Aplikace teorie neuronových sítí

Doc. RNDr. Iveta Mrázová, CSc.

Katedra teoretické informatiky
Matematicko-fyzikální fakulta
Univerzity Karlovy v Praze

Neuronové sítě

– Modulární architektury –

Doc. RNDr. Iveta Mrázová, CSc.

Katedra teoretické informatiky
Matematicko-fyzikální fakulta
Univerzity Karlovy v Praze

Robustnost vrstevnatých neuronových sítí

- **Vzhledem ke ztrátě skrytého neuronu**
  - **„1-neuronová“ robustnost**
  - Modifikace chybové funkce pomocí tzv. „robustnostního členu“
  - „prořezávání a duplikace“ ~ zdvojení takových neuronů, jejichž ztráta způsobí největší chybu na výstupu sítě
- **Vzhledem k drobným odchylkám předkládaných vstupních vzorů**
  - **Separační charakteristika**
  - K libovolné vrstevnaté neuronové síti (a dané konečné množině vzorů) lze zkonstruovat „robustnější“ síť s „téměř stejnými“ výstupy (~ ε-ekvivalentní síť)

ε – ekvivalence

- Existuje mnoho vrstevnatých neuronových sítí s požadovaným vstupně / výstupním chováním
- Dvě vrstevnaté neuronové sítě jsou ε-ekvivalentní, jestliže dávají pro všechny vstupní vzory z dané konečné množiny vzorů tentýž výstup s jistou ε-přesností:

$$\left\|\vec{y}_{B_1} - \vec{y}_{B_2}\right\| \le \varepsilon \quad , \qquad \text{kde}$$

$$\left\|\vec{y}_{B_1} - \vec{y}_{B_2}\right\| = \sum_{i=1}^{m} \left(y_{B_1,i} - y_{B_2,i}\right)^2$$

  - ***m*** … počet výstupních neuronů sítí $\boldsymbol{B_1}$, $\boldsymbol{B_2}$

Robustnost vzhledem ke ztrátě skrytého neuronu

1-neuronová robustnost:

- Určena chybou způsobenou ztrátou jednoho skrytého neuronu

$$\varepsilon = E_{d_m} - E_1 = \frac{1}{2N_h} \sum_p \sum_i \sum_k \left(y_{i,k} - d_i \right)^2 - \frac{1}{2} \sum_p \sum_i \left(y_i - d_i \right)^2$$

  - N_h … počet skrytých neuronů
  - d_i … požadovaný výstup pro daný vstupní vzor p
  - y_i … skutečný výstup výstupního neuronu i
  - $y_{i,k}$ ... výstup neuronu i, jestliže chybí skrytý neuron k

Robustnost vzhledem ke ztrátě skrytého neuronu (2)

Další možnosti vyjádření chyby:

- Určena odchylkou výstupu způsobenou ztrátou jednoho skrytého neuronu $E_{y_m} = \frac{1}{2N_h} \sum_p \sum_i \sum_k (y_i - y_{i,k})^2$

  ($E_{y_m} = 0$, jestliže se při ztrátě skrytého neuronu výstup sítě pro trénovací množinu nezmění)

- Nejhorší možné následky

$$E_d = \max_{p,i,k} |y_{i,k} - d_i|$$

$$E_y = \max_{p,i,k} |y_{i,k} - y_i|$$

Zvýšení robustnosti vrstevnatých NS

1) **Modifikací chybové funkce:** $F = E + \rho E_{rob}$

- ρ … nastavitelný parametr ovlivňující vzájemný poměr minimalizované střední kvadratické odchylky skutečných a požadovaných výstupů a robustnosti učené sítě

$$E_{y_m} = \frac{1}{2N_h} \sum_p \sum_i \sum_k \left(f(\xi_i) - f(\xi_{i,k}) \right)^2$$

$$\xi_i = \sum_j x_j w_{ij} \qquad \xi_{i,k} = \xi_i - x_k w_{ik}$$

- x_j … výstup skrytého neuronu j
- w_{ij} … váha mezi skrytým neuronem j a výstupním neuronem j
- $f(\xi)$ … přenosová funkce (sigmoida)

Zvýšení robustnosti vrstevnatých NS (2)

TRIK: minimalizace $E_{y'_m}$ namísto minimalizace E_{y_m}

$$E_{y'_m} = \frac{1}{2N_h} \sum_p \sum_i \sum_k \left(\xi_i - \xi_{i,k}\right)^2 = \frac{1}{2N_h} \sum_p \sum_i \sum_j \left(x_j w_{ij}\right)^2$$

- **Pro výstupní vrstvu**

$$\frac{\partial E}{\partial w_{ij}} = (y_i - d_i) y_i (1 - y_i) x_j + \frac{\rho}{N_h} x_j^2 w_{ij}$$

- **Pro skrytou vrstvu**

$$\frac{\partial E}{\partial w_{il}} = \sum_i \left(\left(\frac{\partial E}{\partial y_i} y_i (1 - y_i) + \frac{\rho}{N_h} w_{ij} x_j \right) w_{ij} \right) x_j (1 - x_j) z_l$$

z_l … aktivita vstupního neuronu l

Zvýšení robustnosti vrstevnatých NS (3)

2) Prořezáváním a duplikací

- Adaptace sítě pomocí standardního algoritmu zpětného šíření
- **Otestování robustnosti sítě**
  - Identifikace takového skrytého neuronu, jehož ztráta by způsobila největší změnu robustnosti
  - „zdvojení"tohoto neuronu
  - „rozpůlit" váhy obou „kopií" vedoucí k výstupní vrstvě
  - („kopie" nahradí některý z ostatních skrytých neuronů)

