Aplikace teorie neuronových sítí

Doc. RNDr. Iveta Mrázová, CSc.

Katedra teoretické informatiky
Matematicko-fyzikální fakulta
Univerzity Karlovy v Praze

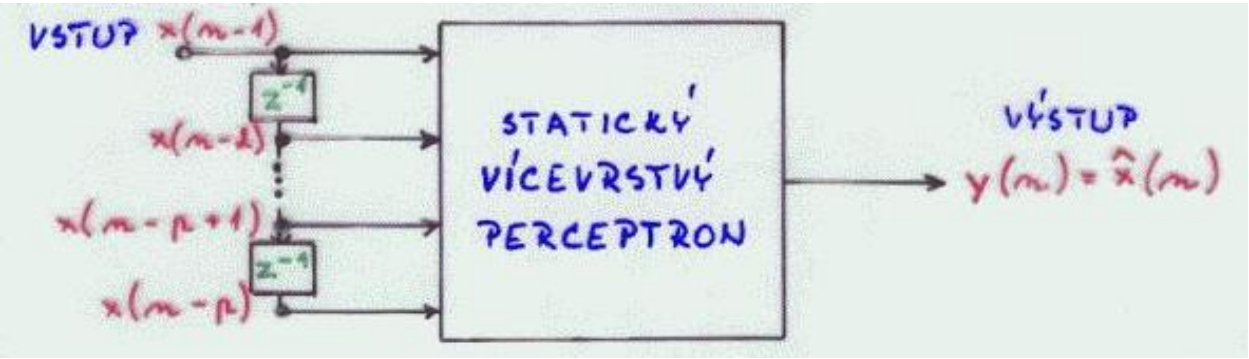
Zpracování časových vzorů (temporal processing)

Standardní algoritmus zpětného šíření:

- „statická“ struktura
- Vstupní vzor $\vec{x}$ zobrazen na výstup $\vec{y}$
- Výhodné pro rozpoznávání vzorů ($\vec{x}$ i $\vec{y}$ jsou prostorové vzory nezávislé na čase)

Zpracování časových vzorů (temporal processing)

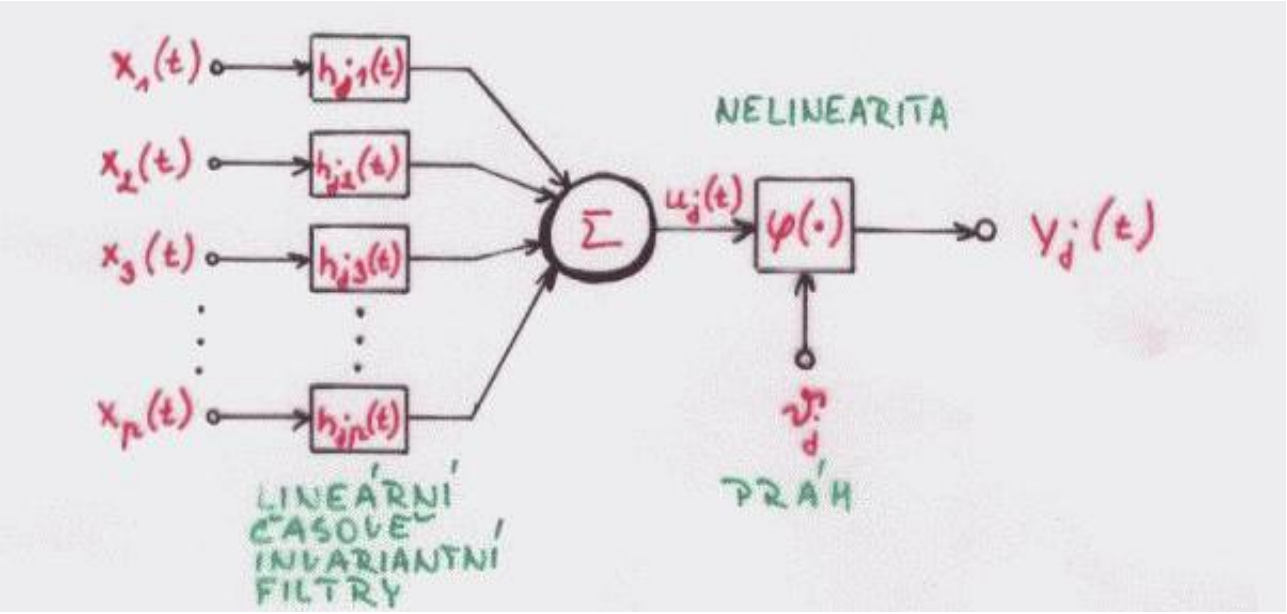
- Vhodné i pro nelineární predikci stacionárních časových řad
  - Jejich charakteristika se v čase nemění
  - Vstupní vektor $\vec{x} = [x(n-1), x(n-2), \ldots, x(n-p)]^T$
  - p …. řád predikce
  - z^{-1} … jednotkové zpoždění



