}^n || \vec{x}| = 1\}$$

kde $k \in V_2, j \in V_2 - \{k\}$ a φ je neeuklidovská metrika, tj. úhel mezi uvedenými vektory.

Funkce sítě tedy v aktivním režimu libovolnému normálnímu vstupu sítě přiřadí vektor kanonické báze s jednotkou na *k*-té pozici právě když vstup sítě leží v *k*-té kategorii. Doba aktivace sítě je shodná s dobou aktivace optimalizačního modelu.

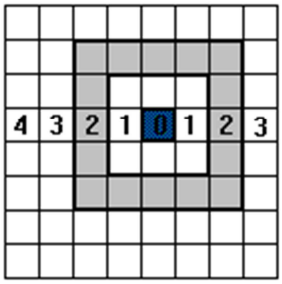
Funkci sítě kompetičního modelu pak můžeme chápat jako klasifikační z hlediska výše uvedených kategorií.

8.1. Kohonenova mapa (SOM)

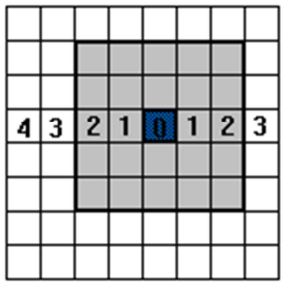
Položíme-li $|V_2| = m^2$, pak můžeme neurony populace V_2 interpretovat jako prvky čtvercové mřížky o rozměru $m \times m$. Definujme čtvercové okolí *r*-tého řádu *k*-tého prvku mřížky jako množinu obsahující všechny prvky mřížky ležící ve vzdálenosti menší nebo rovné řádu *r*, tj. $\sigma(k,r) = \{j \in V_2 | \rho(k,j) \leq r\}$, kde ρ je metrika definovaná na zmíněné mřížce jako sousedství prvků příslušného řádu, a upravme adaptivní dynamiku pro *k*-tý gain neuron:

$$\vec{w}_j(T) = \vec{w}_j(T-1) + \alpha_j(T)(\vec{a}(T) - \vec{w}_j(T-1))$$

kde $j \in \sigma(k,r)$ a míra plasticity klesá globálně s časem adaptivní dynamiky a lokálně s řádem vzdálenosti příslušného neuronu v mřížce populace V_2 od gain neuronu.



sousedé gain neuronu 2. řádu



čtvercové okolí gain neuronu 2. řádu

Uvedenou úpravou adaptivní dynamiky jsme zobecnili laterální inhibici rozšířením excitace gain neuronu na jeho okolí, čímž jsme provázali metriku φ s metrikou ρ.

Budou-li vektory trénovací množiny náhodně rozděleny v *n*-rozměrném prostoru v souladu s nějakou distribuční funkcí, pak po adaptaci sítě budou váhové vektory náhodně rozděleny v tomtéž prostoru v souladu se stejnou distribuční funkcí.

Předložíme-li v aktivním režimu trénovací množinu na ní zadaptované síti, pak mapa četnosti excitací neuronů populace V_2, tzv. *Kohonenova mapa*, nám zobrazí rozmístění vzorů trénovací množiny v *n*-rozměrném prostoru.

Takto zobecněný kompetiční model za předpokladu dostatečně velké mohutnosti populace V_2 provádí shlukovou analýzu trénovací množiny, tj. určení počtu shluků včetně jejich rozmístění v *n*-rozměrném prostoru.

8.2. Vstřícné šíření vah (CPN)

Upravme topologii již zadaptovaného kompetičního modelu přidáním populace V_3 a propojme ji hranami s populací V_2 tak, aby z každého neuronu populace V_2 směřovala hrana do každého neuronu populace V_3 ($\varepsilon(E_3) = V_2 \times V_3$). Nechť nová výstupní populace V_3 má stejnou mohutnost jako vstupní populace V_1. Váhy hran E_3 nastavme následovně:

$$w_{jp(i)} = w_{ij}$$

kde $i \in V_1$, $j \in V_2$, $p(i) \in V_3$ a $p(i)$ je bijektivní obraz *i*-tého neuronu vstupní populace ve výstupní populaci.

Výstupní populace V_3 spolu s ohodnocenými hranami E_3 je tedy obrazem vstupní populace V_1 spolu s ohodnocenými hranami E_1 zrcadlícím se přes nyní skrytou populaci V_2, mluvíme o *vstřícném šíření* synaptických vah hran E_1 na hrany E_3 ve směru orientace hran.

Zvolme aktivační funkce neuronů populace V_3 shodně s aktivačními funkcemi populace V_1, pak během aktivní dynamiky po stabilizaci stavu populace V_2 excitací *k*-tého gain neuronu dostaneme hodnoty potenciálů neuronů populace V_3 následovně:

$$x_{p(i)} = \sum_j y_j w_{jp(i)} = w_{kp(i)} = w_{ik}$$

kde $i \in V_1$, $j \in V_2$, z čehož pro stimul $\vec{x} \in C_k$ plyne následující funkce sítě:

$$\vec{F}(\vec{x}) = \vec{w}_k$$

Funkce sítě kompetičního modelu se vstřícným šířením vah tedy libovolnému normálnímu vstupu sítě přiřadí jeho *prototyp*, tj. jemu nejbližší váhový vektor. Prototypy leží v centrech příslušných shluků a tím uvedené shluky reprezentují, jsou jejich typickými představiteli.

Kompetiční model se vstřícným šířením vah a Kohonenovou mapou můžeme užít pro redukci mohutnosti trénovací množiny, nahradíme-li ji množinou prototypů jejích prvků o volitelné mohutnosti m^2 ($N \gg m^2$).

9. Nelineární model (MLP)

Topologie sítě:

Rozdělme populaci neuronů sítě V na tři disjunktní populace V_1, V_2 a V_3 ($V_1 \cup V_2 \cup V_3 = V,\ V_1 \cap V_2 = \emptyset,\ V_1 \cap V_3 = \emptyset,\ V_2 \cap V_3 = \emptyset,\ |V_1| = n,\ |V_2| = ?,\ |V_3| = m$) a propojme je hranami tak, aby z každého neuronu populace V_1 směřovala hrana do každého neuronu populace V_2 ($\varepsilon(E_1) = V_1 \times V_2$) a z každého neuronu populace V_2 směřovala hrana do každého neuronu populace V_3 ($\varepsilon(E_2) = V_2 \times V_3$), síť je tedy orientována směrem od populace V_1 k populaci V_3. Populaci V_1 resp. V_3 pak chápeme jako populaci vstupní resp. výstupní a populaci V_2 nazveme *skrytou* populací implikující nelinearitu funkce sítě.

Aktivní dynamika:

Definujme aktivační funkci neuronů sítě jako sigmoidu. Nastavme počáteční hodnoty potenciálů neuronů následovně:

$$x_i(0) = v_i$$

a mějme následující aktivní dynamiku:

$$y_k(1) = f_k(\sum_i y_i(0)\, w_{ik})$$
$$y_j(2) = f_j(\sum_k y_k(1)\, w_{kj})$$

kde $i \in V_1$, $k \in V_2$, $j \in V_3$ a vektor $\vec{v} = [v_i | i \in V_1]$ pak chápeme jako vstup sítě.

Adaptivní dynamika:

Nastavme počáteční hodnoty vah následovně:

$$w_{ik}(0) = r_{ik} \qquad\qquad w_{kj}(0) = r_{kj}$$

Položíme-li elasticitu rovnu nule, pak dostaneme následující adaptivní dynamiku:

$$w_{ik}(T) = w_{ik}(T-1) + \alpha\, y_i(T)\, g_k(x_k(T))$$
$$w_{kj}(T) = w_{kj}(T-1) + \alpha\, y_k(T)\, g_j(x_j(T))$$

kde $i \in V_1$, $k \in V_2$, $j \in V_3$, $T \in \Delta T$ a r_{ik} resp. r_{kj} jsou náhodně generované hodnoty.

Mějme trénovací množinu ve tvaru $\{[\vec{a}(T), \vec{b}(T)]|T \in \Delta T\}$, kde $\vec{a} = [a_i|i \in V_1]$, $\vec{b} = [b_j|j \in V_3]$ a definujme globální *chybovou funkci* funkce sítě:

$$E(\vec{w}) = ½ \sum_T \sum_j (F_j(\vec{a}(T)) - f_j(b_j(T)))^2 = \sum_T E_T$$

kde $j \in V_3$ a $T \in \Delta T$.

Globální chybová funkce vyjadřuje součet hodnot lokálních chyb E_T, tj. součet rozdílů mezi skutečnou a požadovanou odezvou sítě přes všechny prvky trénovací množiny při aktuálně nastavené konfiguraci sítě a její funkční hodnotu lze proto chápat jako aktuální chybu funkce sítě vzhledem k trénovací množině. Funkce sítě definovaná během adaptace sítě trénovací množinou, bude bezchybná právě když $E(\vec{w}) = 0$.

Upozornění: V dalším pro zjednodušení zápisu u potenciálů a stavů neuronů nebudeme explicitně uvádět čas adaptivní dynamiky jakožto jejich argument, budeme implicitně v zápisech předpokládat jeho synchronizaci s indexem lokální chyby.

Definujme adaptační funkce neuronů výstupní a skryté populace:

$$g_j(x_j) = -\frac{\partial E_T}{\partial x_j} \qquad g_k(x_k) = \frac{dy_k}{dx_k} \sum_j g_j(x_j)\, w_{kj}$$

kde $k \in V_2$ a $j \in V_3$.

Určeme následující parciální derivace:

$$\frac{\partial E_T}{\partial w_{kj}} = \frac{\partial E_T}{\partial x_j}\frac{\partial x_j}{\partial w_{kj}} = -y_k\, g_j(x_j)$$

$$\frac{\partial E_T}{\partial w_{ik}} = \sum_j \frac{\partial E_T}{\partial x_j}\frac{\partial x_j}{\partial y_k}\frac{dy_k}{dx_k}\frac{\partial x_k}{\partial w_{ik}} = -y_i\, g_k(x_k)$$

kde:

$$\frac{\partial x_j}{\partial y_k} = w_{kj} \qquad \frac{\partial x_k}{\partial w_{ik}} = y_i$$

pro $i \in V_1$, $k \in V_2$, $j \in V_3$.

Adaptivní dynamiku pak dostaneme ve tvaru:

$$w_{ik}(T) = w_{ik}(T-1) - \alpha \frac{\partial E_T}{\partial w_{ik}}$$

$$w_{kj}(T) = w_{kj}(T-1) - \alpha \frac{\partial E_T}{\partial w_{kj}}$$

a dále z

$$\frac{\partial E}{\partial w_{ik}} = \sum_T \frac{\partial E_T}{\partial w_{ik}} \qquad \frac{\partial E}{\partial w_{kj}} = \sum_T \frac{\partial E_T}{\partial w_{kj}}$$

plyne, že adaptivní dynamika je kvazigradientním sestupem po zdola omezené globální chybové funkci, kde jeden gradientní krok odpovídá jednomu předložení trénovací množiny, takže při opakovaném předkládání trénovací množiny bude hodnota chybové funkce v závislosti na volbě velikosti míry plasticity synapse klesat.

Adaptační funkce výstupní a skryté populace určí za předpokladu jednotkových strmostí sigmoid jakožto aktivačních funkcí následující derivace:

$$\frac{\partial E_T}{\partial x_j} = y_j(1-y_j)(y_j - f_j(b_j)) \qquad \frac{dy_k}{dx_k} = y_k(1-y_k)$$

kde $y_j = F_j(\vec{a})$.

V každém adaptačním kroku tedy musíme určit stav sítě pomocí aktivní dynamiky s následujícím počátečním nastavením hodnot potenciálů neuronů vstupní populace:

$$x_i(0) = a_i(T)$$

kde $i \in V_1$.

Kroky adaptivní dynamiky jsou tedy stejně jako u kompetičního modelu podmíněny dynamikou aktivní, přičemž stavy neuronů všech jednotlivých populací se v nich z pohledu adaptivní dynamiky určí synchronně.

Jelikož pro určení hodnot vah hran spojujících vstupní a skrytou populaci neuronů v daném adaptačním kroku musejí být již určeny aktualizované hodnoty vah hran spojujících skrytou a výstupní populaci neuronů, a to na základě velikosti chyby E_T, mluvíme o *zpětném šíření* chyby proti směru orientace hran (BPA).

Funkce sítě:

Libovolnému vstupu sítě $\vec{v}$ pak funkce sítě přiřadí výstup sítě $\vec{F}(\vec{v})$ tak, že uspořádaná dvojice $[\vec{v}, \vec{F}(\vec{v})]$ představuje další funkcí sítě vytvořený a v trénovací množině neobsažený příklad nějaké objektivní vektorové funkce $\vec{G}: \mathbb{R}^n \to \mathbb{R}^m$, mluvíme tak o schopnosti sítě zobecnit (generalizovat) během adaptace získanou zkušenost, nelineární model je tak *univerzálním aproximátorem* libovolné objektivní vektorové funkce, jejíž příklady jsou uvedené v trénovací množině.

Optimální počet neuronů skryté populace je přímo úměrně závislý na míře nelinearity aproximované objektivní vektorové funkce a stanovuje se experimentálně během adaptivní dynamiky na základě vývoje hodnoty globální chybové funkce.

Za předpokladu $E(\vec{w}) = 0$ je funkce nelineárního modelu pro vzory z trénovací množiny stejně jako u lineární paměti heteroasociativní, ale na rozdíl od lineární paměti vykazuje nelineární model robustnost vůči jisté míře poškození vzoru, tj. i do jisté míry poškozený vzor asociuje s odpovídajícím obrazem, a jestliže je obraz shodný se vzorem, tj. $\vec{a}(T) = \vec{b}(T)$ pro $T \in \Delta T$, tj. $n = m$, lze nelineární model užít podobně jako lineární autoasociativní paměť k filtraci šumu. Na druhou stranu funkce nelineárního modelu není nijak určena pro lineární kombinace vzorů, pokud nejsou obsaženy v trénovací množině.

Mějme počet kategorií jistých objektů roven mohutnosti populace V_3 a mějme trénovací množinu ve tvaru $\{[\vec{a}(T), \vec{b}(T)]|T \in \Delta T\}$, kde $\vec{a} = [a_i|i \in V_1]$ představuje daný objekt a $\vec{b} = [b_j|j \in V_3]$, kde $b_j(T) = 0$ pro $j \neq k$ a $b_k(T) = 1$, představuje výrok učitele o příslušnosti daného objektu do *k*-té kategorie, pak po adaptaci sítě na uvedenou trénovací množinu funkce sítě v aktivním režimu klasifikuje libovolný objekt z hlediska daných kategorií.

Mějme časovou posloupnost $\{P(t_0 + k\Delta t)|k \in \mathbb{N}\}$, kde Δt je perioda snímkování a mějme trénovací množinu ve tvaru $\{[\vec{a}(T), \vec{b}(T)]|T \in \Delta T\}$, kde $\vec{a}(T) = [P(t_T - i\Delta t)|i \in V_1]$, $\vec{b}(T) = [P(t_T + j\Delta t)|j \in V_3]$ a $t_T = t_0 + T\Delta t$ pro $T \in \Delta T$, pak po adaptaci sítě na uvedenou trénovací množinu funkce sítě v aktivním režimu na základě předložené historie posloupnosti $[P(t - i\Delta t)|i \in V_1]$ provede predikci posloupnosti $[P(t + j\Delta t)|j \in V_3]$, kde $t = t_0 + k\Delta t$.