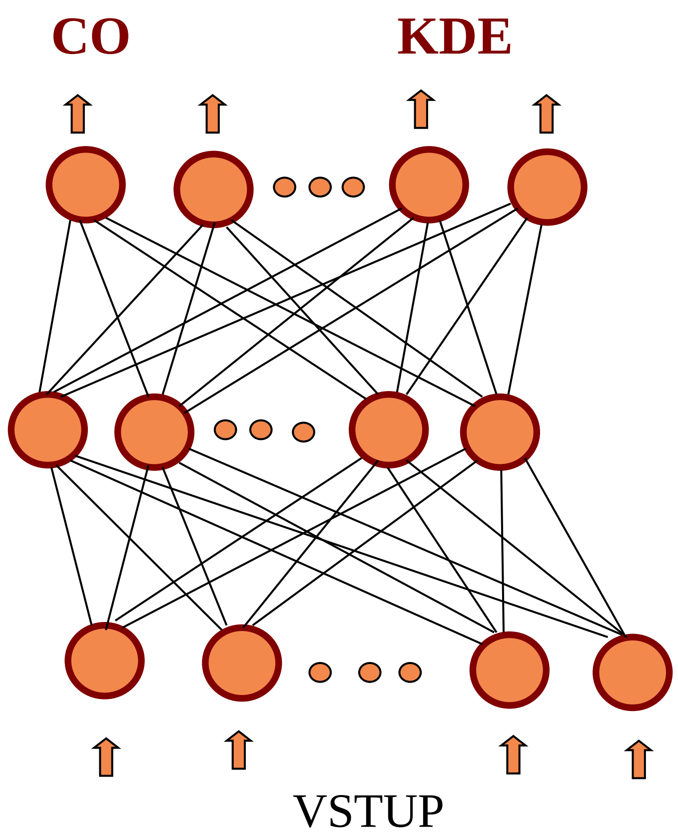
„Prořezávání a duplikace“

Výsledkem jsou:

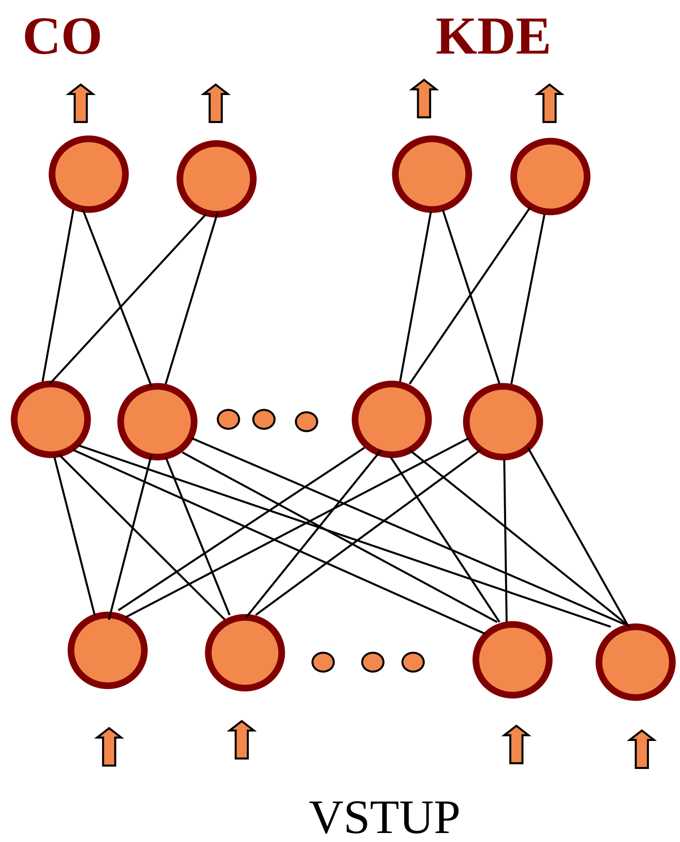
- robustnější přesnější sítě
- malá výpočetní složitost
- lepší generalizace
- omezení problémů souvisejících s výskytem lokálních minim