Časoprostorový model neuronu

Dynamický model neuronu

- Každou synapsi nyní představuje lineární časově invariantní filtr



Časoprostorový model neuronu

Dynamický model neuronu

- $h_{ji}(t)$ … impulzní odezva lineárního časově invariantního filtru na synapsi i neronu j
- $x_i(t)$ … stimul na synapsi i (i = 1,2,…,p)
- odezva synapse i na vstup $x_i(t)$: konvoluce $h_{ji}(t)$ a $x_i(t)$

$$h_{ji}(t) * x_i(t) = \int_{-\infty}^{t} h_{ji}(\lambda)\, x_i(t-\lambda)\, d\lambda$$

- potenciál $v_j(t)$ neuronu j s p synapsemi:

$$v_j(t) = u_j(t) - \vartheta_j = \sum_{i=1}^{p} h_{ji}(t) * x_i(t) - \vartheta_j =$$

$$= \sum_{i=1}^{p} \int_{-\infty}^{t} h_{ji}(\lambda)\, x_i(t-\lambda)\, d\lambda - \vartheta_j$$

- Výstup $y_j(t)$ neuronu pomocí sigmoidy $\varphi(.)$: $y_j(t) = \varphi(v_j(t)) = \frac{1}{1+\exp(-v_j(t))}$

Model impulzní odezvy s konečnou dobou trvání (FIR-model)

~ **F**inite-duration **I**mpulse **R**esponse Model

- Synaptické filtry jsou **kauzální**
  - Synapse nedává žádnou odezvu, dokud jí není předložen nějaký vzor: $h_{ji}(t) = 0 \; ; \; t < 0$
- Synaptické filtry mají „konečnou paměť“
  - $h_{ji}(t) = 0 \; ; \; t > T$

    T … délka paměti (stejná pro všechny synapse)

$$=> \; v_j(t) = \sum_{i=1}^{p} \int_0^T h_{ji}(\lambda)\, x_i(t-\lambda)\, d\lambda - \vartheta_j$$

Model impulzní odezvy s konečnou dobou trvání (FIR-model)

Aproximace pomocí konvoluční sumy:

- Diskrétní čas: $t = n\,\Delta t$; $n \in N$

  Δt … vzorkovací interval

$$v_j(n\Delta t) = \sum_{i=1}^{p}\sum_{l=0}^{M} h_{ji}(l\,\Delta t)\, x_i(\Delta t(n-l))\,\Delta t - \vartheta_j =$$

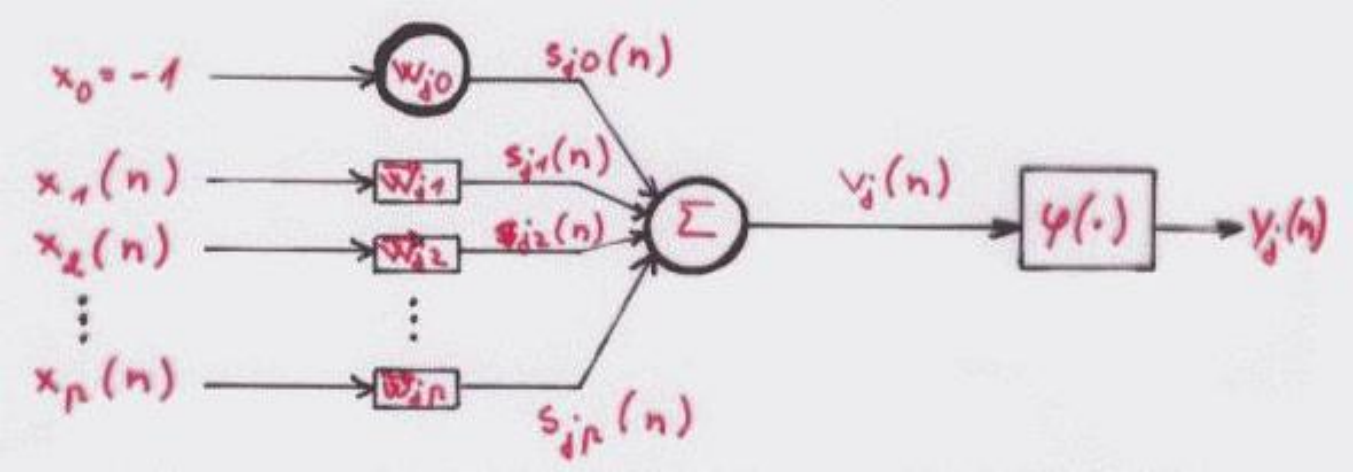
$$= \sum_{i=1}^{p}\sum_{l=0}^{M} w_{ji}(l\,\Delta t)\, x_i(\Delta t(n-l))\,\Delta t - \vartheta_j$$