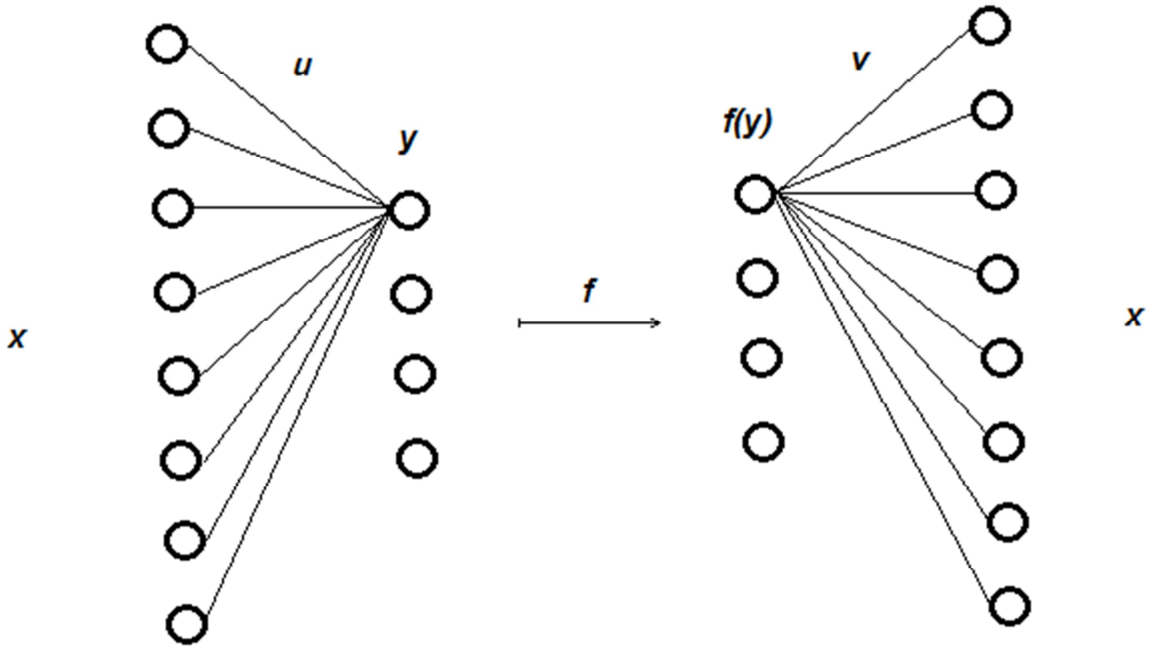
Autoasociativní funkci nelineárního modelu lze také užít k symetrickému šifrování libovolného počítačového souboru, a to během aktivní dynamiky nejprve zakódováním každého jeho bajtu $\lfloor x_1, \ldots, x_8 \rfloor$ do jemu příslušného vektoru $\lfloor y_1, \ldots, y_4 \rfloor$:

$$\lfloor x_1, \ldots, x_8 \rfloor \begin{bmatrix} u_{11} & \cdots & u_{14} \\ \vdots & \ddots & \vdots \\ u_{81} & \cdots & u_{84} \end{bmatrix} = \lfloor y_1, \ldots, y_4 \rfloor$$

a posléze jeho následným dekódováním:

$$\lfloor f_1(y_1), \ldots, f_4(y_4) \rfloor \begin{bmatrix} v_{11} & \cdots & v_{18} \\ \vdots & \ddots & \vdots \\ v_{41} & \cdots & v_{48} \end{bmatrix} = \lfloor x_1, \ldots, x_8 \rfloor$$

kde $\vec{x}$ resp. $\vec{y}$ jsou vektory potenciálů neuronů vstupní/výstupní resp. skryté, tzv. *dělící* vrstvy, $\vec{f}$ je vektor aktivačních funkcí neuronů dělící vrstvy a $[\vec{u}_1, \ldots, \vec{u}_4, \vec{v}_1, \ldots, \vec{v}_4]$ je konfigurace sítě, tj. adaptovaný náhodně generovaný klíč.



9.1. Setrvačnost gradientu chybové funkce

Definujme přírůstek konfigurace sítě během adaptačního kroku následovně:

$$\Delta\vec{w}(T) = \vec{w}(T) - \vec{w}(T-1)$$

a upravme výše uvedenou adaptivní dynamiku:

$$w_{ik}(T) = w_{ik}(T-1) - \alpha\frac{\partial E_T}{\partial w_{ik}} + \mu\,\Delta w_{ik}(T-1)$$

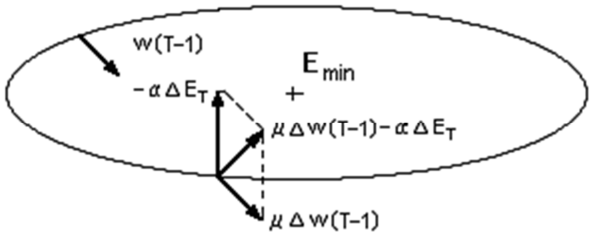
$$w_{kj}(T) = w_{kj}(T-1) - \alpha\frac{\partial E_T}{\partial w_{kj}} + \mu\,\Delta w_{kj}(T-1)$$

kde $i \in V_1$, $k \in V_2$, $j \in V_3$ a $\mu \in (0,1)$ nazveme *mírou setrvačnosti* gradientu.

Nechť během adaptačního kroku všechny parciální derivace globální chybové funkce nezmění znaménko, pak gradient chybové funkce před a po aktualizaci konfigurace významně nemění směr ani orientaci, přičtení minulého přírůstku konfigurace k aktualizovanému gradientu proběhne tedy ve směru původního gradientu a tím aktualizovaný gradient významně nevychýlí, zvětší se pouze velikost gradientního resp. adaptačního kroku ve směru k lokálnímu minimu globální chybové funkce.

Nechť během adaptačního kroku všechny parciální derivace globální chybové funkce změní znaménko, pak gradient chybové funkce před a po aktualizaci konfigurace významně nemění směr, v němž lokální minimum chybové funkce zřejmě neleží, ale mění orientaci. Přičtení minulého přírůstku konfigurace k aktualizovanému gradientu při vhodné míře setrvačnosti gradientu pak aktualizovaný gradient vychýlí směrem k lokálnímu minimu globální chybové funkce.

V obou uvedených typických situacích se tedy přičtením minulého přírůstku konfigurace k aktualizovanému gradientu konvergence hodnoty globální chybové funkce k nule urychlí, gradient globální chybové funkce díky své setrvačnosti nemění prudce směr, trajektorie gradientního sestupu směřující k hlubšímu lokálnímu minimu je přímější a mělčí lokální minima přeskočí.



9.2. Dynamika míry plasticity synapse

Uvažujme adaptivní dynamiku v *makroskopickém* čase τ, kdy konfiguraci sítě adaptujeme vždy po předložení celé trénovací množiny, na rozdíl od výše uvedené adaptivní dynamiky v *mikroskopickém* čase T, kdy jsme konfiguraci sítě adaptovali vždy po předložení příslušného prvku trénovací množiny, pak tuto adaptivní dynamiku dostaneme ve tvaru gradientního sestupu po zdola omezené globální chybové funkci, případně s dynamicky se měnící mírou plasticity synapse jakožto velikostí gradientního kroku:

$$\vec{w}(\tau+1) = \vec{w}(\tau) - \alpha(\tau)\frac{\partial E(\vec{w}(\tau))}{\partial \vec{w}(\tau)}$$

a proveďme Taylorův rozvoj *j*-té složky funkce sítě v okolí konfigurace sítě v čase τ:

$$F_j\big(\vec{w}(\tau+1)\big) \cong F_j\big(\vec{w}(\tau)\big) + \frac{\partial F_j\big(\vec{w}(\tau)\big)}{\partial \vec{w}(\tau)}\big(\vec{w}(\tau+1) - \vec{w}(\tau)\big) =$$
$$= F_j\big(\vec{w}(\tau)\big) - \alpha(\tau)\frac{\partial F_j\big(\vec{w}(\tau)\big)}{\partial \vec{w}(\tau)}\frac{\partial E\big(\vec{w}(\tau)\big)}{\partial \vec{w}(\tau)}$$

a u funkce sítě zaměňme její parametr s proměnnou, tj. konfiguraci sítě s příslušným prvkem trénovací množiny $\left(F_j\big(\vec{a}(T)\big) = F_j^{\vec{w}}\big(\vec{a}(T)\big) = F_j^{\vec{a}(T)}(\vec{w}) = F_j(\vec{w})\right)$, pak dostaneme derivaci globální chybové funkce podle míry plasticity synapse pro $e_j^T(\tau) = F_j\big(\vec{w}(\tau)\big) - f_j(b_j(T))$ ve tvaru:

$$\frac{dE(\vec{w}(\tau+1))}{d\alpha(\tau)} = \sum_T \sum_j \left(F_j\big(\vec{w}(\tau+1)\big) - f_j(b(T))\right)\frac{dF_j(\vec{w}(\tau+1))}{d\alpha(\tau)}$$

kde:

$$\frac{dF_j(\vec{w}(\tau+1))}{d\alpha(\tau)} \cong -\frac{\partial F_j\big(\vec{w}(\tau)\big)}{\partial \vec{w}(\tau)}\frac{\partial E\big(\vec{w}(\tau)\big)}{\partial \vec{w}(\tau)} \equiv -\delta_j(\tau)$$

tj.:

$$\frac{dE(\vec{w}(\tau+1))}{d\alpha(\tau)} \cong -\sum_T \sum_j \left(e_j^T(\tau) - \alpha(\tau)\delta_j(\tau)\right)\delta_j(\tau)$$

a velikost míry plasticity synapse můžeme v každém gradientním kroku optimalizovat:

$$\frac{dE(\vec{w}(\tau+1))}{d\alpha(\tau)} = 0 \quad \Rightarrow \quad \alpha(\tau) = \frac{\sum_j e_j(\tau)\,\delta_j(\tau)}{N\sum_j \delta_j^2(\tau)}$$

kde $j \in V_3$, $T \in \{1, \cdots, N\}$, $\tau \in \mathbb{N}$ a $e_j(\tau) = \sum_T e_j^T(\tau)$.

Upozornění: V dalším pro zjednodušení vyjádření příslušných parciálních derivací nebudeme uvádět makroskopický čas τ adaptivní dynamiky jakožto argument.

Skalární součin δ_j příslušných kolineárních gradientů rozepíšeme do tvaru:

$$\frac{\partial F_j(\vec{w})}{\partial \vec{w}} \frac{\partial E(\vec{w})}{\partial \vec{w}} = \sum_k \frac{\partial F_j}{\partial w_{kj}} \sum_T \frac{\partial E_T}{\partial w_{kj}} + \sum_k \sum_i \frac{\partial F_j}{\partial w_{ik}} \sum_T \frac{\partial E_T}{\partial w_{ik}}$$

tj. pro $z_j(y_j) = y_j(1 - y_j)$, $z_k(y_k) = y_k(1 - y_k)$ a $e_j(T) = y_j(T) - f_j(b_j(T))$:

$$\frac{\partial F_j}{\partial w_{kj}} = \frac{dy_j}{dx_j} \frac{\partial x_j}{\partial w_{kj}} = y_k z_j(y_j) \qquad \frac{\partial F_j}{\partial w_{ik}} = \sum_j \frac{dy_j}{dx_j} \frac{\partial x_j}{\partial y_k} \frac{\partial y_k}{\partial w_{ik}} = y_i z_k(y_k) \sum_j z_j(y_j)\, w_{kj}$$

$$\frac{\partial E_T}{\partial w_{kj}} = y_k(T)\, z_j(y_j(T))\, e_j(T) \qquad \frac{\partial E_T}{\partial w_{ik}} = y_i(T)\, z_k(y_k(T)) \sum_j z_j(y_j(T))\, e_j(T)\, w_{kj}$$

takže pro $u_{kj} = y_k z_j(y_j)$, $u_{ik} = y_i z_k(y_k)$ a $v_{kj} = z_j(y_j)\, w_{kj}$:

$$\frac{\partial F_j}{\partial w_{kj}} \sum_T \frac{\partial E_T}{\partial w_{kj}} = u_{kj} \sum_T u_{kj}(T)\, e_j(T)$$

$$\frac{\partial F_j}{\partial w_{ik}} \sum_T \frac{\partial E_T}{\partial w_{ik}} = u_{ik} \sum_j v_{kj} \sum_T u_{ik}(T) \sum_j v_{kj}(T)\, e_j(T)$$

kde $i \in V_1$, $k \in V_2$, $j \in V_3$ a $T \in \Delta T$.

10. Hybridní model (RBF)

Topologie sítě:

Rozdělme populaci neuronů sítě V na tři disjunktní populace V_1, V_2 a V_3 ($V_1 \cup V_2 \cup V_3 = V$, $V_1 \cap V_2 = \emptyset$, $V_1 \cap V_3 = \emptyset$, $V_2 \cap V_3 = \emptyset$, $|V_1| = n$, $|V_2| = M$, $|V_3| = m$) a propojme je hranami tak, aby z každého neuronu populace V_1 směřovala hrana do každého neuronu populace V_2 ($\varepsilon(E_1) = V_1 \times V_2$) a z každého neuronu populace V_2 směřovala hrana do každého neuronu populace V_3 ($\varepsilon(E_2) = V_2 \times V_3$), síť je tedy orientována směrem od populace V_1 k populaci V_3. Populaci V_1 resp. V_3 pak chápeme jako populaci vstupní resp. výstupní.

Aktivní dynamika:

Definujme aktivační funkci neuronů populace V_1 resp. V_2 resp. V_3 jako identitu resp. linearitu ($f(x) = ½\, x$) resp. sigmoidu. Nastavme počáteční hodnoty potenciálů neuronů následovně:

$$x_i(0) = v_i$$

a mějme následující aktivní dynamiku:

$$y_k(1) = f_k\left(\sum_i y_i(0)\, w_{ik} - \vartheta_k\right)$$
$$y_j(2) = f_j\left(\sum_k y_k(1)\, w_{kj}\right)$$

kde $i \in V_1$, $k \in V_2$, $j \in V_3$ a vektor $\vec{v} = [v_i | i \in V_1]$ pak chápeme jako vstup sítě.

Adaptivní dynamika:

Nastavme počáteční hodnoty vah následovně:

$$w_{ik}(0) = c_{ik} \qquad\qquad w_{kj}(0) = r_{kj}$$

Necháme-li adaptovat pouze vazby mezi neurony populace V_2 a V_3 a položíme-li elasticitu rovnu nule, pak obdržíme následující adaptivní dynamiku:

$$w_{ik}(T) = w_{ik}(T-1)$$
$$w_{kj}(T) = w_{kj}(T-1) + \alpha\, y_k(T)\, g_j(x_j(T))$$

kde $i \in V_1$, $k \in V_2$, $j \in V_3$, $T \in \Delta T$ a r_{kj} jsou náhodně generované hodnoty.

Mějme trénovací množinu ve tvaru $\{[\vec{a}(T), \vec{b}(T)]|T \in \Delta T\}$, kde $\vec{a} = [a_i|i \in V_1]$, $\vec{b} = [b_j|j \in V_3]$ a definujme globální *chybovou funkci* a adaptační funkci neuronů výstupní populace:

$$E(\vec{w}) = ½ \sum_T \sum_j (F_j(\vec{a}(T)) - f_j(b_j(T)))^2 = \sum_T E_T$$

$$g_j(x_j) = -\frac{\partial E_T}{\partial x_j} = -y_j(1 - y_j)(y_j - f_j(b_j))$$

a určeme následující parciální derivaci:

$$\frac{\partial E_T}{\partial w_{kj}} = \frac{\partial E_T}{\partial x_j}\frac{\partial x_j}{\partial w_{kj}} = -y_k\, g_j(x_j)$$

pak adaptivní dynamiku dostaneme ve tvaru:

$$w_{kj}(T) = w_{kj}(T-1) - \alpha \frac{\partial E_T}{\partial w_{kj}}$$

kde $k \in V_2$, $j \in V_3$, $T \in \Delta T$, přičemž z

$$\frac{\partial E}{\partial w_{kj}} = \sum_T \frac{\partial E_T}{\partial w_{kj}}$$

plyne, že adaptivní dynamika je kvazigradientním sestupem po zdola omezené globální chybové funkci, kde jeden gradientní krok odpovídá jednomu předložení trénovací množiny, takže při opakovaném předkládání trénovací množiny bude hodnota chybové funkce konvergovat k nule.