Modulární struktura BP-sítí

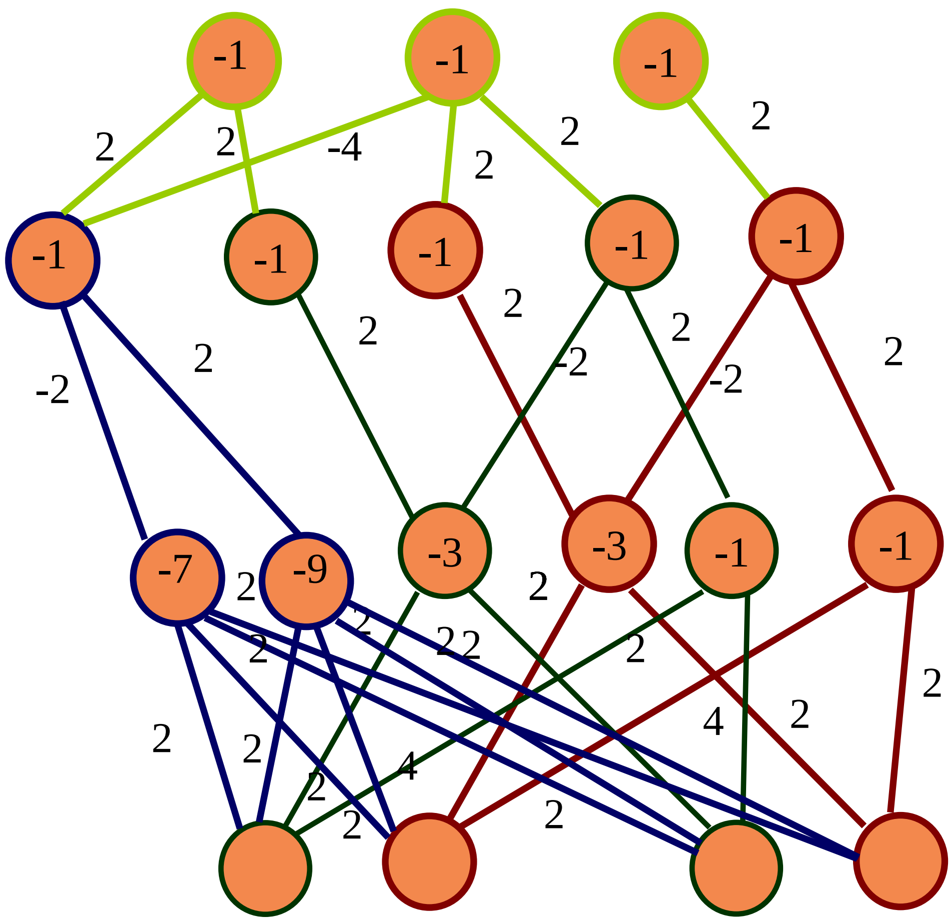
Jednoduchá vrstevnatá neuronová síť



Modulární vrstevnatá neuronová síť



Modulární struktura BP-sítí: binární sčítání



A BP-network detecting those input patterns where the carry over two orders is necessary to add two binary 2-bit numbers (01 + 11 and 11 + 01)

A BP-network for the binary addition of two binary 1-bit numbers

A BP-network for the binary addition of two binary 1-bit numbers

Modulární struktura BP-sítí

- Rozdělení úlohy do jednotlivých podúloh
- Návrh a vytvoření modulární architektury
  - Strategie pro extrakci ε-ekvivalentních modulů BP-sítí
    - Eliminace přebytečných skrytých a/nebo vstupních neuronů
    - Vhodná pro „již natrénované“ sítě
    - Kompromis mezi požadovanou přesností extrahovaného modulu a jeho optimální architekturou
- Vzájemná komunikace mezi jednotlivými moduly
  - Paralelní a sériová kompozice BP-sítí

Definice modulu BP-sítě

Nechť $\boldsymbol{B}$ je BP-síť s $\boldsymbol{l}$ skrytými vrstvami.

BP-síť $\boldsymbol{M}$ je modulem $\boldsymbol{B}$, jestliže obě sítě mají stejný počet vrstev a jestliže je každá skrytá vrstva L_k^M sítě $\boldsymbol{M}$ podmnožinou skryté vrstvy L_k^B sítě $\boldsymbol{B}$:

$$L_k^M \subseteq L_k^B \quad ; \quad 1 \leq k \leq l$$

Věta o extrakci modulů

Nechť S je konečná množina vstupních vzorů $\mathbf{x}$ a nechť B je BP-síť s množinou H_l neuronů poslední skryté vrstvy, $|H_l| > 1$.

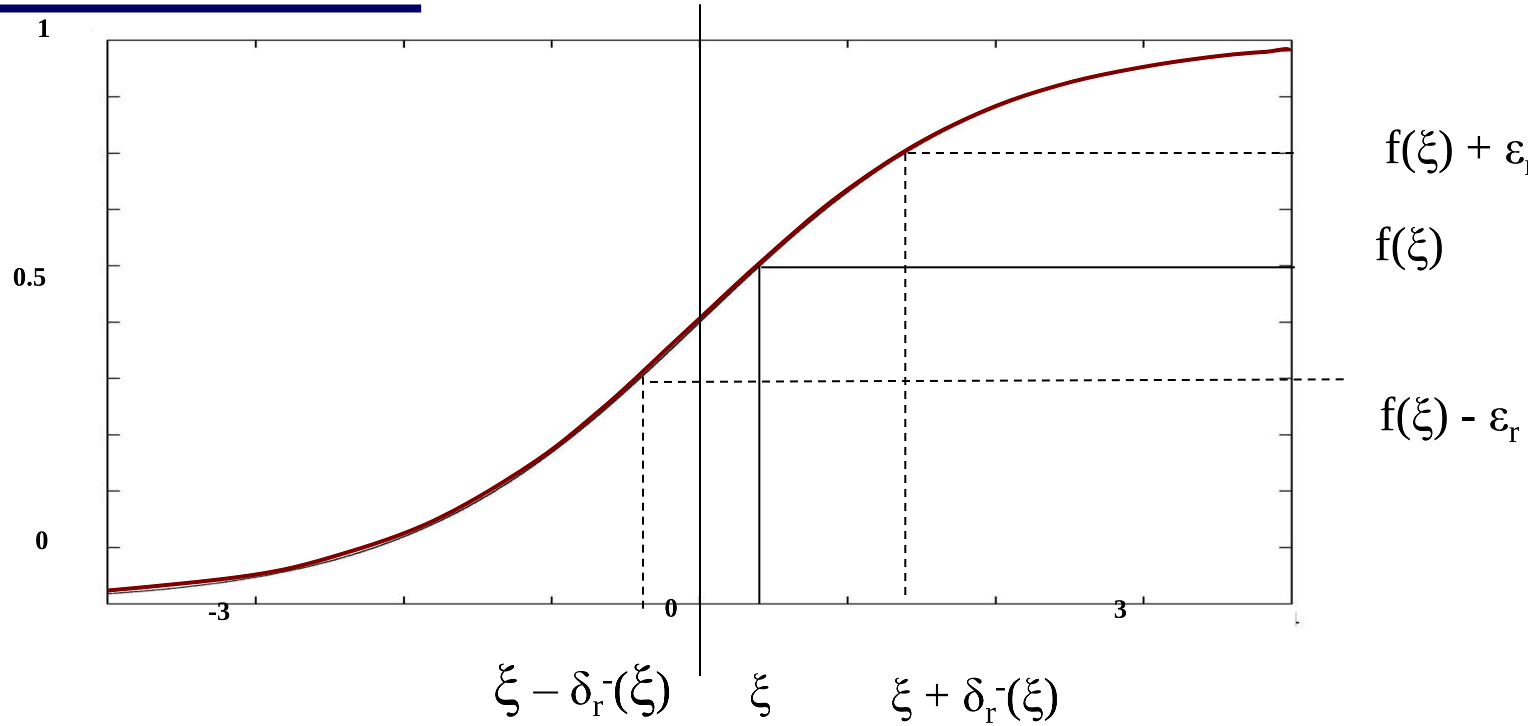
Potom k libovolnému $\varepsilon > 1$ existuje taková konstanta $K_\varepsilon > 0$, že platí: Jestliže existuje v l –té skryté vrstvě neuron k takový, že:

$$(\forall \mathbf{x} \in S)(\forall i \in O); \quad |w_{ki} z_k| \leq K_\varepsilon$$

potom je BP-síť M^k, která je totožná se sítí B bez neuronu k, ε-ekvivalentní modul B, ($M^k \sim_\varepsilon^S B$) .

Věta o extrakci modulů (2)

Idea důkazu:



Věta o extrakci modulů (3)

<u>Idea důkazu</u> (pokračování):

ξ

- „povolená odchylka“ potenciálu $\delta_r^-(\xi)$ je v tomto případě menší než „povolená odchylka“ potenciálu $\delta_r^+(\xi)$
- Potenciál by se měl změnit „směrem k dělicí nadrovině“
- Změněný potenciál by měl zachovat umístění vzorku v témže poloprostoru

Věta o extrakci modulů (4)

Idea důkazu (pokračování):

ξ

- „povolené odchylky" potenciálu by měly být nezávislé na jednotlivých vzorech.
- Eliminace neuronu, pokud je $\forall x \in S$ $|\, w_{ki} z_k \,|$ malé
  - $\delta_r / (l + 1)$ pro poslední skrytou vrstvu
  - $l\, \delta_r / (l + 1)$ pro ostatní skryté vrstvy

Neuronové sítě s modulární strukturou

- Adaptivní směsi lokálních neuronových sítí (Jacobs & Jordan)
- Selektorové sítě s množinou verzí (Partridge & Sharkey)
- Kombinační sítě s dynamickou volbou klasifikátoru => HYBRIDNÍ MODEL (Lee, Srihari & Nowlan)

Neuronové sítě s modulární strukturou (2)

Důležité vlastnosti modelů:

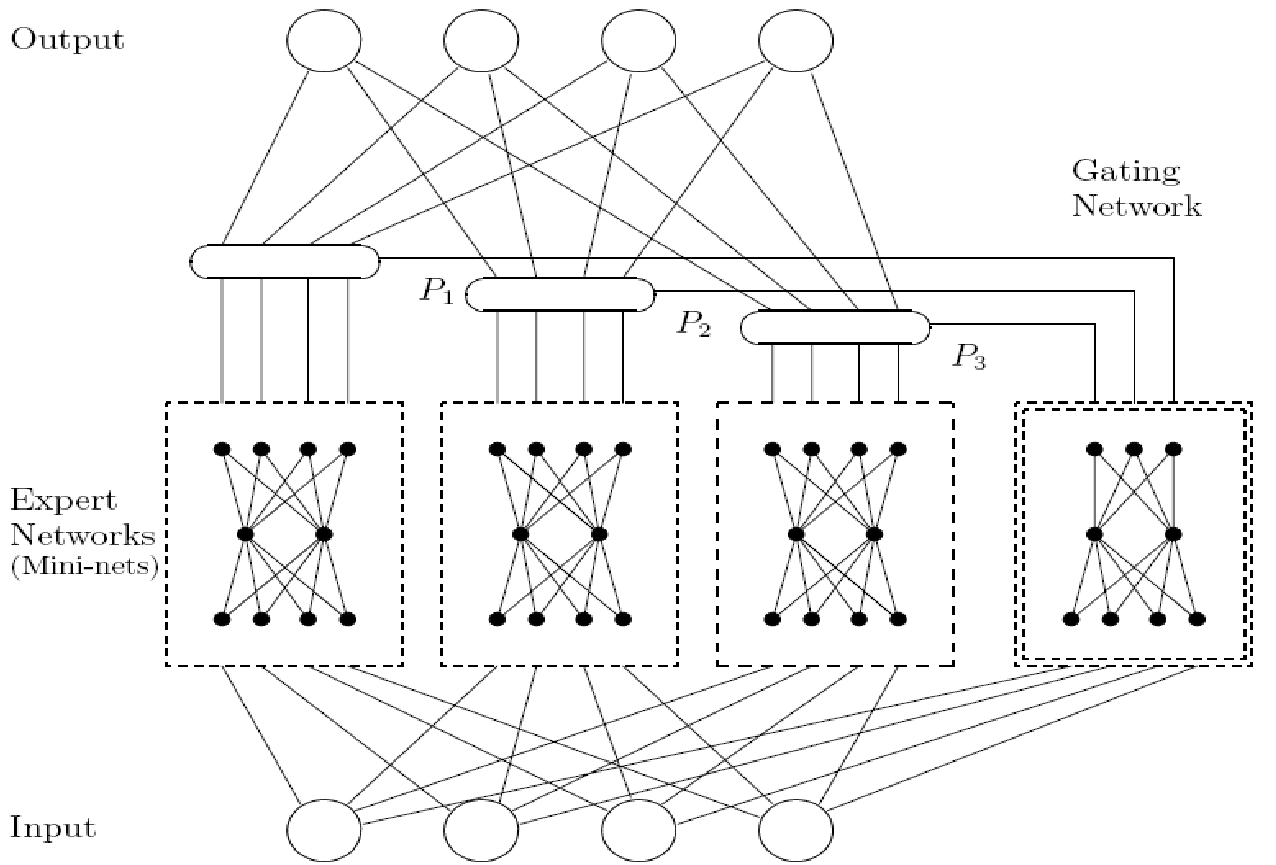
- Rychlost a konvergence procesu učení
- Optimalizace architektury
- Robustnost a generalizace
- „užitečná různorodost" jednotlivých modulů (~ „useful diversity")