- $M = T / \Delta t$
- $w_{ji}(l\,\Delta t) = h_{ji}(l\,\Delta t)(l\,\Delta t)$ $(i = 1, \ldots, p)$

Model impulzní odezvy s konečnou dobou trvání (FIR-model)

Zjednodušení (n je diskrétní časová proměnná):

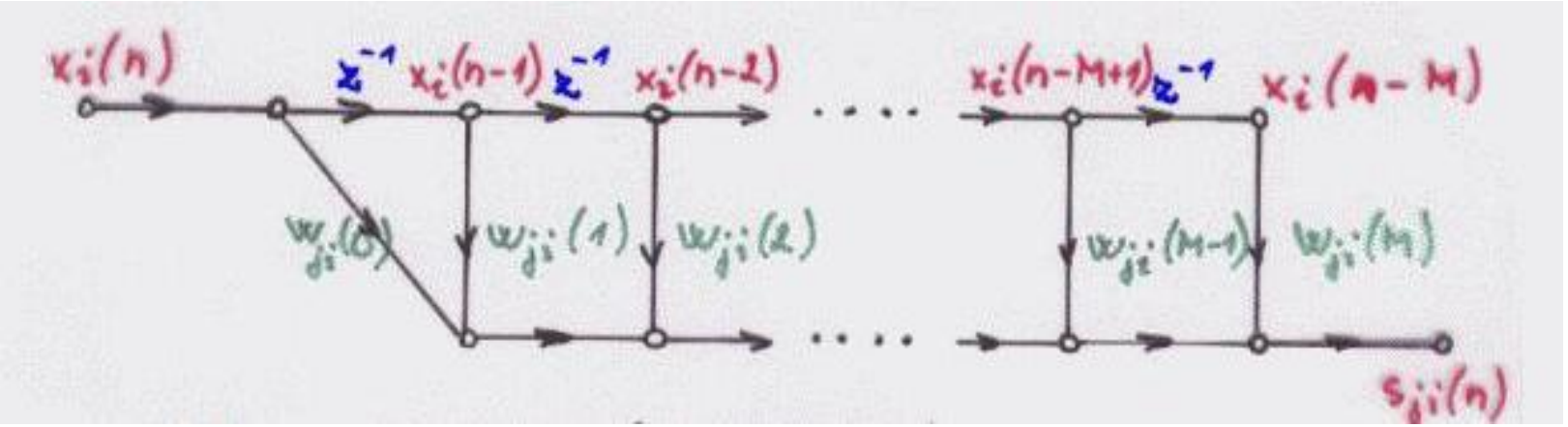
$$v_j(n) = \underbrace{\sum_{i=1}^{p} \underbrace{\sum_{l=0}^{M} w_{ji}(l)\, x_i(n-l)}_{\text{chování neuronu j v čase}}}_{\text{chování neuronu j v prostoru}} - \vartheta_j$$



Model impulzní odezvy s konečnou dobou trvání (FIR-model)

Zjednodušení (n je diskrétní časová proměnná):

- $\vec{w}_{ji}$ … váhový vektor synapse i neuronu j
- w_{jo} … práh ϑ_j (pro pevný vstup $x_0 = -1$)



z^{-1} … jednotkové zpoždění

Vícevrstvý perceptron typu FIR (Finite-duration Impulse Response)

$w_{ji}(l)$ … váha l-tého kroku FIR-filtru na synapsi mezi neuronem i a j; $0 \le l \le M$; M … celkové zpoždění

$s_{ji}(n)$ … signál na výstupu i-té synapse neuronu j

$x_i(n)$ … vstupní signál (v diskrétním čase n)

$$s_{ji}(n) = \sum_{l=0}^{M} w_{ji}(l)\, x_i(n-l)$$

Maticový zápis:

$$\vec{x}_i(n) = [x_i(n), x_i(n-1), \ldots, x_i(n-M)]^T$$
$$\vec{w}_{ji} = [w_{ji}(0), w_{ji}(1), \ldots, w_{ji}(M)]^T$$
$$s_{ji}(n) = \vec{w}_{ji}^{\,T}\, \vec{x}_i(n); \quad i = 1, \ldots, p$$

Vícevrstvý perceptron typu FIR (Finite-duration Impulse Response)

Výstup $y_j(n)$ neuronu j (s p synapsemi)