V každém adaptačním kroku tedy musíme určit stav sítě pomocí aktivní dynamiky s následujícím počátečním nastavením hodnot potenciálů neuronů vstupní populace:

$$x_i(0) = a_i(T)$$

kde $i \in V_1$.

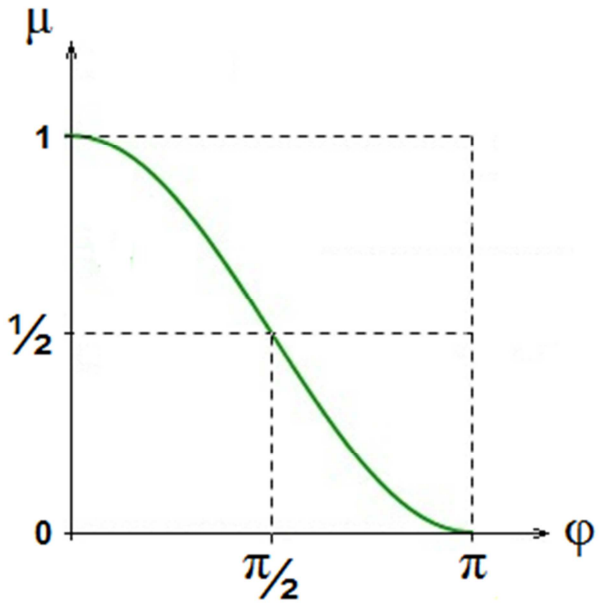
Kroky adaptivní dynamiky jsou tedy stejně jako u kompetičního modelu podmíněny dynamikou aktivní, přičemž stavy neuronů všech jednotlivých populací se v nich z pohledu adaptivní dynamiky určí synchronně.

Funkce sítě:

Nechť vektor $\vec{c}_k = [c_{ik}|i \in V_1]$ je prototyp k-tého shluku normálních n-složkových vektorů, nastavíme-li $-\vartheta_k = 1$, pak za předpokladu normality vektorů $\vec{a}$ trénovací množiny bude stav k-tého neuronu skryté populace v příslušném adaptačním kroku vyjadřovat míru vzdálenosti μ vektoru $\vec{a}(T)$ od k-tého prototypu ve smyslu neeuklidovské metriky φ:

$$y_k(T) = ½ \left(\sum_i a_i(T)\, c_{ik} + 1\right) = ½ \left(\vec{a}(T)\, \vec{c}_k + 1\right) = ½ \left(\cos \varphi(\vec{a}(T), \vec{c}_k) + 1\right)$$

kde $T \in \{1, -, N\}$ a $k \in \{1, -, M\}$ při $N \gg M$.



Libovolný normální vstup sítě $\vec{v}$ funkce sítě v prvním okamžiku aktivní dynamiky asociuje s mírami jeho vzdálenosti od všech prototypů shluků množiny $\{\vec{a}(T)|T \in \Delta T\}$ jakožto se stavy neuronů skryté populace a v druhém okamžiku aktivní dynamiky asociuje stav skryté populace s funkční hodnotou $\vec{F}(\vec{v})$.

První část sítě $\varepsilon(E_1)$ je tvořena kompetičním modelem s odebranými zápornými interními vazbami a druhá část sítě $\varepsilon(E_2)$ je tvořena horní částí nelineárního modelu, mluvíme tedy o *hybridní* síti.

Hybridní model je tak stejně jako nelineární model univerzálním aproximátorem libovolné objektivní vektorové funkce $\vec{G}: \mathbb{R}^n \to \mathbb{R}^m$, jejíž příklady jsou uvedené v trénovací množině.

11. Dodatek A: Perceptron

Topologie sítě:

Rozdělme populaci neuronů sítě V na dvě disjunktní populace V_1 a V_2 ($V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$, $|V_1| = n$, $|V_2| = 1$) a propojme je hranami tak, aby z každého neuronu populace V_1 směřovala hrana do neuronu populace V_2 ($\varepsilon(E) = V_1 \times V_2$), síť je tedy orientována směrem od populace V_1 k populaci V_2. Populaci V_1 resp. V_2 pak chápeme jako populaci vstupní resp. výstupní.

Aktivní dynamika:

Definujme aktivační funkci neuronů populace V_1 resp. V_2 jako identitu resp. ostrou nelinearitu ($f_j(0) \equiv 0$), pak pro $x_i(0) = v_i$ dostaneme následující aktivní dynamiku:

$$y_j(1) = f_j(\sum_i v_i\, w_{ij}) = f_j(\vec{v}\, \vec{w}_j)$$

kde $i \in V_1$, $j \in V_2$ a $\vec{w}_j = \left[w_{ij} | i \in V_1\right]$. Vektor $\vec{v} = [v_i | i \in V_1]$ pak chápeme jako vstup sítě.

Adaptivní dynamika:

Zvolme $w_{ij}(0) = 0$ a položme plasticitu resp. elasticitu rovnu jedné resp. nule, pak dostaneme následující adaptivní dynamiku:

$$w_{ij}(T) = w_{ij}(T-1) + x_i(T)\, g_j(x_j(T))$$

kde $i \in V_1$, $j \in V_2$ a $T \in \Delta T$.

Mějme trénovací množinu ve tvaru $\{[\vec{a}(T), b(T)] | T \in \Delta T\}$, kde $\vec{a} = [a_i | i \in V_1]$ a $b \in \{0,1\}$ uspořádanou tak, aby $b(1) = 1$ a definujme *chybovou funkci*:

$$E(T) = b(T) - f_j(\vec{a}(T)\, \vec{w}_j(T-1)) \qquad (E(1) = b(1) = 1)$$

a dále zvolme adaptační funkci výstupního neuronu ve tvaru $g_j(x_j(T)) = E(T)$, pak:

$$\vec{w}_j(T) = \vec{w}_j(T-1) + \vec{a}(T)E(T) \qquad (\vec{w}_j(1) = \vec{a}(1))$$

tj. pro $\Delta T = \{1, \cdots, N\}$:

$$\vec{w}_j(N) = \sum_T \vec{a}(T)E(T)$$

kde $j \in V_2$ a $T \in \Delta T$.

Funkce sítě:

Zvolme $\vec{a}(T) \in \alpha \Rightarrow b(T) = 1$ a $\vec{a}(T) \in \beta \Rightarrow b(T) = 0$ (viz definice níže) a nechť vektory $\vec{a}$ jsou vzájemně ortonormální a vektory $\vec{w}_j$ normální, pak platí:

$$f_j(\vec{a}(T)\,\vec{w}_j(T)) = b(T) \qquad (f_j(\vec{a}(1)\,\vec{w}_j(1)) = f_j(\vec{a}(1)\,\vec{a}(1)) = f_j(1) = b(1) = 1)$$

neboť z Cauchy - Shwarzovy nerovnosti $\left|\vec{a}(T)\,\vec{w}_j(T-1)\right| < |\vec{a}(T)|\,\left|\vec{w}_j(T-1)\right| = 1$ plyne pro $T \neq 1$ a $E(T) \neq 0$:

$$f_j(\vec{a}(T)(\vec{w}_j(T-1) + \vec{a}(T)E(T))) = f_j(\vec{a}(T)\,\vec{w}_j(T-1) + E(T)) = f_j(E(T)) = b(T)$$

Poznámka: $f_j(E(T)) = b(T)$ plyne z následujících implikací:
$E(T) = -1 \Rightarrow b(T) = 0 \wedge f_j(\vec{a}(T)\,\vec{w}_j(T-1)) = 1 \Rightarrow \vec{a}(T)\,\vec{w}_j(T-1) > 0 \Rightarrow \vec{a}(T) \in \alpha$
$E(T) = +1 \Rightarrow b(T) = 1 \wedge f_j(\vec{a}(T)\,\vec{w}_j(T-1)) = 0 \Rightarrow \vec{a}(T)\,\vec{w}_j(T-1) < 0 \Rightarrow \vec{a}(T) \in \beta$

Definice: $\alpha, \beta \subset \mathbb{R}^n$, $\alpha \cap \beta = \emptyset$
α, β – lineárně separabilní nadrovinou $\rho \Leftrightarrow \exists \vec{w} \in \mathbb{R}^n\ (\vec{v} \in \alpha \Rightarrow \vec{v}\,\vec{w} > 0) \wedge (\vec{v} \in \beta \Rightarrow \vec{v}\,\vec{w} < 0)$
kde $\rho = \{\vec{v} \in \mathbb{R}^n | \vec{v}\,\vec{w} = 0\}$ a $\alpha \cup \beta \cup \rho = \mathbb{R}^n$

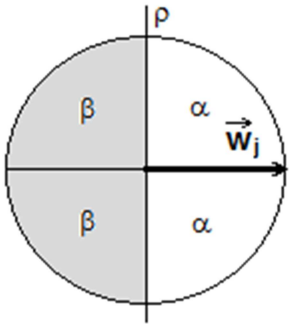
Z výše uvedeného pro $\tau = \{1, \cdots, N\}$ platí:

$$F_j\left(\vec{a}(\tau)\right) = f_j(\vec{a}(\tau) \sum_T \vec{a}(T)E(T)) = f_j(E(\tau)) = b(\tau)$$

a pro $\vec{v} \in \mathbb{R}^n$, kde $|\vec{v}| = 1$, platí:

$$f_j\left(\vec{v}\,\vec{w}_j\right) = f_j(|\vec{v}|\left|\vec{w}_j\right|\cos\varphi) = f_j(\cos\varphi)$$

kde $j \in V_2$, $T \in \Delta T$, tj. perceptron klasifikuje libovolný normální vektor z hlediska dvou lineárně separabilních kategorií α a β, přičemž vektor ležící v *dělící* nadrovině ρ přiřadí do β.



12. Dodatek B: Vícevrstvý perceptron (zobecnění BPA)

Mějme lokální chybu prvku trénovací množiny nelineárního modelu vyjádřenou součtem rozdílů skutečné a požadované hodnoty stavu neuronů výstupní vrstvy sítě V_3 a vyjádřeme si stavy a potenciály neuronů skryté vrstvy sítě V_2:

$$E_T = ½ \sum_j (y_j(T) - z_j(T))^2 \qquad y_k = (1 + e^{-p_k x_k})^{-1} \qquad x_k = \sum_i y_i \, w_{ik} + \vartheta_k$$

Rozšiřme seznam nezávisle proměnných chybové funkce nelineárního modelu, dosud tvořený synaptickými vahami vazeb mezi neurony jednotlivých vrstev sítě, tj. w_{ik} a w_{kj}, o parametry aktivační funkce neuronů skryté vrstvy sítě, tj. ϑ_k a p_k, a určeme následující derivace:

$$\frac{\partial E_T}{\partial w_{kj}} = \frac{\partial E_T}{\partial x_j}\frac{\partial x_j}{\partial w_{kj}} = y_k \, p_j y_j (1 - y_j)(y_j - z_j) \; = -y_k \, g_j(x_j)$$

$$\frac{\partial E_T}{\partial w_{ik}} = \sum_j \frac{\partial E_T}{\partial x_j}\frac{\partial x_j}{\partial y_k}\frac{dy_k}{dx_k}\frac{\partial x_k}{\partial w_{ik}} = y_i \, p_k y_k (1 - y_k) \sum_j p_j y_j (1 - y_j)(y_j - z_j) \, w_{kj} = -y_i \, g_k(x_k)$$

$$\frac{\partial E_T}{\partial \vartheta_k} = \sum_j \frac{\partial E_T}{\partial x_j}\frac{\partial x_j}{\partial y_k}\frac{dy_k}{dx_k}\frac{\partial x_k}{\partial \vartheta_k} = p_k y_k (1 - y_k) \sum_j p_j y_j (1 - y_j)(y_j - z_j) \, w_{kj} = -g_k(x_k)$$

$$\frac{\partial E_T}{\partial p_k} = \sum_j \frac{\partial E_T}{\partial x_j}\frac{\partial x_j}{\partial y_k}\frac{dy_k}{dp_k} = x_k \, y_k (1 - y_k) \sum_j p_j y_j (1 - y_j)(y_j - z_j) \, w_{kj} = -\frac{x_k}{p_k} g_k(x_k)$$

kde:

$$\frac{\partial E_T}{\partial x_j} = \frac{dy_j}{dx_j}(y_j - z_j) \qquad \frac{\partial x_j}{\partial y_k} = w_{kj} \qquad \frac{\partial x_j}{\partial w_{kj}} = y_k \qquad \frac{\partial x_k}{\partial w_{ik}} = y_i \qquad \frac{\partial x_k}{\partial \vartheta_k} = 1$$

$$\frac{dy_j}{dx_j} = p_j \, y_j(1 - y_j) \qquad \frac{dy_k}{dx_k} = p_k \, y_k(1 - y_k) \qquad \frac{dy_k}{dp_k} = x_k \, y_k(1 - y_k)$$

pak dostaneme adaptivní dynamiku nelineárního modelu zobecněnou o adaptaci prahů a strmostí aktivačních funkcí neuronů skryté vrstvy sítě jakožto nezávisle proměnných chybové funkce nelineárního modelu:

$$w_{ik}(T) = w_{ik}(T-1) - \alpha \frac{\partial E_T}{\partial w_{ik}}$$

$$w_{kj}(T) = w_{kj}(T-1) - \alpha \frac{\partial E_T}{\partial w_{kj}}$$

$$\vartheta_k(T) = \vartheta_k(T-1) - \alpha \frac{\partial E_T}{\partial \vartheta_k}$$

$$p_k(T) = p_k(T-1) - \alpha \frac{\partial E_T}{\partial p_k}$$

kde $i \in V_1$, $k \in V_2$, $j \in V_3$ a $T \in \Delta T$.