Adaptivní směsi lokálních neuronových sítí

Soustava lokálních sítí s řídicí sítí:

- Lokální sítě jsou vrstevnaté neuronové sítě typu zpětného šíření
- Všechny lokální sítě mají stejné vstupy a stejný počet výstupních neuronů
- Řídicí síť je vrstevnatá síť s výstupy P_j a stejnými vstupy, jako mají lokální sítě
- Výstupy systému odpovídají $\sum_j P_j \, y^{(j)}$

Adaptivní směsi lokálních neuronových sítí (2)



Adaptivní směsi lokálních neuronových sítí (3)

- Nechť se soustava skládá z t lokálních sítí N_i s vahami $W^{(i)}$ a prahy $\vartheta^{(i)}$; $i=1,\dots,t$; a jedné řídicí sítě N_g s vahami $W^{(g)}$ a prahy $\vartheta^{(g)}$
- Dále nechť je pro daný trénovací vzor $(\vec{x}, \vec{d}\,)$ skutečný výstup lokálních sítí $\vec{y}^{(i)} = \left(y_1^{(i)},\dots,y_n^{(i)}\right); 1 \le i \le t$ a skutečný výstup řídicí sítě $\vec{y}^{(g)} = \left(y_1^{(g)},\dots,y_n^{(g)}\right)$

$$y_j^{(i)} \in (0,1); \quad \forall\, j \in \{1,\dots,n\} \wedge i \in \{1,\dots,t\}$$

$$y_j^{(g)} \in (0,1); \quad \forall\, j \in \{1,\dots,m\}$$

Adaptivní směsi lokálních neuronových sítí (4)

- Pro koeficienty p_i by mělo platit:

$$p_i = \frac{y_i^{(g)}}{\sum_{j=1}^{t} y_j^{(g)}}; \; i = 1, \dots, t \qquad \left(p_i \in (0,1) \wedge \sum_{i=1}^{t} p_i = 1 \right)$$

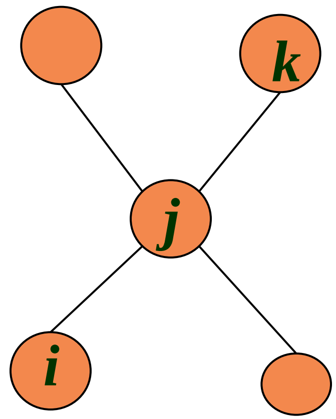
- Cílová funkce: $E = \sum_p E_p$

$$E_p = \frac{1}{2} \sum_{i=1}^{t} \left(p_i \left(\vec{y}^{(i)} - \vec{d} \right) \right)^2 = \frac{1}{2} \sum_{i=1}^{t} \left(\sum_k g_k^{(i)2} \right)$$

$$g_k^{(i)} = \begin{cases} p_i \left(y_k^{(i)} - d_k \right) & k \text{ je výstupní neuron} \\ 0 & \text{jinak} \end{cases}$$

Adaptivní směsi lokálních neuronových sítí (5)

Proces učení a adaptační pravidla:



- Adaptace vah a prahů lokálních sítí podle:

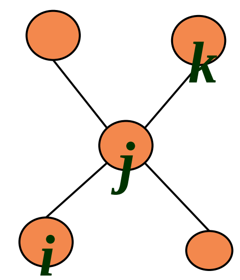
$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial \vartheta_j} y_i$$

$$\frac{\partial E}{\partial \vartheta_j} = f'(\xi_j)\left(g_j + \sum_k \frac{\partial E}{\partial \vartheta_k} w_{jk}\right)$$

$$g_j^{(i)} = \begin{cases} p_i \left(y_j^{(i)} - d_j\right) & j \text{ je výstupní neuron} \\ 0 & \text{jinak} \end{cases}$$

Adaptivní směsi lokálních neuronových sítí (6)

- Adaptace vah a prahů řídicí sítě podle:



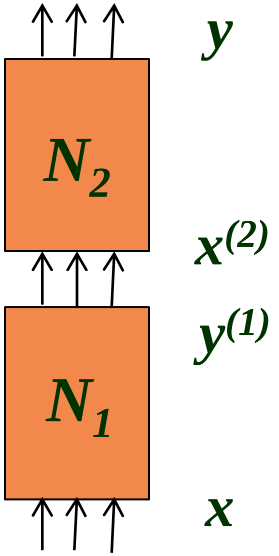
$$\frac{\partial E}{\partial w_{ij}^{(g)}} = \frac{\partial E}{\partial \vartheta_j^{(g)}} \, y_i^{(g)}$$

$$\frac{\partial E}{\partial \vartheta_j^{(g)}} = f'\left(\xi_j^{(g)}\right)\left(g_j^{(g)} + \sum_k \frac{\partial E}{\partial \vartheta_k^{(g)}} w_{jk}^{(g)} \right)$$

$$g_j^{(g)} = \begin{cases} \left(\frac{1}{2}\left(\vec{y}^{(j)} - \vec{d}\right)^2 - E \right) \cdot \left(\sum_i y_i^{(g)} \right)^{-1} & j \text{ je výstupní neuron} \\ 0 & \text{jinak} \end{cases}$$