- Potenciál $v_j(n)$ neuronu j s prahem ϑ_j

$$v_j(n) = \sum_{i=1}^{p} s_{ji}(n) - \vartheta_j = \sum_{i=0}^{p} \vec{w}_{ji}^{T}\, \vec{x}_i(n) - \vartheta_j$$

- Nelineární přenosová funkce $\varphi(.)$: $y_j(n) = \varphi(v_j(n))$

Vícevrstvý FIR-perceptron

- Analogicky vytvářen jako statická verze
- Synaptické spoje nahrazeny dynamickou verzí

Učení vícevrstvého FIR-perceptronu

Učení s učitelem:

- Chybová funkce: $E(n) = \frac{1}{2} \sum_j e_j^2(n)$
  - j … neurony výstupní vrstvy
  - $e_j(n) = d_j(n) - y_j(n)$ ….. chybový signál
  - $y_j(n)$ ….. skutečný výstup v čase n
  - $d_j(n)$ ….. požadovaný výstup v čase n
- Cílová funkce: $E_{TOTAL} = \sum_n E(n)$
- **Gradientní přístup:** $\frac{\partial E_{TOTAL}}{\partial \vec{w}_{ji}} = \sum_n \frac{\partial E(n)}{\partial \vec{w}_{ji}}$

Učení vícevrstvého FIR-perceptronu

Rozvinutí sítě v čase:

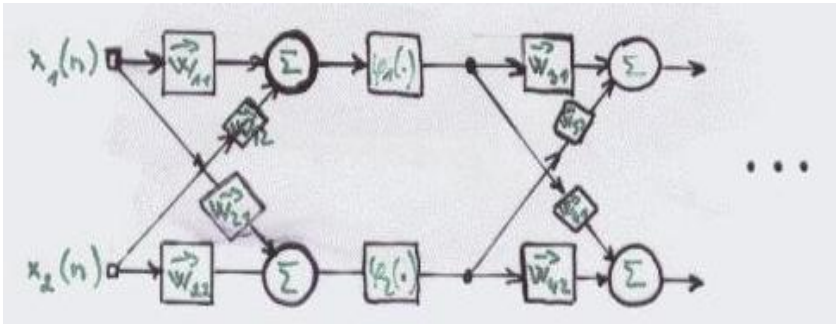
- Vytvoření ekvivalentní „statické“ sítě s větším počtem neuronů pomocí standardního algoritmu zpětného šíření
- **Dopředné rozvinutí v čase**
  - Začne se u vstupní vrstvy a postupuje se dále vrstvu po vrstvě
  - Nárůst architektury: O(DN)
    - D … celkový počet zpoždění
    - N … celkový počet volných parametrů sítě
- **Zpětné rozvinutí v čase**
  - Začne se od výstupní vrstvy, postupuje se po vrstvách směrem zpět
  - Geometrický nárůst architektury