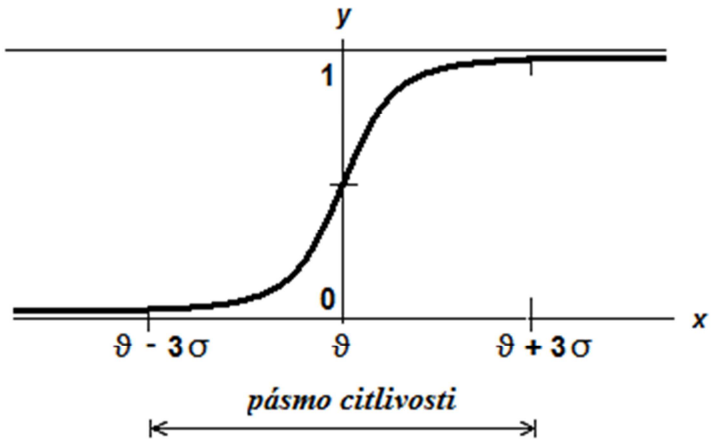
Prahy a strmosti aktivačních funkcí (sigmoidální funkce resp. Gaussova funkce resp. funkce „mexický klobouk“) neuronů vstupní a výstupní vrstvy sítě můžeme určit v závislosti na trénovací množině pomocí dále uvedených transformací:

sigmoidální transformace: $f(x) = \left(1 + e^{-p(x-\vartheta)}\right)^{-1}$

$$0{,}95 = \left(1 + e^{-p(\vartheta+3\sigma-\vartheta)}\right)^{-1} \quad \Rightarrow \quad e^{-3p\sigma} = \frac{0{,}05}{0{,}95}$$

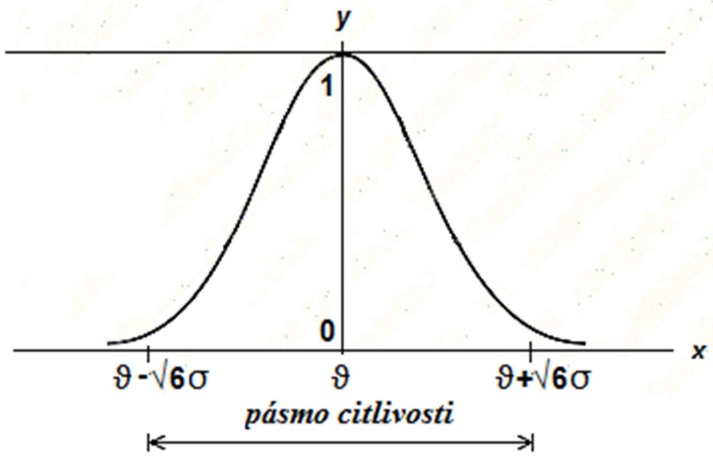
$$0{,}05 = \left(1 + e^{-p(\vartheta-3\sigma-\vartheta)}\right)^{-1} \quad \Rightarrow \quad e^{3p\sigma} = \frac{0{,}95}{0{,}05}$$

$$\Rightarrow \quad p = \frac{\ln 0{,}95 - \ln 0{,}05}{3\sigma} \cong \frac{1}{\sigma}$$



radiální transformace I: $f(x) = e^{-p(x-\vartheta)^2}$

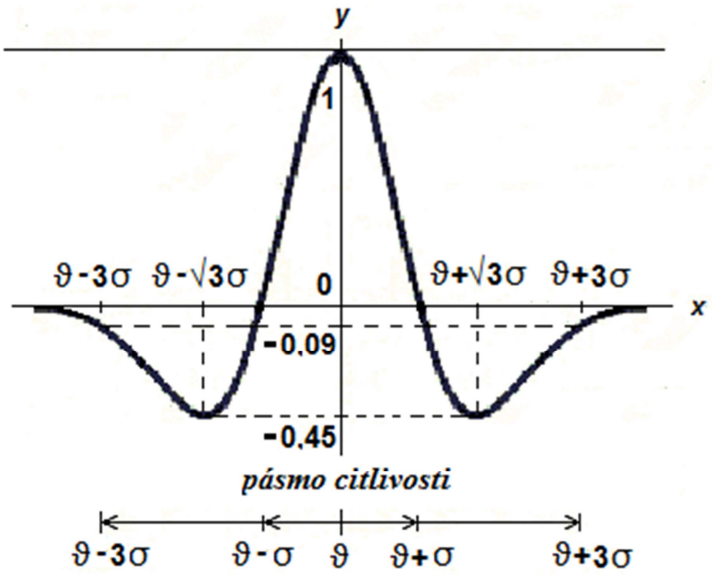
$$0{,}05 = e^{-p\left(\vartheta \pm \sqrt{6}\sigma - \vartheta\right)^2} \quad \Rightarrow \quad p = -\frac{1}{6\sigma^2}\ln 0{,}05 \cong \frac{1}{2\sigma^2}$$



radiální transformace II: $g(x) = -\sigma^2 f''(x)$

$$f(x) = e^{-p(x-\vartheta)^2} \Rightarrow f'(x) = -2p(x-\vartheta)f(x) \Rightarrow f''(x) = -2p(1-2p(x-\vartheta)^2)f(x)$$

$$g(x) = -\sigma^2 f''(x) = \left(1 - \frac{1}{\sigma^2}(x-\vartheta)^2\right) f(x) \Rightarrow g'(x) = \frac{1}{\sigma^2}(x-\vartheta)\left(\frac{1}{\sigma^2}(x-\vartheta)^2 - 3\right) f(x)$$

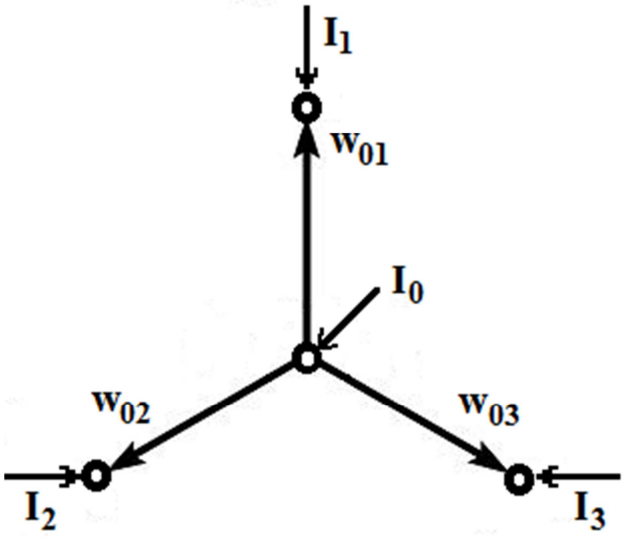


pozn.: Výše uvedené transformaci resp. její nezáporné části odpovídají různá pásma citlivosti.

ϑ – střední hodnota dat přiváděných na daný neuron z trénovací množiny
σ – směrodatná odchylka dat přiváděných na daný neuron z trénovací množiny

13. Dodatek C: Instar učení ve spojitém čase

Mějme následující neuronovou síť:



kde pro zjednodušení uvažujme aktivační a adaptační funkci jako identitu a míru plasticity nastavme shodně s mírou elasticity na hodnotu jedna, pak za předpokladu nulových počátečních podmínek pro $-\vartheta_j = I_j > 0$, kde $j \in \{0,1,2,3\}$, dostaneme následující aktivní a adaptivní dynamiku:

Aktivní dynamika:

$$\frac{d}{dt}x_0(t) + x_0(t) = I_0$$

tj.

$$x_0(t) = I_0\,(1 - e^{-t})$$

tj.

$$\lim_{t\to\infty} x_0(t) = x_0(\infty) = I_0$$

a pro $j \in \{1,2,3\}$

$$\frac{d}{dt}x_j(t) + x_j(t) = x_0(t - \Delta t)\,w_{0j} + I_j$$

tj.

$$x_j(t) = I_0\,w_{0j}\,(1 - q(t)\,e^{-t}) + I_j\,(1 - e^{-t}) \qquad q(t) = (1 + t\,e^{\Delta t})$$

tj.

$$\lim_{t\to\infty} x_j(t) = x_j(\infty) = I_0\,w_{0j} + I_j$$

Adaptivní dynamika:

$$\frac{d}{dT} w_{0j}(T) + x_j(\infty)\, w_{0j}(T) = x_0(\infty)\, x_j(\infty)$$

tj.

$$\frac{d}{dT} w_{0j}(T) = -(a\, w_{0j}^2(T) - b\, w_{0j}(T) + c)$$

pro $j \in \{1,2,3\}$, kde

$$a = I_0 \qquad b = I_0^2 - I_j \qquad c = -I_0\, I_j$$

tj. po separaci proměnných:

$$T = -\int \frac{1}{a\, w_{0j}^2 - b\, w_{0j} + c}\, dw_{0j}$$

tj. obecně:

$$T = -\frac{1}{\sqrt{D}} \ln \left| \frac{w_{0j} - \frac{b+\sqrt{D}}{2a}}{w_{0j} - \frac{b-\sqrt{D}}{2a}} \right| + C \qquad D = b^2 - 4ac = \left(I_0^2 + I_j\right)^2 > 0$$

přičemž platí $b - \sqrt{D} < 0 < b + \sqrt{D}$ a kde C je integrační konstanta.

Za prvé uvažujme $w_{0j}(T) \notin \langle \frac{b-\sqrt{D}}{2a}, \frac{b+\sqrt{D}}{2a} \rangle$, pak dostaneme hyperbolický průběh:

$$e^{-\sqrt{D}(T-C)} = \frac{w_{0j}(T) - \frac{b+\sqrt{D}}{2a}}{w_{0j}(T) - \frac{b-\sqrt{D}}{2a}}$$

tj. pro $T \neq C$:

$$w_{0j}(T) = \frac{1}{2a}\left(b - \sqrt{D}\, \frac{e^{-\sqrt{D}(T-C)} + 1}{e^{-\sqrt{D}(T-C)} - 1} \right)$$

tj.

$$\lim_{T\to\infty} w_{0j}(T) = w_{0j}(\infty) = \frac{b+\sqrt{D}}{2a} = I_0$$

a

$$\lim_{T\to-\infty} w_{0j}(T) = w_{0j}(-\infty) = \frac{b-\sqrt{D}}{2a} = -\frac{I_j}{I_0}$$

Pro $T = 0$ dále dostaneme:

$$C = \frac{1}{\sqrt{D}} \ln \frac{w_{0j}(0) - \frac{b+\sqrt{D}}{2a}}{w_{0j}(0) - \frac{b-\sqrt{D}}{2a}}$$

takže pro $w_{0j}(0) > \frac{b+\sqrt{D}}{2a}$ platí $C < 0$ a pro $w_{0j}(0) < \frac{b-\sqrt{D}}{2a}$ platí $C > 0$, z čehož plyne, že z pohledu předpokladu nulových počátečních podmínek je hodnota $w_{0j}(0)$ sice vždy dobře definovaná, nicméně leží vždy mimo interval $\langle \frac{b-\sqrt{D}}{2a}, \frac{b+\sqrt{D}}{2a} \rangle$, a proto nemůže nikdy nabýt nulové hodnoty, tj. pro toto řešení adaptivní dynamiky nelze splnit požadavek nulových počátečních podmínek.

Za druhé uvažujme $w_{0j}(T) \in (\frac{b-\sqrt{D}}{2a}, \frac{b+\sqrt{D}}{2a})$, pak dostaneme sigmoidální průběh:

$$e^{-\sqrt{D}(T-C)} = \frac{\frac{b+\sqrt{D}}{2a} - w_{0j}(T)}{w_{0j}(T) - \frac{b-\sqrt{D}}{2a}}$$

tj.

$$w_{0j}(T) = \frac{1}{2a}\left(b - \sqrt{D}\frac{e^{-\sqrt{D}(T-C)} - 1}{e^{-\sqrt{D}(T-C)} + 1}\right)$$

tj.

$$\lim_{T\to\infty} w_{0j}(T) = w_{0j}(\infty) = \frac{b+\sqrt{D}}{2a} = I_0$$

a

$$\lim_{T\to-\infty} w_{0j}(T) = w_{0j}(-\infty) = \frac{b-\sqrt{D}}{2a} = -\frac{I_j}{I_0}$$

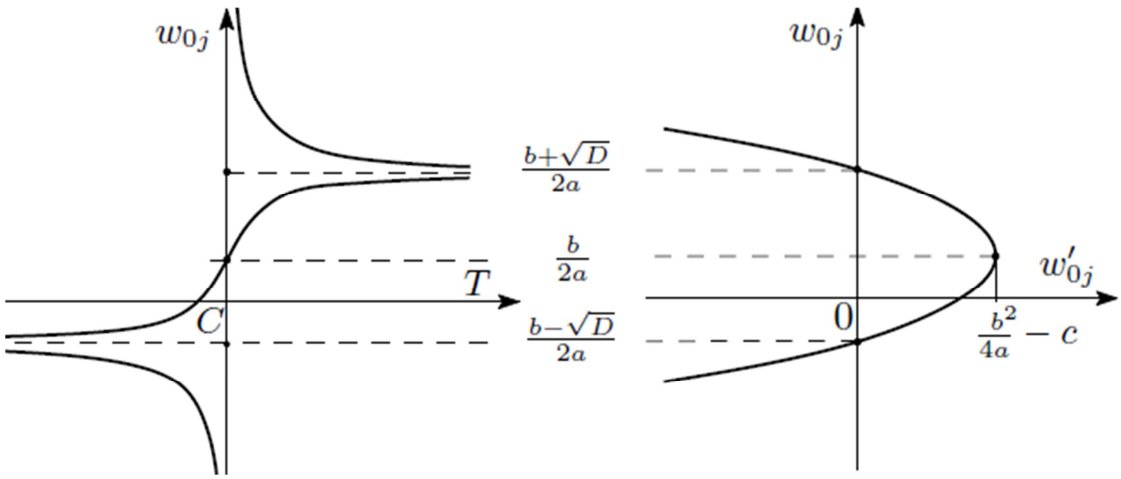
Pro $T = 0$ dále dostaneme:

$$C = \frac{1}{\sqrt{D}} \ln \frac{\frac{b+\sqrt{D}}{2a} - w_{0j}(0)}{w_{0j}(0) - \frac{b-\sqrt{D}}{2a}}$$

tj. pro nulové počáteční podmínky je:

$$C = \frac{1}{\sqrt{D}} \ln \frac{\sqrt{D}+b}{\sqrt{D}-b}$$

tj. jedině v případě $I_0^2 = I_j$ je $C = 0$



Z obecného řešení adaptivní dynamiky plyne, že pokud budou synaptické váhy naučené sítě před zahájením její adaptace již nastaveny na požadované hodnoty, tj. na limitní hodnoty v nekonečném čase adaptivní dynamiky, adaptivní dynamika nemá smysl, tj. naučenou síť již nemá smysl učit.

V opačném případě funkci naučené sítě při obráceném toku času, tj. síť si pamatuje budoucnost, dostaneme ve tvaru:

$$\vec{F}([I_0, 0,0,0]) = [I_0, -I_1, -I_2, -I_3]$$

a při běžném toku času, tj. síť si pamatuje minulost, dostaneme ve tvaru:

$$\vec{F}([I_0, 0,0,0]) = [I_0, I_0^2, I_0^2, I_0^2]$$

tj. funkce sítě je tedy závislá na orientaci percepční šipky času.

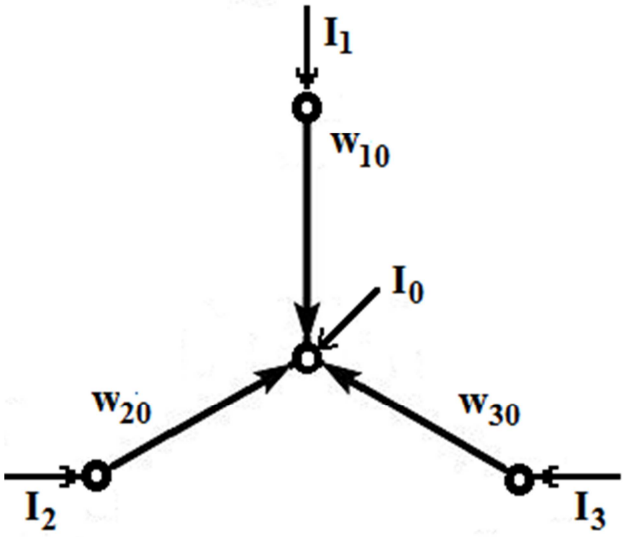
Z partikulárního řešení adaptivní dynamiky daného nulovými počátečními podmínkami při nulové integrační konstantě a běžném toku času dostaneme funkci naučené sítě v následujícím tvaru:

$$\vec{F}([I_0, 0,0,0]) = [I_0, I_1, I_2, I_3]$$

Během *instar učení* ($\beta = 1$) se do paměti synaptické vazby reálně uloží po dostatečně dlouhou dobu jí procházející informační signál I_0 (presynaptická aktivita), tj. kapacita paměti je omezená hodnotou I_0.