Adaptivní směsi lokálních neuronových sítí (7)

- tendence používat ke klasifikaci předkládaných vzorů vždy jen jednu síť
- **Sériová kompozice dvou soustav lokálních sítí s řídící sítí:**



- Přenos chybových členů mezi nejnižšími skrytými neurony N_2 (její vstupní neurony se nyní neuvažují) a výstupními neurony N_1

  $\vartheta_i^{(k)}$… práh i-tého výstupního neuronu k-té lokální sítě z N_1

Adaptivní směsi lokálních neuronových sítí (8)

$$\frac{\partial E}{\partial \vartheta_i^{(k)}} = f'\left(\xi_i^{(k)}\right)\frac{\partial E}{\partial y_i^{(k)}} = f'\left(\xi_i^{(k)}\right)\sum_j \sum_l \frac{\partial E}{\partial y_l^{(j)}} \frac{\partial y_l^{(j)}}{\partial y_i^{(k)}}$$

$$= f'\left(\xi_i^{(k)}\right)\sum_j \sum_l \frac{\partial E}{\partial y_l^{(j)}} f'\left(\xi_l^{(j)}\right)\frac{\partial y_l^{(j)}}{\partial y_i^{(k)}}$$

$$= f'\left(\xi_i^{(k)}\right)\sum_j \sum_l \frac{\partial E}{\partial \vartheta_l^{(j)}} \frac{\partial \xi_l^{(j)}}{\partial y_i^{(k)}}$$

j … index přes všechny lokální sítě a řídicí síť z $\boldsymbol{N_2}$

l … index neuronů nad vstupy ***j***

Adaptivní směsi lokálních neuronových sítí (9)

$$\frac{\partial \xi_l^{(j)}}{\partial y_i^{(k)}} = w_{il}^{(j)} p_k^{(1)}$$

$$\frac{\partial E}{\partial \vartheta_i^{(k)}} = f'\left(\xi_i^{(k)}\right) p_k^{(1)} \sum_j \sum_l \frac{\partial E}{\partial \vartheta_l^{(j)}} w_{il}^{(j)}$$

pro lokální sítě

$$\frac{\partial E}{\partial \vartheta_i^{(g)}} = \frac{1}{\sum_n y_n^{(g)}} f'\left(\xi_i^{(g)}\right) \sum_j \sum_l \frac{\partial E}{\partial \vartheta_l^{(j)}} \sum_m w_{ml}^{(j)} \left(y_m^{(i)} - y_m^{(j)}\right)$$

pro řídicí síť

m … index přes všechny lokální sítě z N_1

y^i_m … ***m***-tý výstup ***i***-té lokální sítě z N_1

y^j_m … ***m***-tý vstup ***j***-té lokální/řídicí sítě z N_2

Modulární neuronové sítě: inverzní kinematika robota

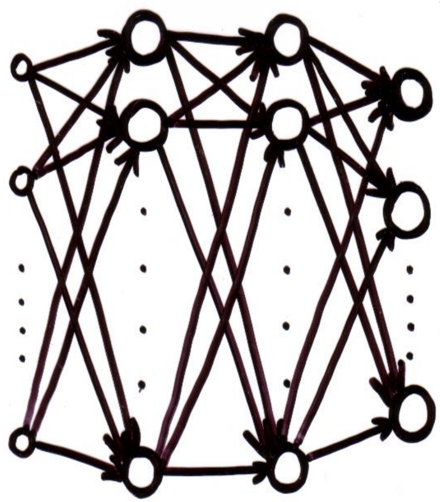
- Kinematická rovnice: $\vec{x} = \vec{f}(\theta)$
  - Převod informace (kartézské souřadnice ($\vec{x}$) => polární ($\boldsymbol{\theta}$))
- Požadavky na rychlost: $\dot{\boldsymbol{x}} = \boldsymbol{J}\,(\,\boldsymbol{\theta}\,)\,\boldsymbol{\theta}^{\bullet}$
- Prvky $\boldsymbol{J}$: $\partial x_i / \partial \theta_j \qquad \forall i, j$
- Řešení: $\boldsymbol{\theta}^{\bullet} = \boldsymbol{J}^{-1}\,(\,\boldsymbol{\theta}\,)\,\dot{\boldsymbol{x}}$
- Problémy:
  - Inverze matice $\boldsymbol{J}$
  - Znalost přesných kinematických parametrů robota, případně jejich odhad

Modulární neuronové sítě: inverzní kinematika robota (2)