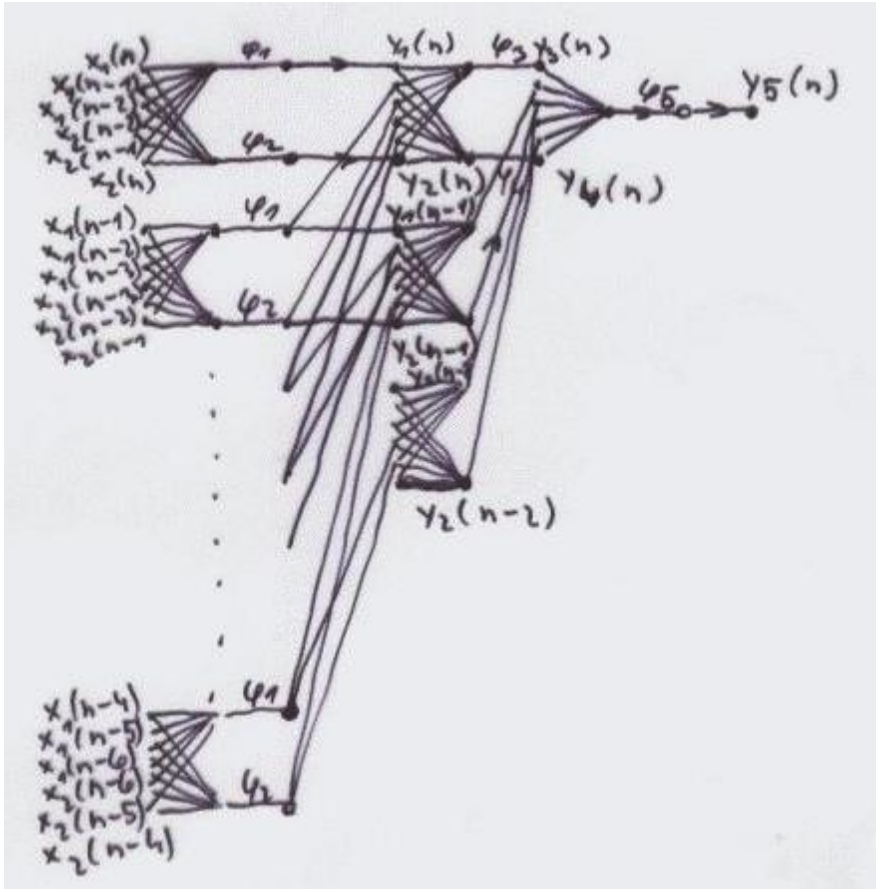
Rozvinutí sítě v čase - příklad

Příklad: síť 2 – 2 – 2 – 1

Dopředné rozvinutí sítě:

- 102 vah
- ~ (5x12)+(3x12) +6=102





Rozvinutí sítě v čase

Problémy:

- Velké množství redundantních vah
- Ztráta „symetrie“ mezi dopředným šířením stavu neuronů a zpětným šířením chybových členů
- Chybí rekurzivní vztah pro šíření chybových členů
- Nutná globální evidence „statických vah“

Časový algoritmus zpětného šíření

- Alternativní vyjádření:

$$\frac{\partial E_{TOTAL}}{\partial \vec{w}_{ji}(n)} = \sum_{n} \frac{\partial E_{TOTAL}}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial \vec{w}_{ji}(n)}$$

  n … časový index přes $v_j(n)$, ne přes $E(n)$

- Navíc platí: $\frac{\partial E_{TOTAL}}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial \vec{w}_{ji}(n)} \neq \frac{\partial E(n)}{\partial \vec{w}_{ji}(n)}$

  „=“ platí pouze pro sumy přes všechna n

- Adaptace vah podle:

$$\vec{w}_{ji}(n+1) = \vec{w}_{ji}(n) - \eta \frac{\partial E_{TOTAL}}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial \vec{w}_{ji}(n)}$$

  η … parametr učení

Časový algoritmus zpětného šíření

$$\frac{\partial v_j(n)}{\partial \vec{w}_{ji}(n)} = \vec{x}_i(n) \quad \text{a} \quad \delta_j(n) = -\frac{\partial E_{TOTAL}}{\partial v_j(n)}$$

- Potom: $\vec{w}_{ji}(n+1) = \vec{w}_{ji}(n) + \eta\, \delta_j(n)\, \vec{x}_i(n)$

1. Neuron j je z výstupní vrstvy:

$$\delta_j(n) = -\frac{\partial E_{TOTAL}}{\partial v_j(n)} = -\frac{\partial E(n)}{\partial v_j(n)} = e_j(n)\, \varphi'(v_j(n))$$

$e_j(n)$ ….. chyba měřená na výstupu neuronu j

$$\varphi'(v_j(n)) = \frac{\partial \varphi(v_j(n))}{\partial v_j(n)} = \frac{\partial y_j(n)}{\partial v_j(n)}$$

Časový algoritmus zpětného šíření

2. Neuron j je ze skryté vrstvy:

A ….. množina všech neuronů, které dostávají svůj vstup od neuronu j

$v_m(n)$ ..potenciál neuronu m z A

$$\delta_j(n) = -\frac{\partial E_{TOTAL}}{\partial v_j(n)} = -\sum_{m\in A}\sum_{n}\frac{\partial E_{TOTAL}}{\partial v_m(n)}\frac{\partial v_m(n)}{\partial v_j(n)} =$$

$$= \sum_{m\in A}\sum_{n}\delta_m(n)\frac{\partial v_m(n)}{\partial v_j(n)} =$$

$$= \sum_{m\in A}\sum_{n}\delta_m(n)\frac{\partial v_m(n)}{\partial y_j(n)}\frac{\partial y_j(n)}{\partial v_j(n)} =$$

$$= \varphi'(v_j(n))\sum_{m\in A}\sum_{n}\delta_m(n)\frac{\partial v_m(n)}{\partial y_j(n)}$$

Časový algoritmus zpětného šíření

2. Neuron j je ze skryté vrstvy (pokračování):

$y_j(n)$ ….. výstup neuronu j

$\varphi'(v_j(n)) = \partial y_j(n) / \partial v_j(n)$ … týká se neuronu $j \notin A$

$$v_m(n) = \sum_{j=0}^{p} \sum_{l=0}^{M} w_{mj}(l)\, y_j(n-l)$$

($w_{m0}(l) = \vartheta_m$ a $y_0(n-l) = -1$ $\forall l, n$)