14. Dodatek D: Outstar učení ve spojitém čase

Mějme následující neuronovou síť:



kde pro zjednodušení uvažujme aktivační a adaptační funkci jako identitu a míru plasticity resp. elasticity nastavme na hodnotu jedna resp. nula, pak za předpokladu nulových počátečních podmínek pro $-\vartheta_j = I_j > 0$, kde $j \in \{0,1,2,3\}$, dostaneme následující aktivní a adaptivní dynamiku:

Aktivní dynamika:

$$\frac{d}{dt}x_0(t) + x_0(t) = \sum_j x_j(t - \Delta t)\, w_{j0} + I_0$$

tj.

$$x_0(t) = \sum_j I_j\, w_{j0}\ (1 - q(t)\, e^{-t}) + I_0\, (1 - e^{-t}) \qquad q(t) = (1 + t\, e^{\Delta t})$$

tj.

$$\lim_{t\to\infty} x_0(t) = x_0(\infty) = \sum_j I_j\, w_{j0} + I_0$$

a pro j∈{1,2,3}:

$$\frac{d}{dt}x_j(t) + x_j(t) = I_j$$

tj.

$$x_j(t) = I_j\, (1 - e^{-t})$$

tj.

$$\lim_{t\to\infty} x_j(t) = x_j(\infty) = I_j$$

Adaptivní dynamika:

$$\frac{d}{dT} w_{j0}(T) = x_j(\infty)\, x_0(\infty)$$

tj.

$$\frac{d}{dT} w_{j0}(T) = I_j(\sum_i I_i w_{i0}(T) + I_0)$$

pro $i, j \in \{1,2,3\}$, tj. obecně:

$$w_{j0}(T) = w_{j0}(0) + \frac{I_j}{\omega}\, C\, (e^{\omega T} - 1)$$

kde $\omega = \left|\vec{I}\right|^2$ pro $\vec{I} = [I_1, I_2, I_3]$ a kde integrační konstanta $C = \sum_i I_i w_{i0}(0) + I_0$, pak platí:

$$\lim_{T \to \infty} w_{j0}(T) = w_{0j}(\infty) = \infty$$

z čehož plyne, že nemá smysl nechat čas adaptivní dynamiky běžet do nekonečna. Nechme na síť působit stimul I_0 a $\vec{I}$ po dobu adaptace (T_0, T_1) a vzápětí stimul J_0 a $\vec{J} = [J_1, J_2, J_3]$ po dobu adaptace (T_1, T_2), kde:

$$T_0 = 0 \qquad T_1 = \left|\vec{I}\right|^{-2} \ln 2 \qquad T_2 = \left|\vec{J}\right|^{-2} \ln 2$$

Z partikulárního řešení adaptivní dynamiky daného nulovými počátečními podmínkami pro dobu adaptace (T_0, T_1) dostaneme funkci naučené sítě ve tvaru:

$$\vec{F}([0, I_1, I_2, I_3]) = [I_0, I_1, I_2, I_3]$$

neboť:

$$\sum_j I_j\, w_{j0}\,(T_1) = I_0$$

a dále z partikulárního řešení adaptivní dynamiky daného počátečními podmínkami $w_{j0}(T_1) = \left|\vec{I}\right|^{-2} I_j\, I_0$ pro dobu adaptace (T_1, T_2) dostaneme:

$$w_{j0}(T_2) = \frac{I_j\, I_0}{\left|\vec{I}\right|^2} + \frac{J_j\, J_0}{\left|\vec{J}\right|^2} + \frac{(\vec{I} \cdot \vec{J})}{\left|\vec{I}\right|^2 \left|\vec{J}\right|^2}\, J_j\, I_0$$

tj.

$$\sum_j I_j\, w_{j0}\,(T_2) = I_0 + \frac{(\vec{I}\cdot\vec{J})}{|\vec{J}|^2} J_0 + \frac{(\vec{I}\cdot\vec{J})}{|\vec{I}|^2 |\vec{J}|^2}\,(\vec{I}\cdot\vec{J})\, I_0$$

$$\sum_j J_j\, w_{j0}\,(T_2) = J_0 + \frac{(\vec{I}\cdot\vec{J})}{|\vec{I}|^2} I_0 + \frac{(\vec{I}\cdot\vec{J})}{|\vec{I}|^2 |\vec{J}|^2}\,(\vec{J}\cdot\vec{J})\, I_0$$

tj.

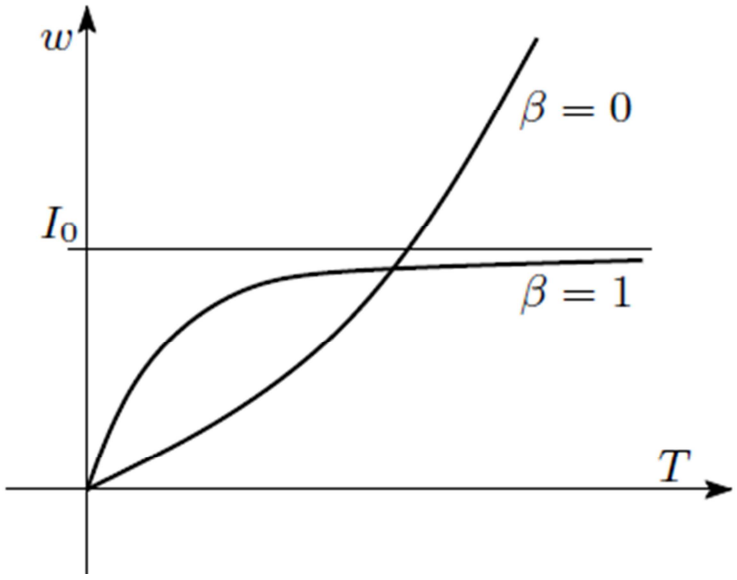
$$\sum_j I_j\, w_{j0}\,(T_2) = I_0 + \frac{|\vec{I}|}{|\vec{J}|} J_0\, cos\varphi(\vec{I},\vec{J}) + I_0\, cos^2\varphi(\vec{I},\vec{J})$$

$$\sum_j J_j\, w_{j0}\,(T_2) = J_0 + \frac{|\vec{J}|}{|\vec{I}|}\, I_0\, cos\varphi(\vec{I},\vec{J}) + \frac{|\vec{J}|}{|\vec{I}|}\, I_0\, cos\varphi(\vec{I},\vec{J})$$

tj. pro vzájemně ortogonální vstupy $\vec{I},\vec{J}$ dostaneme funkci naučené sítě ve tvaru:

$$\vec{F}([0,I_1,I_2,I_3]) = [I_0,I_1,I_2,I_3] \qquad \vec{F}([0,J_1,J_2,J_3]) = [J_0,J_1,J_2,J_3]$$

Během *outstar učení* ($\beta = 0$) se do paměti synaptické vazby uloží po určitou dobu jí procházející normovaný informační signál I_j, vážený stimulem I_0 (postsynaptická aktivita), injektovaným do neuronu přijímacího procházející signál. Analogicky můžeme neomezeně načítat do paměti hodnoty dané páry stimulů $\vec{J}$ a J_0 injektovaných do obou vrstev neuronů, tj. do cípů resp. středu hvězdy, tj. kapacita paměti je neomezená.



15. Dodatek E: Simulované žíhání

Evoluční algoritmy se užívají k nalezení dostatečně kvalitního řešení optimalizačních úloh v dostatečně krátkém čase. Mezi evoluční algoritmy inspirované přírodou se zahrnuje celé spektrum optimalizačních heuristických technik, např. genetické algoritmy či simulované žíhání. Heuristiky můžeme popsat jako zkratkovitý postup prohledávání prostoru řešení bez záruky správného výsledku, nicméně jsou zbaveny celé řady neduhů konvenčních optimalizačních metod, jako např. požadavek spojitosti či diferencovatelnosti objektivní resp. vazební funkce, respektování omezujících podmínek, uvíznutí v mělkém lokálním minimu atd. Na druhou stranu však je při jejich aplikaci zapotřebí nastavení jistých volných parametrů, které je nutné „naladit" v závislosti na konkrétním optimalizačním problému, tak jako např. počáteční resp. konečnou teplotu a počet iterací dále popsaného algoritmu simulovaného žíhání, vycházejícího z evoluce termodynamických systémů. Žíháním označujeme ve fyzice proces, při kterém je těleso zahřáté na vysokou teplotu postupně ochlazováno, čímž se odstraňují vnitřní defekty tělesa. Vlivem vysoké teploty se částice látky v tělese náhodně uspořádají, tím se defekty krystalické mřížky zahladí a postupným ochlazováním pak částice ustalujeme do rovnovážných poloh spolu s poklesem pravděpodobnosti vzniku defektů nových.

Představme si, že argument optimalizované funkce jednoznačně určuje makrostav nějakého termodynamického systému o energii rovné funkční hodnotě, pak můžeme vyjádřit jeho termodynamickou pravděpodobnost:

$$P(E_i) = \left|\{\vec{x} \in \mathbb{R}^n \,|\, f(\vec{x}) = E_i\}\right|$$

jako počet jemu odpovídajících mikrostavů.

Ponoříme-li uvedený systém nabývající různých makrostavů o energiích E_i do tepelné lázně o energii E_0, pak dle Boltzmannovy rovnice, pro jednotkovou velikost Boltzmannovy konstanty, pomocí Taylorova rozvoje diferencovatelné funkce můžeme po vyrovnání teplot vyjádřit pro $E = E_0 + E_i = konst$ a $E >> E_i$ entropii lázně:

$$S(E_i) = S(E) - \frac{dS(E)}{dE_i} E_i = \ln P(E - E_i)$$

a dále užitím definice teploty $dS(E)/dE = (1/T)$ pro $T > 0$ vyjádříme termodynamickou pravděpodobnost makrostavu tepelné lázně jako funkci energie makrostavu vloženého systému, tj. pomocí Boltzmannova faktoru:

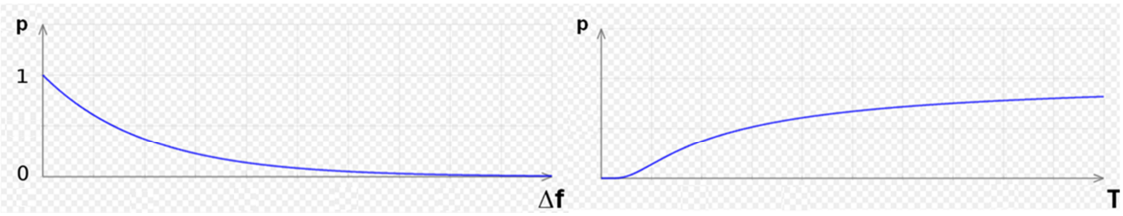
$$P(E - E_i) = ce^{-\frac{E_i}{T}}$$

Algoritmus simulovaného žíhání spočívá v perturbaci kandidáta na optimum a následném rozhodnutí o jeho nahrazení perturbací v každé iteraci algoritmu dle Metropolisova kritéria:

$$p(\vec{x}_i \to \vec{x}_j) = \frac{P(E_j)}{P(E_i)} = e^{-\frac{\Delta E}{T}} \qquad \Delta E > 0$$
$$p(\vec{x}_i \to \vec{x}_j) = 1 \qquad \Delta E \leq 0$$

vyjadřujícího pravděpodobnost přechodu systému z jednoho makrostavu do druhého, kde $\Delta E = E_j - E_i$ a $\Delta E/T$ vyjadřuje přírůstek entropie, tj. v souladu s druhou větou termodynamickou nemožný jev je v uvedeném kritériu uměle předefinován na jev jistý.

Posloupnost akceptovaných perturbací, tj. přípustných řešení optimalizační úlohy, tvoří Markovův řetězec s pamětí řádu jedna, tj. výskyt daného řešení je podmíněn pouze výskytem řešení předcházejícího. Perturbace ležící mimo oblast přípustných řešení se zamítají automaticky.



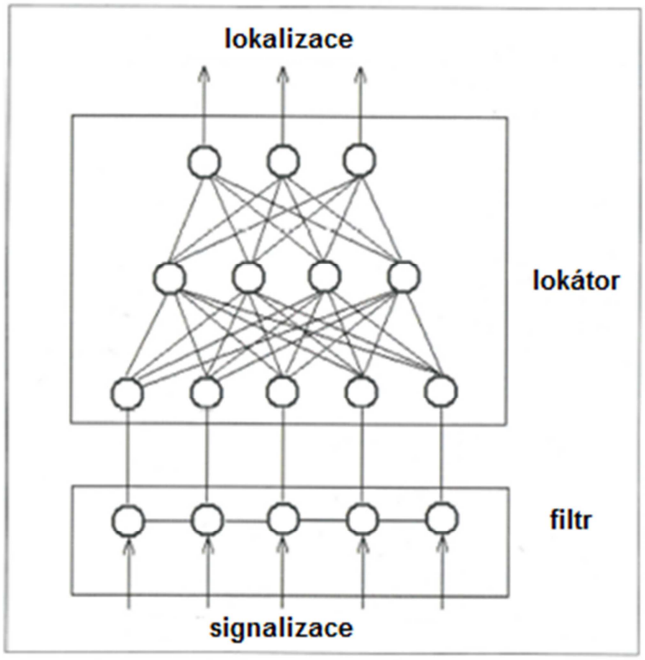
Ze závislosti $p(\Delta f)$ je zřejmé, že výrazně „horší“ řešení se akceptuje vůči předcházejícímu řešení s mnohem menší pravděpodobností než řešení jen o málo „horší“. Závislost $p(T)$ lze užít k řízení pravděpodobnosti akceptace řešení během iteračního cyklu. Iterační cyklus startujeme s tak vysokou teplotou, aby se po jistou dobu akceptovalo téměř každé navržené řešení, což případně umožní počáteční aproximaci řešení „vyklouznout“ z oblasti mělkých lokálních minim, ke konci iteračního cyklu naopak teplotu dostatečně snížíme tak, aby se neakceptovalo téměř žádné „horší“ řešení, tj. během iteračního cyklu chladíme systém představující optimalizační úlohu z dostatečně vysoké teploty na dostatečně nízkou teplotu tak, že nám v závěru cyklu řešení „zamrzne“ v dostatečně hlubokém lokálním minimu. Pokles teploty může být zvolen např. jako exponenciální:

$$T = T_0 e^{-\frac{iter}{\tau}} \qquad \tau = -\frac{N}{\ln(T_\infty / T_0)} \qquad T_\infty \approx \lim_{iter \to \infty} T_0 e^{-\frac{iter}{\tau}} = 0$$

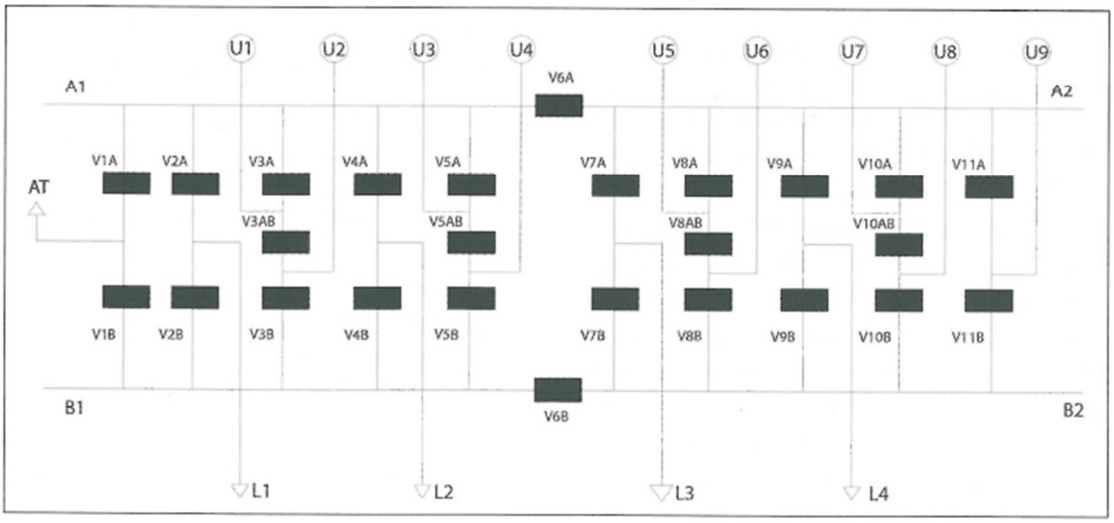
kde T_0 resp. T_∞ je počáteční resp. konečná teplota a N je počet iterací algoritmu.