„Naivní“ architektura neuronové sítě pro inverzní kinematiku

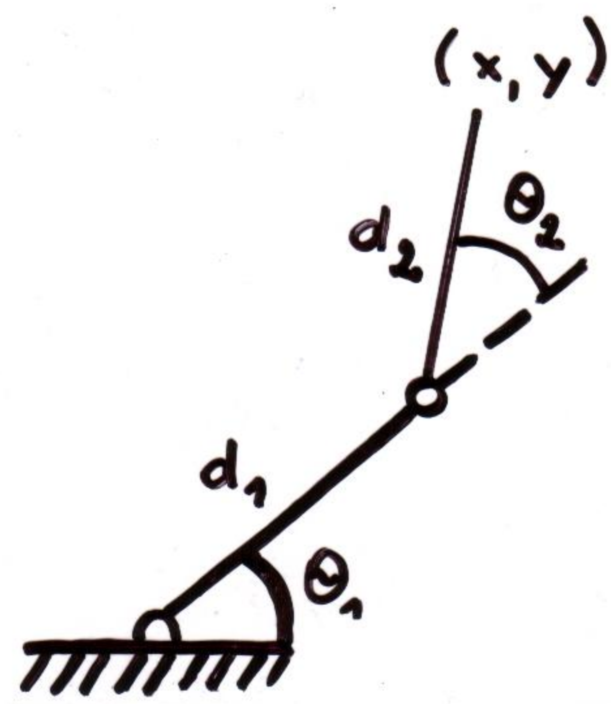
n úhlů

n kartézských souřadnic (rychlost)



n úhlových rychlostí

Rameno se 2 stupni volnosti



Simulace ramene se 2 stupni volnosti

- **Kartézské souřadnice (x , y):**

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} d_1 \cos\theta_1 + d_2 \cos(\theta_1 + \theta_2) \\ d_1 \sin\theta_1 + d_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$$

- **Rovnice pro vyjádření rychlosti:**

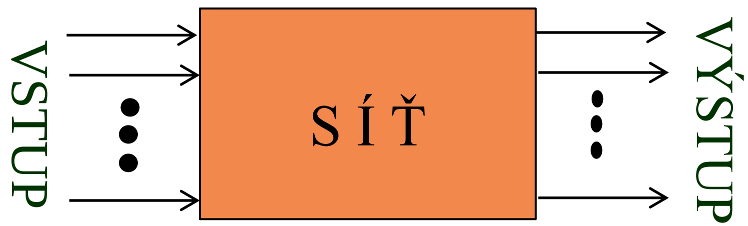
$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -d_1 \sin\theta_1 - d_2 \sin(\theta_1 + \theta_2) & -d_2 \sin(\theta_1 + \theta_2) \\ d_1 \cos\theta_1 + d_2 \cos(\theta_1 + \theta_2) & d_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

- **Analytické řešení:**

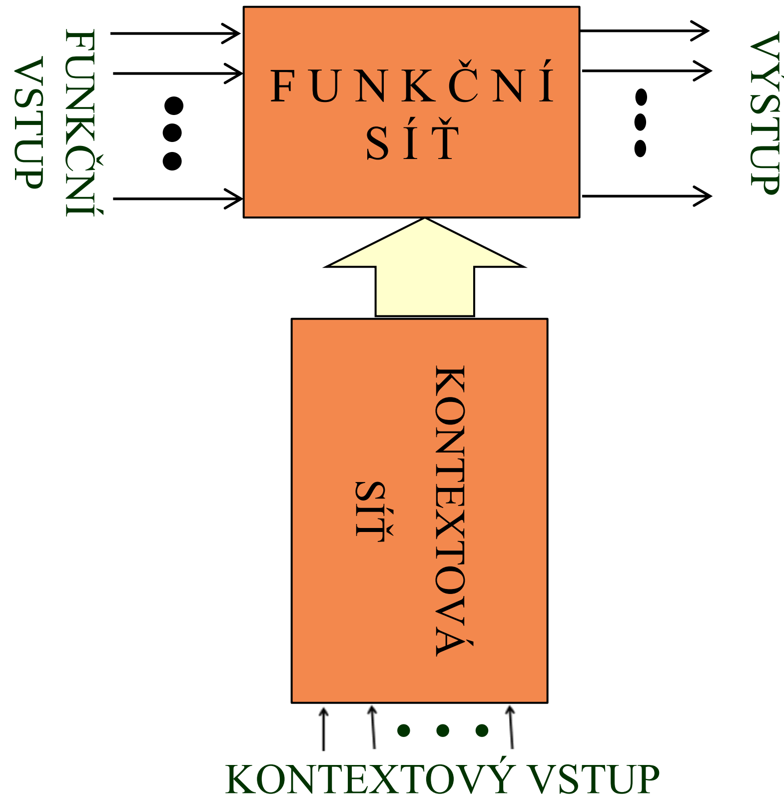
$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \dfrac{\cos(\theta_1 + \theta_2)}{d_1 \sin\theta_2} & \dfrac{\sin(\theta_1 + \theta_2)}{d_1 \sin\theta_2} \\ -\dfrac{\cos\theta_1}{d_2 \sin\theta_2} - \dfrac{\cos(\theta_1 - \theta_2)}{d_1 \sin\theta_2} & -\dfrac{\sin\theta_1}{d_2 \sin\theta_2} - \dfrac{\sin(\theta_1 - \theta_2)}{d_1 \sin\theta_2} \end{bmatrix} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

Modulární neuronové sítě: inverzní kinematika robota (3)

„Standardní“ neuronová síť



Kontextová síť



Kontextové sítě

Inverzní kinematický problém:

- Vstup sítě: $\boldsymbol{\theta}, \dot{\boldsymbol{x}}$
- Výstup sítě: $\dot{\boldsymbol{\theta}}$
- Příklad:
  - robot se 6 stupni volnosti
  - (12 vstupních a 6 výstupních neuronů)

  => velká časová náročnost

Kontextové sítě (2)

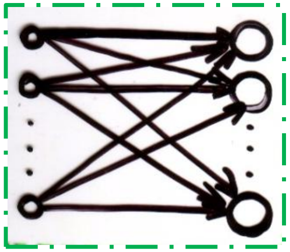
Architektura a vlastnosti:

- Dopředné sítě (feed-forward)
- Algoritmus zpětného šíření
- Potenciál neuronu j: $x_j = \sum_i w_{ij} y_i + b_j$
  - w_{ij} … váha mezi neuronem i a j
  - x …… potenciál neuronu
  - y …… výstup neuronu
  - b_j ….. Práh neuronu j
- Výstup neuronu j, y_j, pomocí sigmoidy

Kontextová síť pro inverzní kinematiku robota (4)

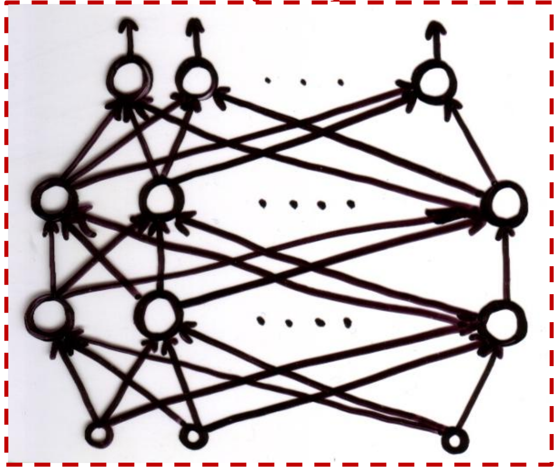
FUNKČNÍ SÍŤ

n kartézských souřadnic (rychlost)



n úhlových rychlostí

n^2 výstupů



n úhlů

KONTEXTOVÁ SÍŤ

Kontextové sítě (3)

Přenos informace mezi kontextovou a funkční sítí:

- Potenciál neuronu j funkční sítě:

$$x_j = \sum_i w_{ij}\, y_i = \sum_i g_l(y_l)\, y_i$$

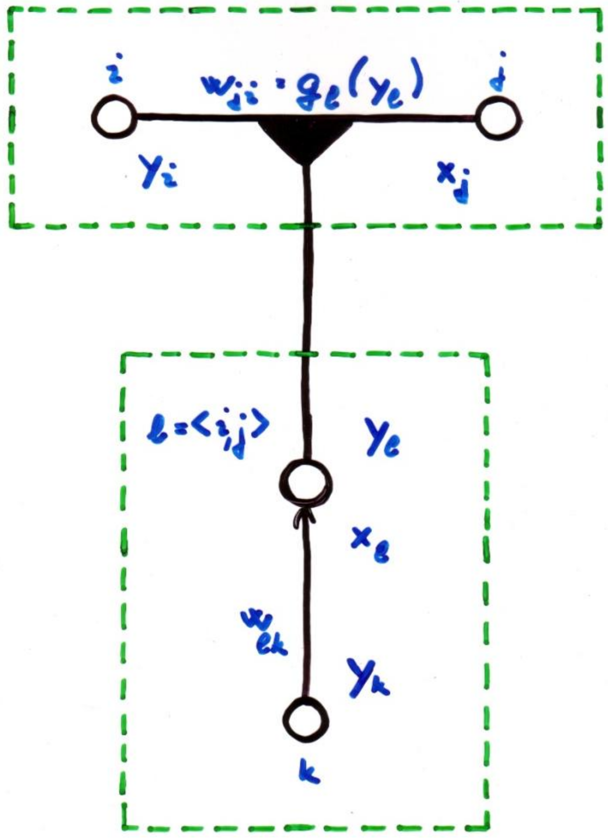
  - w_{ij} … váha (funkční sítě) mezi neuronem i a j
  - l …… neuron ve výstupní vrstvě kontextové sítě (určují váhy funkční sítě)
  - $g(.)$ … přenos informace mezi neuronem l kontextové sítě a váhou w_{ij} funkční sítě
- Speciální případ: g je lineární funkce

$$g(x) = a_l\, x\,;\;\; a_l \neq 0 \text{ je konstanta} \;\Rightarrow\; x_j = \sum_i a_l\, y_l\, y_i$$

Kontextová síť pro inverzní kinematiku robota (5)

Propojení kontextové a funkční sítě:

FUNKČNÍ SÍŤ



KONTEXTOVÁ SÍŤ

Kontextové sítě (4)

Přenos informace mezi kontextovou a funkční sítí
(pro n stupňů volnosti):

- Volba kontextu: funkční síť by měla reprezentovat lineární funkci => menší složitost
- Kontextový vstup:
  - konfigurace ramene (dána vektorem $\boldsymbol{\theta}$)
- Funkční vstup: $\vec{x}$
  - Pro daný kontext $\boldsymbol{\theta}$ – lineární vztah $\boldsymbol{\theta}\cdot$ a $\vec{x}$
  - Funkční síť tedy reprezentuje lineární funkce

Kontextové sítě (5)

Přenos informace mezi kontextovou a funkční sítí
(pro n stupňů volnosti):

- **Funkční síť:**
  - ***n*** vstupních a ***n*** výstupních neuronů
  - Výstupní neurony nemají práh => výstup $f(x) = x$
  - Síť má n^2 vah; ty odpovídají $J^{-1}(\theta)$
- **Kontextová síť:**
  - n^2 "podsítí"; každá z nich odpovídá skalární funkci pro položky J
  - ***n*** vstupních neuronů společných pro všechny „podsítě" (se dvěma skrytými vrstvami)
  - Sigmoidální přenosové funkce

Kontextové sítě (6)

Přenos informace mezi kontextovou a funkční sítí
(pro n stupňů volností):

- **Zjednodušení:**
  - Převod zobrazení $n \rightarrow n^2$ na n^2 funkcí $n \rightarrow 1$
  - Jasnější funkce skrytých neuronů
  - Paralelní učení jednotlivých funkcí
  - S rostoucím počtem stupňů volnosti roste pouze počet funkcí