M …. maximální počet zpoždění synaptického filtru m

p ….. počet synapsí neuronu m

\+ komutativní konvoluce

Potom: $$v_m(n) = \sum_{j=0}^{p} \sum_{l=0}^{M} y_j(l)\, w_{mj}(n-l)$$

Časový algoritmus zpětného šíření

2. Neuron j je ze skryté vrstvy (pokračování):

$$\frac{\partial v_m(n)}{\partial y_j(n)} = \begin{cases} w_{mj}(n-l) & 0 \le n-l \le M \\ 0 & \text{jinak} \end{cases}$$

a tedy: $$\delta_j(n) = \varphi'(v_j(n)) \sum_{m \in A} \sum_{n=l}^{M+l} \delta_m(n)\, w_{mj}(n-l) =$$

$$= \varphi'(v_j(n)) \sum_{m \in A} \sum_{n=0}^{M} \delta_m(n+l)\, w_{mj}(n)$$

dále definujeme: $$\vec{\Delta}_m(n) = [\delta_m(n), \delta_m(n+1), \ldots, \delta_m(n+M)]^T$$

$$\Rightarrow \quad \delta_j(n) = \varphi'(v_j(n)) \sum_{m \in A} \vec{\Delta}_m^T(n)\, \vec{w}_{mj}$$

Časový algoritmus zpětného šíření

Adaptační pravidla:

$$\vec{w}_{ji}(n+1) = \vec{w}_{ji}(n) + \eta\,\delta_j(n)\,\vec{x}_i(n)$$

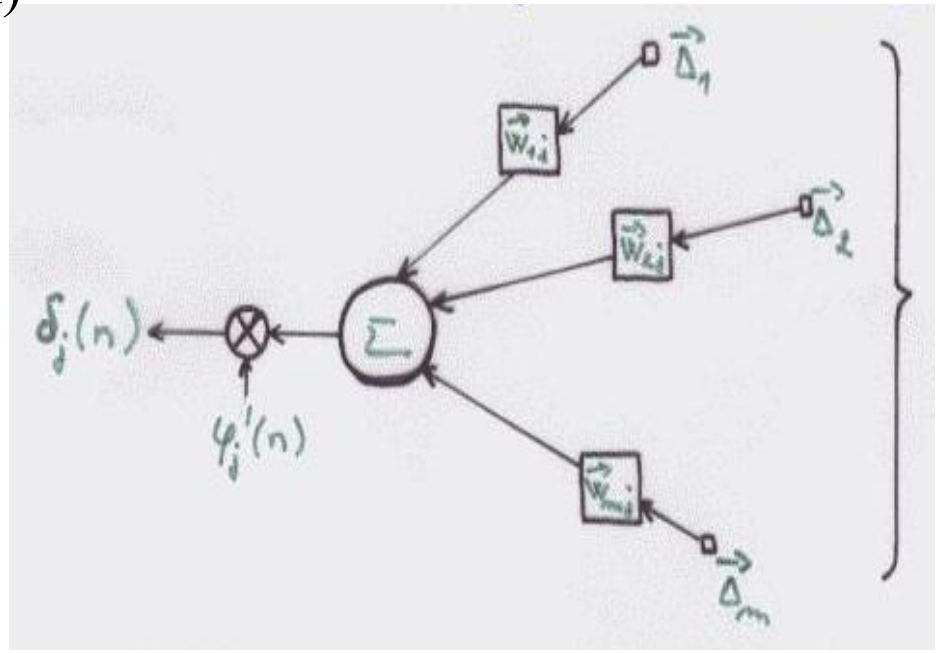
$$\delta_j(n) = \begin{cases} e_j(n)\,\varphi'\left(v_j(n)\right) & j \text{ z výstupní vrstvy} \\ \varphi'\left(v_j(n)\right)\sum_{m\in A}\vec{\Delta}_m^T(n)\,\vec{w}_{mj} & j \text{ ze skryté vrstvy} \end{cases}$$

- Zpětné šíření chybových členů $\delta_j(n)$ zahrnuje „zpětnou filtraci“ přes každou synapsi
- Zachování „symetrie“ mezi dopředným výpočtem a zpětným šířením chyby

Časový algoritmus zpětného šíření

Adaptační pravidla (analýza metody):

- Každý synaptický filtr je při výpočtu δ použit jen jednou (odpadá výpočet redundantních členů)
- Předpoklad pevných synaptických filtrů při výpočtu gradientu
- Zpětné šíření „lokálních“ gradientů
- **Výpočet $\delta_j(n)$ není kauzální** (vyžaduje znalost budoucích hodnot δ a w



neurony m v množině A

Časový algoritmus zpětného šíření

Kauzální omezení:

- Adaptace vah podle aktuálních a předchozích hodnot chybových signálů

  => přidání konečného počtu zpožďovacích jednotek na příslušná místa sítě

  - Pro výstupní vrstvu lze
  - Pro poslední skrytou vrstvu (n -- > n-M)

$$\delta_j(n-M) = \varphi'\left(v_j(n-M)\right) \sum_{m\in A} \vec{\Delta}_m^T(n-M)\,\vec{w}_{mj}$$

$$\vec{\Delta}_m(n-M) = \left[\delta_m(n-M), \delta_m(n+1-M), \ldots, \delta_m(n)\right]^T$$

Časový algoritmus zpětného šíření

Kauzální omezení (pokračování):

- NUTNOST uchovávat stavy $\vec{x}_i(n-M)$ kvůli výpočtu $\delta_j(n-M)\vec{x}_i(n-M)$ při adaptaci vah
- Pro další skryté vrstvy analogicky s větším zpožděním

Kauzální adaptační pravidla:

- Neuron j je z výstupní vrstvy:

$$\vec{w}_{ji}(n+1) = \vec{w}_{ji}(n) + \eta\,\delta_j(n)\,\vec{x}_i(n)$$

$$\delta_j(n) = e_j(n)\,\varphi'_j(n)$$

- Neuron j je ze skryté vrstvy:

$$\vec{w}_{ji}(n+1) = \vec{w}_{ji}(n) + \eta\,\delta_j(n-l\,M)\,\vec{x}_i(n-l\,M)$$

$$\delta_j(n-l\,M) = \varphi'_j\left(v_j(n-l\,M)\right)\sum_{m\in A}\vec{\Delta}^T_m(n-l\,M)\,\vec{w}_{mj}$$

Časový algoritmus zpětného šíření

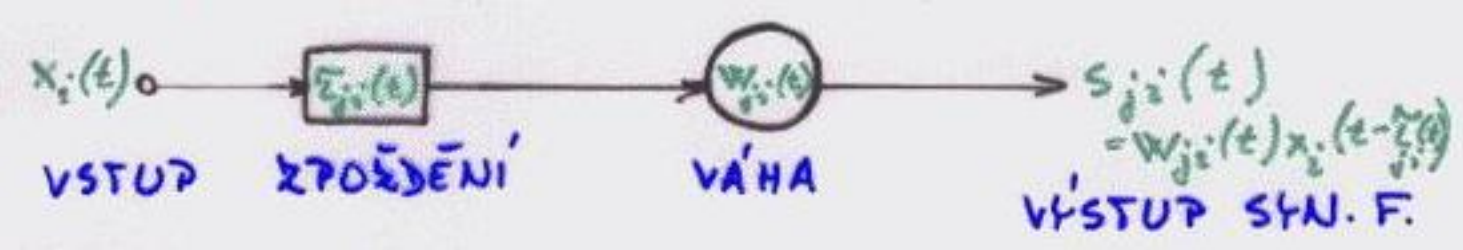
Kauzální adaptační pravidla:

- M ….. délka synaptického zpoždění
- l …… označuje l-tou vrstvu pod výstupní vrstvou
  (l=1 … poslední skrytá vrstva)

- Chybové členy δ se v síti šíří bez zpoždění
- Je třeba uchovávat i zpožděné hodnoty stavů $\vec{x}_i(n)$
  × zpětné šíření δ probíhá bez zpoždění
- Zachovaná „symetrie“ mezi dopředným šířením stavů a zpětným šířením chyb

Časový algoritmus zpětného šíření s adaptivním zpožděním

- Synaptické zpoždění $\tau_{ji}(t)$
- Váha synapse $w_{ji}(t)$
- Výstup synaptického filtru: $s_{ji}(t) = w_{ji}(t)\, x_i(t - \tau_{ji}(t))$



- Potenciál neuronu j:

$$u_j(t) = \sum_{i=1}^{p} s_{ji}(t) = \sum_{i=1}^{p} w_{ji}(t)\, x_i\left(t - \tau_{ji}(t)\right)$$

Časový algoritmus zpětného šíření s adaptivním zpožděním

- Výstup neuronu j: $y_j(t) = \varphi (u_j(t) - \tau_{ji}(t))$
- Chybová funkce:

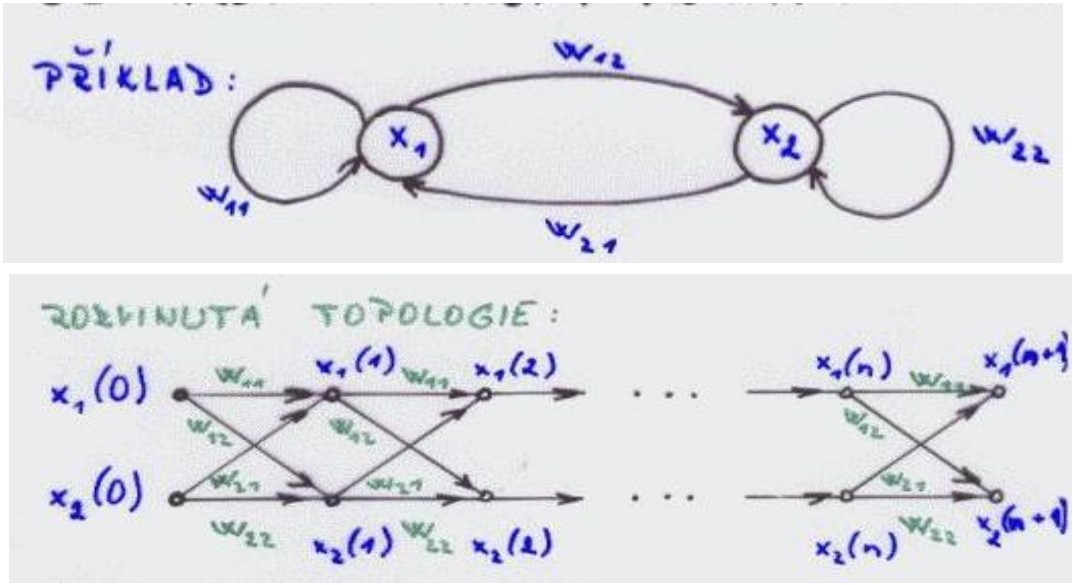
$$E(t) = \frac{1}{2} \sum_{j \in A} \left[d_j(t) - y_j(t) \right]^2$$

=> adaptace obou parametrů ($w_{ji}(t)$ a $\tau_{ji}(t)$) podle:

$$\Delta w_{ji}(t) = -\eta_1 \frac{\partial E(t)}{\partial w_{ji}(t)} \quad \text{a} \quad \Delta \tau_{ji}(t) = -\eta_2 \frac{\partial E(t)}{\partial \tau_{ji}(t)}$$

Algoritmus zpětného šíření v čase

- Učení rekurentních sítí
- Rozšíření standardního algoritmu zpětného šíření
- Odvození rozvinutím časových operací v síti do vícevrstvé dopředné sítě (jejíž topologie s každým krokem narůstá o 1 vrstvu)



Algoritmus zpětného šíření v čase

Odvození adaptačních pravidel:

- Trénovací vzory jsou rozděleny na ETAPY
  - n_0 … začátek etapy
  - n_1 … konec etapy
- Cílová funkce pro učení: $E_{TOTAL}(n_0, n_1) = \frac{1}{2} \sum_{n=n_0}^{n_1} \sum_{j \in A} e_j^2(n)$
  - A … množina indexů j se specifikovaným požadovaným výstupem
  - $e_j(n)$ … chyba na výstupu neuronu j

=> **Etapový algoritmus zpětného šíření v čase**

Algoritmus zpětného šíření v čase

Etapový algoritmus zpětného šíření v čase

- výpočet parciálních derivací $E_{TOTAL}(n_0,n_1)$ podle synaptických vah v síti
- Nejprve se provede dopředný průchod sítí přes interval $[n_0,n_1]$. Uchovávají se veškerá vstupní data, stav sítě (~ synaptické váhy) a požadované výstupy
- Provede se zpětný průchod sítí a spočítají se hodnoty lokálních gradientů:

$$\delta_j(n) = -\frac{\partial E_{TOTAL}(n_0,n_1)}{\partial v_j(n)} \quad ; \quad \forall j \in A, \quad n_0 \le n \le n_1$$

Algoritmus zpětného šíření v čase

Etapový algoritmus zpětného šíření v čase

$$\delta_j(n) = \begin{cases} \varphi'(v_j(n))\, e_j(n) & \text{if } n = n_1 \\ \varphi'(v_j(n)) \left[e_j(n) + \sum_{k \in A} w_{kj}\ \delta_k(n+1) \right] & \text{if } n_0 \le n \le n_1 \end{cases}$$

- φ′(.) … derivace přenosové funkce podle argumentu
- Výpočet $\delta_j(.)$ pak pokračuje od n_1 zpět směrem k n_0; počet provedených kroků odpovídá délce etap
- Poté, co bylo zpětným šířením dopočítáno $\delta_j(n_0+1)$, provede se adaptace váhy w_{ji} podle (η je parametr učení, $x_i(n-1)$ označuje i-tý vstup neuronu j v čase n-1):

$$\Delta w_{ji} = -\eta \frac{\partial E_{TOTAL}(n_0, n_1)}{\partial w_{ji}} = \eta \sum_{n=n_0+1}^{n_1} \delta_j(n)\, x_i(n-1)$$