16. Příklad aplikace lineárního modelu

Příklad popisuje aplikaci předřazení lineárního modelu jakožto projekční vrstvy (filtr) modelu nelineárnímu (lokátor), jakožto asociační vrstvě, a to při lokalizaci poruchy (zkratu) v rozvodně elektrické energie z důvodu filtrace možného šumu obsaženého ve snímku signalizací stavů vypínačů a ochran, na základě kterého pak probíhá lokalizace (asociace) místa poruchy v rozvodně. Snímek signalizací je zpracováván dvoufázově: nejprve proběhne jeho projekce do okruhu známých vzorů (filtrace šumu), která je pak asociována s místem poruchy (lokalizace poruchy):



Rozvodna se skládá ze čtyř sběrnic (A1, A2, B1, B2), čtyř vývodů vedení (L1-L4) a deseti vývodů strojů (U1-U9, AT), tj. devíti generátorů a autotransformátoru, a dále z dvaceti šesti vypínačů a čtyřiceti ochran:



Lokátor je tvořen vstupní vrstvou o 66 neuronech odpovídajících stavům jednotlivých vypínačů a ochran a výstupní vrstvou o 18 neuronech odpovídajících jednotlivým sběrnicím a vývodům. Mezi vstupní a výstupní vrstvu byly vloženy dvě skryté vrstvy o 45 a 25 neuronech. Požadovaná funkce lokátoru o 5% chybovosti byla dosažena po přibližně 6800 předložených tréninkových dat s parametry učení 0,05 (míra plasticity synapse) a 0,7 (míra setrvačnosti gradientu).

Tréninková množina lokátoru je uvedena v následující tabulce signalizací stavů vypínačů (levá část) a ochran (pravá část) v závislosti na místě poruchy v rozvodně, kde hodnota 1 signalizuje vypnutí vypínače resp. působení ochrany:

	A1	A2	B1	B2	L1	L2	L3	L4	U1	U2	U3	U4	U5	U6	U7	U8	U9	AT
V1A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
V2A	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
V3A	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
V4A	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
V5A	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
V6A	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V3AD	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
V5AB	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
V1B	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
V2B	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
V3B	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
V4B	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
V5B	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
V6B	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V7A	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
V8A	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
V9A	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
V10A	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
V11A	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
V8AB	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
V10AB	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0
V7B	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
V8B	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
V9B	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
V10B	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
V11B	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0

	A1	A2	B1	B2	L1	L2	L3	L4	U1	U2	U3	U4	U5	U6	U7	U8	U9	AT
RA1/H	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RA1/Z	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RB1/H	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RB1/Z	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RA2/H	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RA2/Z	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RB2/H	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RB2/Z	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ASV/A1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ASV/A2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ASV/B1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ASV/B2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DL1/H	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
DL1/Z	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
DL2/H	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
DL2/Z	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
DL3/H	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
DL3/Z	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
DL4/H	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
DL4/Z	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
RU1/H	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
RU1/Z	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
RU2/H	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
RU2/Z	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
RU3/H	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
RU3/Z	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
RU4/H	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
RU4/Z	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
RU5/H	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
RU5/Z	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
RU6/H	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
RU6/Z	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
RU7/H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
RU7/Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
RU8/H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
RU8/Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
RU9/H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
RU9/Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
RAT/H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
RAT/Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Lokátor se jeví sám o sobě jako robustní vzhledem k výpadku jednotlivých signálů ve snímku signalizací vypínačů a ochran, což znamená, že je schopen správně lokalizovat poruchu i v případě částečného zašumění snímku. Porucha je lokalizována excitací neuronu odpovídajícímu příslušnému prvku rozvodny. Robustnost lokalizace poruch může být podpořena předběžnou filtrací šumu uvedeného snímku, tj. nejprve filtr provede ortogonální projekci zašuměného snímku do nadroviny generované 18 lineárně nezávislými vzájemně ortogonálními snímky o 66 složkách, která je poté předložena lokátoru.

Byla simulována porucha na sběrnici A1 provázená výpadkem signalizace vypínače V6A a ochrany RA1, porucha na vývodu L1 provázená výpadkem signalizace ochrany DL1 a porucha na vývodu U1 provázená výpadkem signalizace vypínačů V3A a V3AB. Výsledky filtrace šumu (první tabulka) a lokalizace poruchy (druhá tabulka) jsou uvedeny dále:

	A1		L1		U1	
filtrace	před	po	před	po	před	po
V1A	1	0,7	0	0,0	0	-0,1
V2A	1	0,7	1	0,5	0	-0,1
V3A	1	0,7	0	0,1	0	0,5
V4A	1	0,7	0	0,0	0	-0,1
V5A	1	0,7	0	0,1	0	-0,1
V6A	0	0,5	0	0,1	0	-0,1
V3AB	0	0,1	0	0,0	0	0,4
V2B	0	0,1	1	0,5	0	0,0
RA1/H	0	0,6	0	0,1	0	-0,1
RA1/Z	0	0,6	0	0,1	0	-0,1
ASV/A1	1	0,6	0	0,1	0	-0,1
DL1/H	0	0,1	0	0,5	0	0,0
DL1/Z	0	0,1	0	0,5	0	0,0
RU1/H	0	0,1	0	0,0	1	0,6
RU1/Z	0	0,1	0	0,0	1	0,6

	A1		L1		U1	
lokalizace	bez filtru	s filtrem	bez filtru	s filtrem	bez filtru	s filtrem
A1	0,5	0,9	0,0	0,2	-0,1	-0,1
A2	0,0	-0,1	0,0	0,0	0,0	0,0
B1	0,0	-0,1	0,1	0,2	-0,1	0,0
B2	0,0	0,0	0,0	-0,1	0,2	0,1
L1	0,0	0,1	0,4	0,7	-0,1	0,0
L2	0,0	0,1	-0,1	-0,1	-0,1	0,0
L3	0,0	0,0	0,1	0,0	0,1	0,0
L4	0,2	0,0	-0,1	0,1	0,2	0,0
U1	0,0	0,1	0,0	0,1	0,4	0,9
U2	-0,1	0,0	0,0	-0,1	-0,1	-0,2
U3	0,1	0,1	-0,1	-0,1	-0,1	0,0
U4	-0,1	0,0	0,0	0,0	0,0	0,0
U5	0,0	0,0	0,0	0,0	0,2	0,1
U6	0,0	0,0	0,0	0,0	0,0	0,0
U7	0,1	0,1	0,1	0,1	-0,1	0,0
U8	-0,1	-0,1	0,2	0,1	0,2	0,1
U9	0,1	-0,1	0,0	0,0	0,2	0,0
AT	0,2	0,1	0,3	0,0	0,2	0,1

Bez filtrace šumu je porucha na vývodu L1 lokalizována na hranici rozpoznatelnosti, tj. jí odpovídající neuron je excitován na hodnotu 0,4 a neuron odpovídající poruše vývodu AT je excitován na hodnotu 0,3, při filtraci šumu je jí odpovídající neuron excitován na hodnotu 0,7, zatímco falešně excitovaný neuron zůstává inhibován.

17. Příklad aplikace optimalizačního modelu

Mějme úplný orientovaný graf s ohodnocenými hranami, tj. uspořádanou čtveřici $[V, E, \varepsilon, w]$, kde:

V	množina vrcholů
E	množina hran
ε	zobrazení incidence hran s vrcholy ($\varepsilon: E \to V \times V$)
ρ	ohodnocení hran ($\rho: \varepsilon(E) \to \mathbb{R}$),

pro $\rho([i,j]) \equiv \rho_{ij}$ a $|V| = m$, $n = m^2$ dostaneme otevřený problém obchodního cestujícího:

$$f: \mathbb{R}^n \to \mathbb{R} \quad f(\vec{x}_0) = \min_{\vec{x} \in \Omega} f(\vec{x}) \quad \Omega \subset \{0,1\}^n \quad f(\vec{x}) = \sum_i \sum_j \rho_{ij} x_{ij}$$

každý vektor $\vec{x} \in \Omega$ vybírá jistou cestu v grafu, tj. x_{ij} je stav hrany směřující z i-tého do j-tého vrcholu pro $i, j \in V$, kde $x_{ii} = 0$.

Následující optimalizační experiment, řešící výše uvedený problém, byl proveden na souboru 45 evropských měst (Tab. 1) užitím optimalizačního modelu resp. simulovaného žíhání, vždy pro dva způsoby sestupu teploty, tj. exponenciální a lineární (Obr. 1). Experiment byl prováděn pro hodnoty počátečních teplot (0,1 ÷ 1000) resp. hodnoty parametru SEED (0 ÷ 1000) a pro počet iterací (10^3 ÷ 10^6) resp. (10^3 ÷ 10^9), viz Tab. 2 (optimalizační model) a Tab. 3 (simulované žíhání) obsahující i průměrné optimální vzdálenosti Ø a průměrné absolutní odchylky σ. Všechny experimenty simulovaného žíhání byly provedeny pro počáteční teplotu rovnou jedné, s výjimkou poslední řady (10^9 iterací) experimentů pro lineární sestup, kde počáteční teplota byla nastavena na pět.

Amsterdam	Dublin	Kobenhavn	Munchen	Strasbourg
Ankara	Dubrovnik	Lisboa	Narvik	Venezia
Athenai	Edinburgh	Liverpool	Oslo	Warszawa
Barcelona	Frankfurt	London	Palermo	Wien
Beograd	Geneve	Luxenbourg	Paris	Zurich
Berlin	Hamburg	Madrid	Praha	
Bratislava	Hammerfest	Malaga	Roma	
Bruxelles	Helsinki	Marseille	Salzburg	
Bucuresti	Instanbul	Milano	Sofia	
Budapest	Kijev	Moskva	Stockholm	

Tab. 1 Užitá evropská města

Exp	0.1	1	5	10	50	100	500	1000	Ø	σ
3	63599	44225	43778	42580	47373	43111	43770	46660	46887	4300
4	63599	35968	35403	36682	37668	40212	36563	41637	40967	5826
5	34278	38222	38557	34278	33610	37330	35180	39517	36372	2035
6	38912	35892	35559	33668	35559	38912	35892	35559	36244	1334

Lin	0.1	1	5	10	50	100	500	1000	Ø	σ
3	63599	49691	56409	63494	ERR	ERR	ERR	ERR	58298	5248
4	63599	35818	39703	49691	56409	63599	ERR	ERR	51470	9733
5	78595	37522	40870	35818	39703	45432	57134	63317	49799	12412
6	78595	35559	35559	35815	36239	35818	40657	40764	42376	9055

Tab. 2 Optimální vzdálenosti pro různé počáteční hodnoty teploty a řády počtu iterací

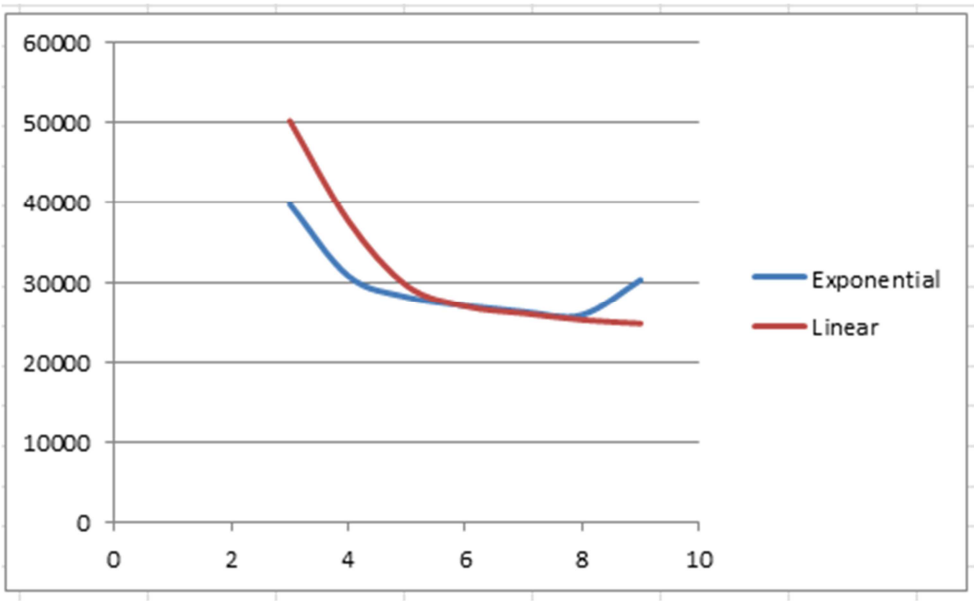
Exp	0	1	5	10	50	100	500	1000	Ø	σ
3	39544	39009	39315	40824	38577	42079	42890	36130	39796	1601
4	30663	33507	26551	29417	33686	30116	31876	31343	30895	1708
5	27146	26944	27969	30623	29658	27899	27421	28596	28282	1008
6	26534	25932	29001	27070	26157	28419	27805	26466	27173	927
7	26528	25460	26452	25361	24428	26454	29066	27468	26402	989
8	25934	24786	24557	26812	25545	26471	28403	25428	25992	928
9	33664	33617	31528	31320	26114	29320	27255	30521	30417	2141

Lin	0	1	5	10	50	100	500	1000	Ø	σ
3	48024	54057	48038	49988	53922	53167	42890	51830	50240	3005
4	40581	38116	36496	37282	35368	40152	37476	36299	37721	1421
5	29916	29618	29475	31216	29126	29384	28417	30391	29693	611
6	27273	27171	25851	28421	26072	26830	27566	27584	27096	634
7	26904	25917	25618	25798	25442	26152	26941	26758	26191	507
8	25191	24922	25343	26162	24741	26373	24504	25713	25369	536
9	24360	24074	24881	25577	25071	24558	24924	25811	24907	439

Tab. 3 Optimální vzdálenosti pro různé počáteční hodnoty SEED a řády počtu iterací

Řešení získané optimalizačním modelem resp. simulovaným žíháním je 33 610 km (Tab. 4, Obr. 2) resp. 24 074 km (Tab. 5, Obr. 3). Optimálním řešením je 23 932 km, zatímco nejlepší řešení získané simulovaným žíháním je 24 074 km, tj. pouze o 142 km (0,6%) horší.

Algoritmus simulované žíhání poskytuje mnohem lepší výsledky než kvazigradientní sestup optimalizačního modelu, trajektorie kvazigradientího sestupu optimalizačního modelu, na rozdíl od algoritmu simulovaného žíhání, během experimentu zřejmě většinou nepřekročila oblast hlubšího lokálního minima, v kterém uvázla.



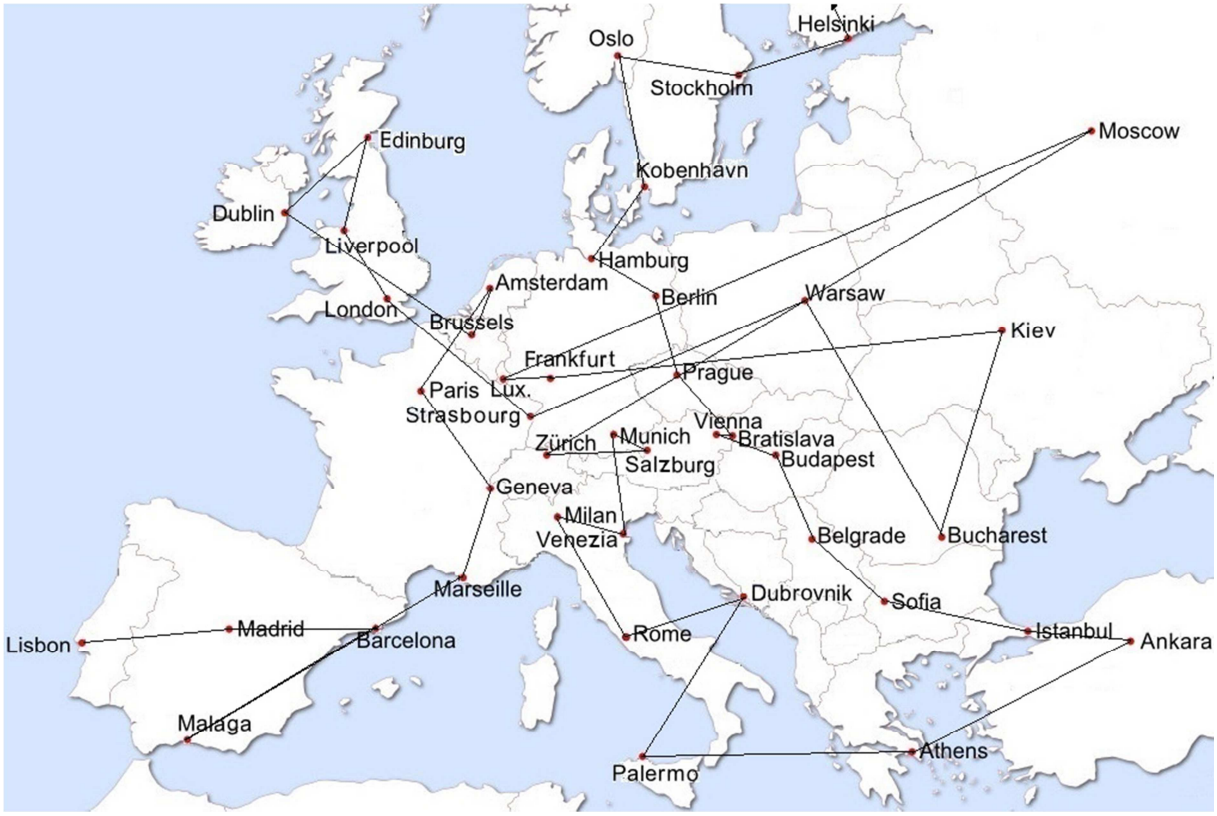
Obr. 1 Průměrná optimální vzdálenost v závislosti na řádu počtu iterací

Lisboa	Edinburgh	Zurich	Instanbul	Oslo
Madrid	Liverpool	Salzburg	Sofia	Stockholm
Barcelona	London	Munchen	Beograd	Helsinki
Malaga	Strasbourg	Venezia	Budapest	Narvik
Marseille	Warszawa	Milano	Wien	Hammerfest
Geneve	Bucuresti	Roma	Bratislava	
Paris	Kijev	Dubrovnik	Praha	
Amsterdam	Frankfurt	Palermo	Berlin	
Bruxelles	Luxenbourg	Athenai	Hamburg	
Dublin	Moskva	Ankara	Kobenhavn	

Tab. 4 Řešení optimalizačního modelu

Lisboa	Athenai	Salzburg	London	Helsinki
Malaga	Ankara	Munchen	Dublin	Stockholm
Madrid	Instanbul	Zurich	Liverpool	Oslo
Barcelona	Bucuresti	Geneve	Edinburgh	Narvik
Marseille	Sofia	Strasbourg	Kobenhavn	Hammerfest
Milano	Beograd	Frankfurt	Hamburg	
Venezia	Budapest	Luxenbourg	Berlin	
Roma	Bratislava	Paris	Warszawa	
Dubrovnik	Wien	Bruxelles	Kijev	
Palermo	Praha	Amsterdam	Moskva	

Tab. 5 Řešení simulovaného žíhání



Obr. 2 Řešení optimalizačního modelu

Obr. 3 Řešení simulovaného žíhání

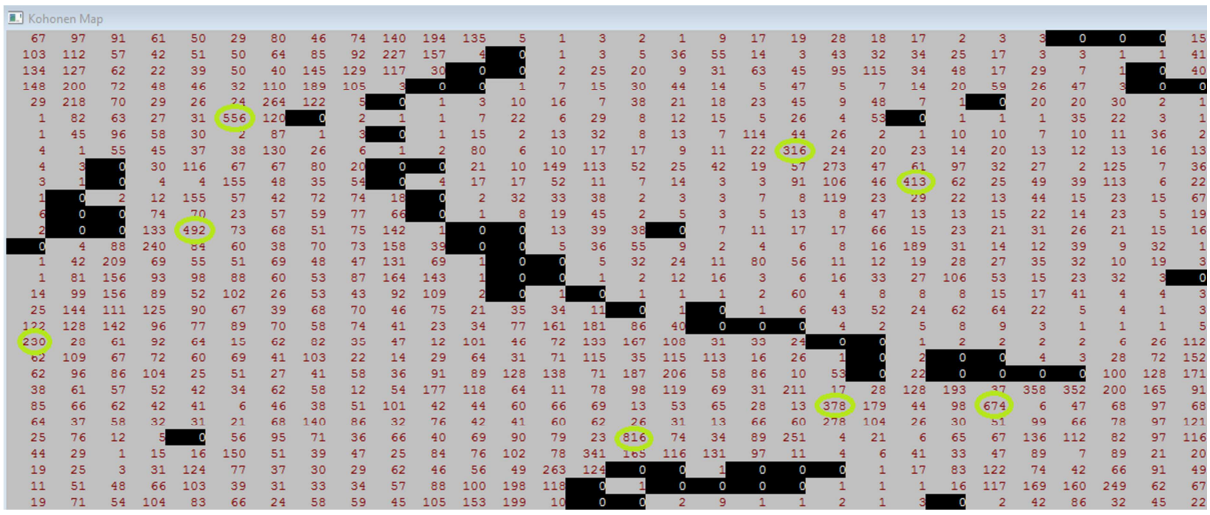
18. Příklad aplikace kompetičního modelu

Experiment proběhl nad souborem zaznamenaným během průběhu kampaně jedné z portugalských bank, spočívající v oslovování klientů s nabídkou termínovaného vkladu. Zmíněný soubor obsahoval 45 211 klientů charakterizovaných 16 atributy, přičemž u každého klienta byla uvedena informace o výsledku nabídky, tj. 5 289 klientů nabídku akceptovalo, zbylých 39 922 klientů nabídku neakceptovalo. Cílem experimentu bylo užití neuronové sítě ke klasifikaci klientů z hlediska dvou výše uvedených kategorií, resp. predikci, zda je klient náchylný uvedenou nabídku akceptovat či nikoli, a to na základě jemu přiřazených 16 následujících atributů:

bank client data:
1 - age
2 - job: 0-unknown, 1-unemployed, 2-student, 3-housemaid, 4-retired, 5-blue-collar, 6-administrator, 7-technician, 8-services, 9-management, 10-self-employed, 11-entrepreneur
3 - marital: 0-divorced/widowed, 1-single, 2-married
4 - education: 0-unknown, 1-primary, 2-secondary, 3-tertiary
5 - has credit in default? 1-yes, 0-no
6 - average yearly balance in euros
7 - has housing loan? 1-yes, 0-no
8 - has personal loan? 1-yes, 0-no
related with the last contact of the current campaign:
9 - contact: 0-unknown, 1-cellular, 2-telephone
10 - last contact day of the month
11 - last contact month of year
12 - last contact duration in seconds
other attributes:
13 - number of contacts performed during this campaign and for this client
14 - number of days that passed by after the client was last contacted from a previous campaign
15 - number of contacts performed before this campaign and for this client
16 - outcome of the previous marketing campaign: -1-failure, 0-unknown, 1-success, 2-other

V první fázi experimentu byla nejprve provedena segmentace (shluková analýza) klientů dle jejich atributů, a to jednak s cílem redukce jejich počtu z důvodu příliš velké mohutnosti souboru klientů pro tvorbu tréninkových dat klasifikátoru, a jednak s cílem eliminace možných rozporů v chování klientů, tj. odlišností v rozhodování vzájemně si podobných klientů. V druhé fázi experimentu pak proběhla vlastní adaptace klasifikátoru klientů.

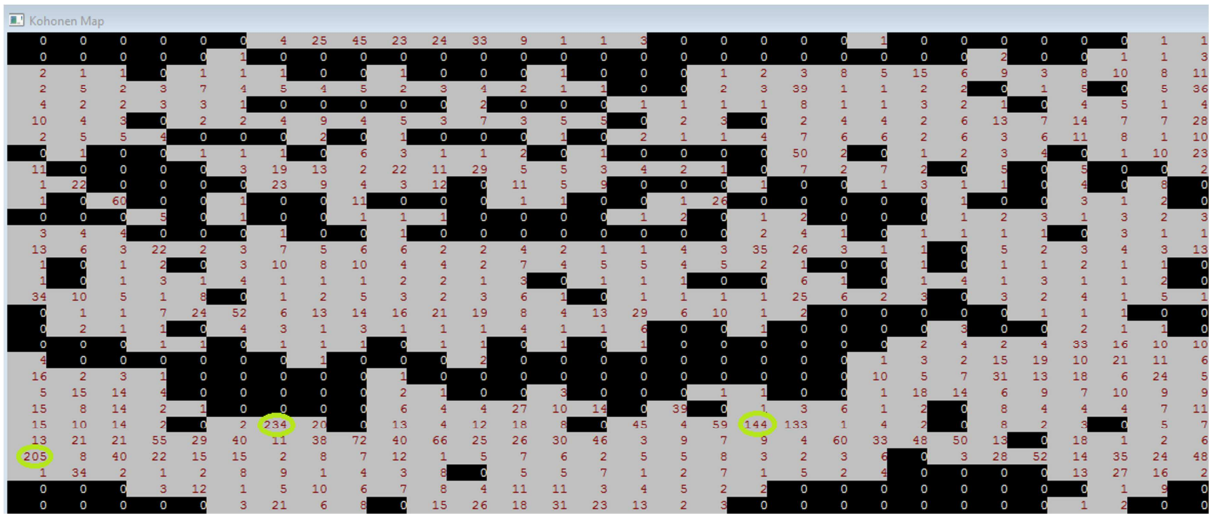
Na následujícím obrázku je uvedena Kohonenova mapa zobrazující rozmístění všech klientů v prostoru daného jejich atributy:

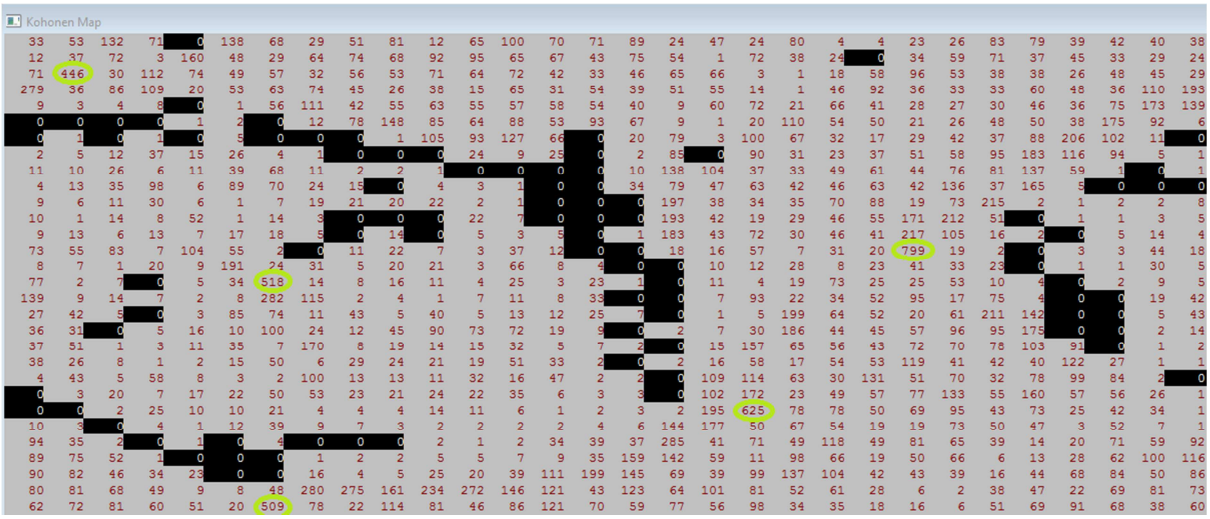


Odečtem Kohonenovy mapy bylo identifikováno osm nejvýznamnějších shluků klientů, avšak z poměrného zastoupení klientů obou kategorií v každém shluku nelze shluky klientů jednoznačně identifikovat s žádnou z jejich kategorií:

	1	2	3	4	5	6	7	8	Σ
yes	97	102	80	100	70	29	44	15	537
no	719	572	476	392	343	349	272	215	3338
Σ	816	674	556	492	413	378	316	230	3875
yes/no	0,13	0,18	0,17	0,26	0,20	0,08	0,16	0,07	0,16

Z výše uvedeného důvodu byla provedena dekompozice souboru klientů do dvou souborů daných jejich chováním „yes or no“ spolu se zobrazením jejich rozmístění:





Odečtem výše uvedených Kohonenových map byly identifikovány tři nejvýznamnější shluky klientů s chováním „yes“ a pět nejvýznamnějších shluků klientů s chováním „no“, prototypy (reprezentanti) každého ze zmíněných shluků jsou uvedeny níže:

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
yes	234	29	6	1	2	0	1,233	0	0	1	8	6	5,12	2	0,0	0	0
	205	30	7	1	2	0	1,575	0	0	1	24	6	5,44	2	0,0	0	0
	144	62	5	2	2	0	2,356	0	0	1	13	6	4,95	2	0,0	0	0
no	799	29	6	1	2	0	0,983	1	0	1	27	5	2,22	3	-0,1	0	0
	625	29	6	1	2	0	0,916	1	0	1	5	5	2,29	2	-0,1	0	0
	518	35	6	1	2	0	0,926	1	0	1	13	5	2,19	2	36,0	3	0
	509	59	6	2	2	0	2,094	0	0	1	8	6	2,28	2	-0,1	0	0
	446	58	6	2	2	0	1,729	0	0	1	25	7	2,05	4	-0,1	0	0
yes	ϕ	43	6	1	2	0	1,935	0	0	1	15	6	5,07	2	9,8	2	0
	max	62	7	2	2	0	4,401	1	0	1	25	7	6,93	3	27,4	4	1
no	ϕ	41	7	1	2	0	1,363	1	0	1	16	6	2,22	3	5,8	1	0
	max	59	7	2	2	0	2,094	1	0	1	29	8	2,52	5	36,0	5	0
							*1000						*100		*10		

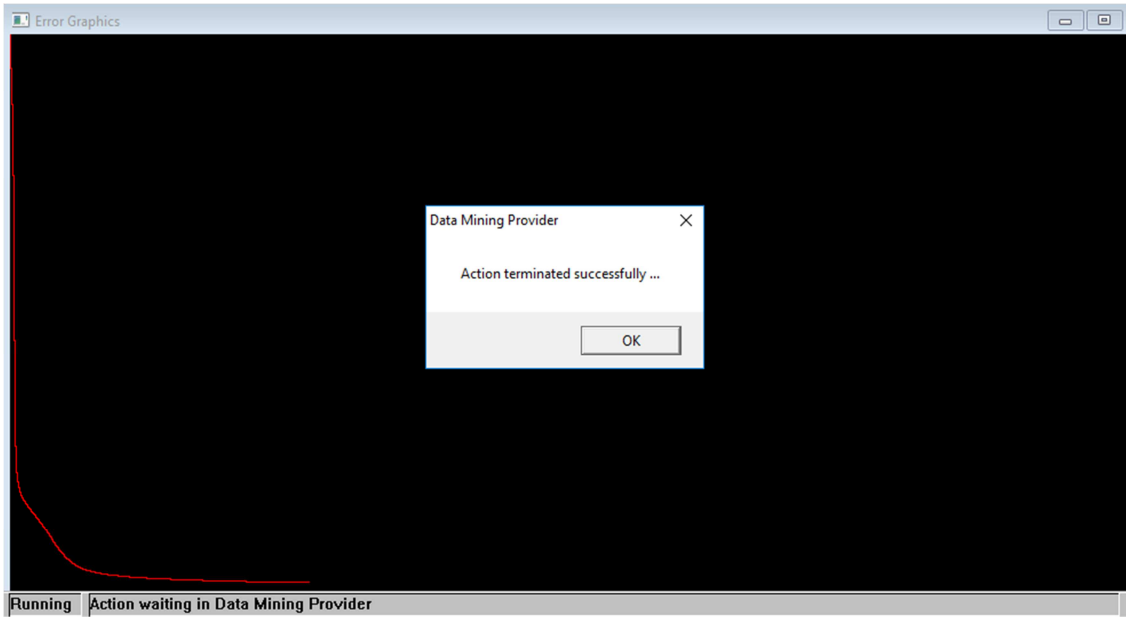
spolu s průměrnou a maximální hodnotou každého atributu, počítanou přes všechny klienty příslušného chování, obsažené ve zmíněných shlucích, přičemž původní hodnoty atributů 6 resp. 12 resp. 14 byly normalizovány čísly 1000 resp. 100 resp. 10.

Váhové vektory obou výše uvedených Kohonenových map pak byly užity jako prvky tréninkových dat klasifikátoru, přičemž pro klienty s chováním „no“ byly užity pouze ty váhové vektory, ke kterým se přihlásilo minimálně deset klientů a více, čímž byl uveden do vzájemné proporce počet klientů obou dvou způsobů jejich chování.

Jako prvků tréninkových dat klasifikátoru bylo užito 1 207 vzorových klientů, z toho 603 klientů s chováním „yes“ a 604 klientů s chováním „no“, na kterých byla provedena korelační analýza vlivu jednotlivých atributů na chování klienta:

	KENDALL		SPEARMAN	
	yes	no	yes	no
1	0,0	0,0	0,1	-0,1
2	-0,8	0,8	-0,3	0,3
3	-0,8	0,8	-0,4	0,4
4	0,0	0,0	0,0	0,0
5	0,0	0,0	0,0	0,0
6	0,5	-0,5	0,7	-0,7
7	-1,0	1,0	-0,8	0,8
8	0,0	0,0	0,0	0,0
9	0,0	0,0	0,0	0,0
10	-0,1	0,1	0,0	0,0
11	-0,1	0,1	0,3	-0,3
12	0,7	-0,7	0,9	-0,9
13	-1,0	1,0	-0,6	0,6
14	0,4	-0,4	0,5	-0,5
15	0,1	-0,1	0,4	-0,4
16	-1,7	1,7	0,2	-0,2

z které vyplývá, že významnější korelaci s chováním klientů vykazují pouze atributy 2, 3, 6, 7, 12 a 13, ostatní atributy byly z tréninkových dat vyřazeny. Adaptace klasifikátoru proběhla s parametry A=0,001, B=0,5 a C=1,01 (rychlost, setrvačnost, zrychlení) při povolené chybě klasifikace do výše 20%:



Z výše uvedeného přibližně hyperbolického průběhu chyby funkce sítě pro zvolený limit iterací 10 000 plyne, že se zadanou přesností ±20% si adaptace klasifikátoru vyžádala cca 2 700 iterací.

Funkce sítě byla testována na 4 521 z tréninkových dat náhodně vybraných klientech, přičemž chybovost klasifikace je výrazně posunuta v neprospěch klientů s chováním „yes“, a to zřejmě z důvodu jejich menšího zastoupení v pozorovaných datech, síť má s nimi menší zkušenost.

Analogicky byl proveden ještě experiment pro pozměněnou skladbu atributů klientů, tj. k šesti původním atributům byl jednak přidán atribut 1 a jednak atributy 1, 10, 11, 14 a 15 s výsledkem viz výše, jako optimální skladba atributů se jeví skladba se sedmi atributy, kde v sumě testovacích dat resp. pro klienty s chováním „no“ je chybovost klasifikace 20% resp. 17% sice nejvyšší, ale naproti tomu došlo k poklesu chybovosti klasifikace u klientů s chováním „yes“ na minimálních 36%, což je cca pouhý dvojnásobek chybovosti klasifikace klientů s chováním „no“:

6 atr	invalid	valid	Σ	error [%]
Σ	805	3716	4521	18
yes	201	320	521	39
no	604	3396	4000	15
7 atr				
Σ	883	3638	4521	20
yes	188	333	521	36
no	695	3305	4000	17
11 atr				
Σ	705	3816	4521	16
yes	207	314	521	40
no	498	3502	4000	12

Obecně lze říci, že klienti s chováním „yes vs. no“ se markantně liší ve velikosti atributu 12, přičemž obě skupiny se dále rozpadají na podskupiny třicátníků a šedesátníků, vzájemně se lišících ve velikosti atributu 6.

19. Příklad aplikace nelineárního modelu (deep learning)

Výpočetní experiment s cílem ověřit efektivitu hloubkového učení byl proveden na hluboké neuronové síti o deseti vrstvách, každá o deseti neuronech s výjimkou poslední vrstvy, která se sestávala ze tří neuronů. Síť byla adaptována na klasifikaci objektů (popsaných deseti atributy) z hlediska tří kategorií. Během učení byly adaptovány všechny synaptické váhy a parametry aktivačních funkcí (prahy a strmosti) neuronů skrytých vrstev:

$$w_{ij}(T) = w_{ij}(T-1) + \alpha\, y_i(T)\, g_j(x_j(T)) + \mu\, \Delta w_{ij}(T-1)$$

$$\vartheta_i(T) = \vartheta_i(T-1) + \alpha\, g_i(x_i(T)) + \mu\, \Delta \vartheta_i(T-1)$$

$$p_i(T) = p_i(T-1) + \alpha\, x_i(T)\, g_i(x_i(T)) + \mu\, \Delta p_i(T-1)$$

$$g_k(x_k) = p_k y_k (1-y_k)(z_k - y_k) \qquad g_i(x_i) = p_i y_i (1-y_i) \sum_j g_j(x_j)\, w_{ij}$$

pro $k \in V_N$, $j \in V_L$, $i \in V_{L-1}$, $L \in \{2, \dots, N\}$, $T \in \{1, \dots, M\}$, kde:

x_i potenciál *i*-tého neuronu

y_i skutečný stav *i*-tého neuronu

z_i požadovaný stav *i*-tého neuronu

g_i adaptační funkce *i*-tého neuronu

ϑ_i práh *i*-tého neuronu

p_i strmost aktivační funkce *i*-tého neuronu

w_{ij} synaptická váha vazby *i*-tého neuronu s *j*-tým neuronem

α rychlost učení

μ míra setrvačnosti gradientního sestupu

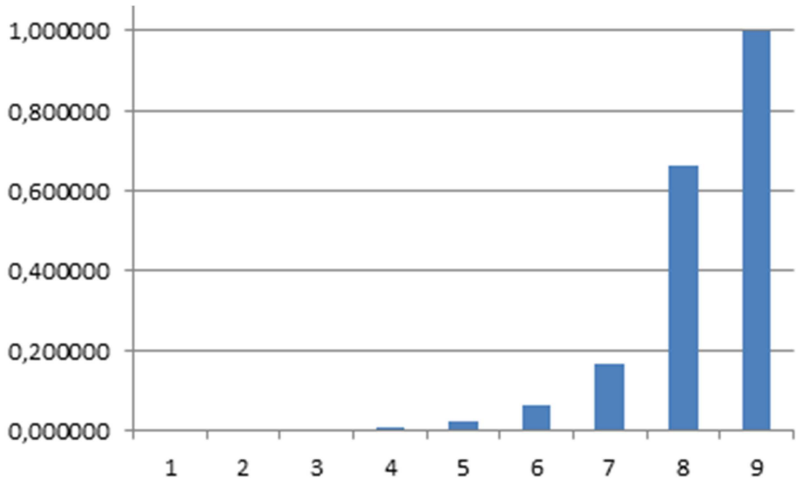
V_L populace neuronů *L*-té vrstvy

N počet vrstev sítě

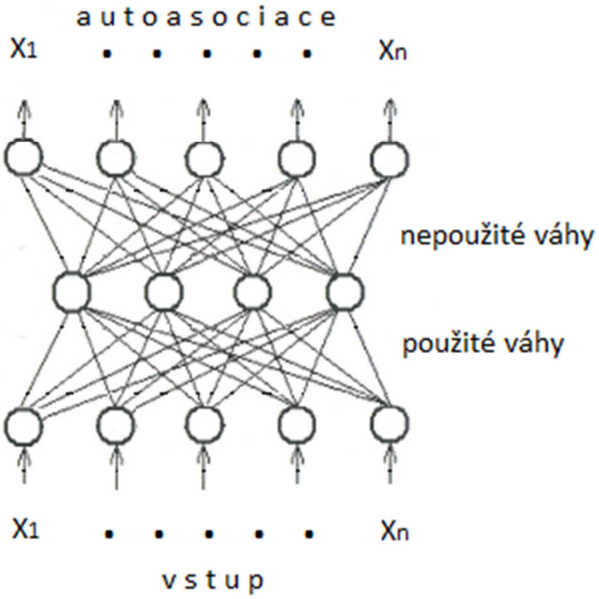
M počet prvků trénovací množiny

Δ předcházející přírůstek příslušné proměnné

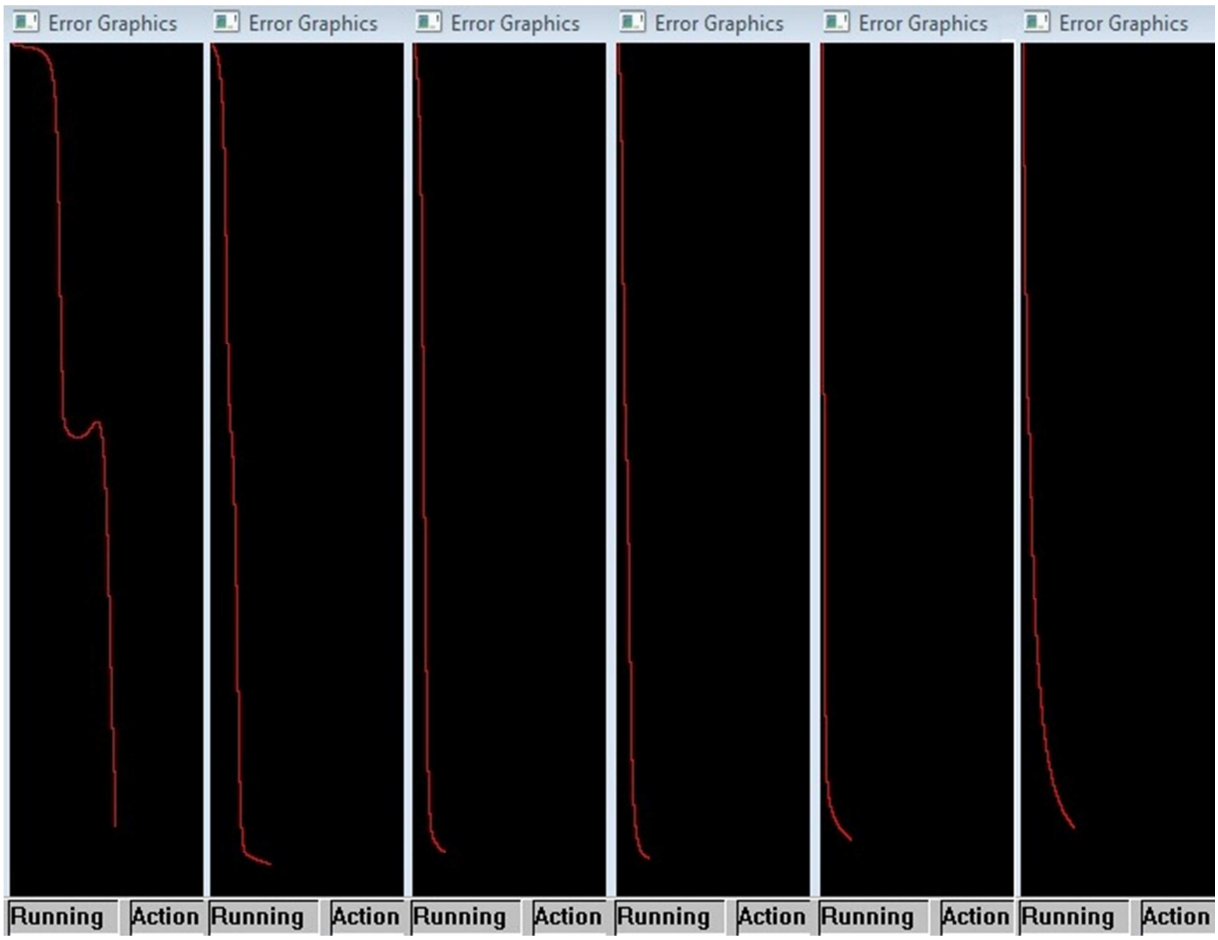
Nejprve se síť adaptovala pouze výše uvedeným algoritmem BPA, pak se určila míra změn vah v jednotlivých vrstvách sítě ve směru od poslední vrstvy k první vrstvě, tj. během zpětného šíření chyby požadované funkce sítě:



Poté se síť adaptovala ve dvou fázích (hloubkové učení), v první fázi se předučovaly jednotlivé vrstvy vah postupně od první vrstvy do čtvrté vrstvy a poté od první vrstvy do páté vrstvy atd. až do deváté vrstvy, a to na autoasociativní funkci sítě o dvou vrstvách vah, z nichž první vrstva byla vždy příslušná skutečná vrstva původní sítě a druhá vrstva byla vždy dodatečně přidaná fiktivní vrstva z důvodu zkompletování příslušných postupně nad sebe řazených dvouvrstvých autokodérů:



V druhé fázi se pak celá předučená síť doučila opět výše uvedeným algoritmem BPA. Jako optimálním, z hlediska minimalizace počtu předučovaných vrstev vah, se jeví předučení první až páté vrstvy vah, od které se váhy směrem k poslední deváté vrstvě během BPA doučení pak již dostatečně mění, viz následující průběhy chyby funkce sítě v závislosti na počtu předložení tréninkových dat během BPA učení s předučenými čtyřmi, pěti až devíti vrstvami:



Při předučení pouze první resp. první až druhé resp. první až třetí vrstvy vah se síť požadovanou funkci naučila, ale až po výrazně větším počtu předložení tréninkových dat, než v ostatních na obrázku výše uvedených případech.

20. Zkratky

ANN	Artificial Neural Network
ART	Adaptive Resonance Theory
BAM	Bi-directional Associative Memory
BPA	Back Propagation Algorithm
CHN	Continuous Hopfield Network
CPN	Counter Propagation Network
HAM	Hopfield Associative Memory
LAM	Linear Associative Memory
MLP	Multi-Layer Perceptron
RAM	Recurrent Associative Memory
RBF	Radial Basis Function
SOM	Self-Organizing Map

21. Literatura

Grossberg S. - The Adaptive Brain, North-Holland, 1987

Hebb D.O. - The organization of behavior, New York: Wiley, 1949

Hecht-Nielsen R. - Counterpropagation networks, Appl. Opt. 26(23), 1987

Hopfield J.J. - Artificial Neural Networks, IEEE Circuits & Devices 4, 1988

Hornik K. - Multilayer Feedforward Networks Are Universal Approximators, Neural Networks 2, 1989

Kohonen T. - The Self-Organizing Map, Proc. IEEE 78(9), 1990

Kosko B. - Adaptive Bidirectional Associative Memories, Appl. Opt. 26(23), 1987

McCulloch W. S., Pitts W. - A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics 5:115–133, 1943

Rosenblatt F. - The Perceptron - a perceiving and recognizing automaton, Report 85–460–1, Cornell Aeronautical Laboratory, 1957