

Algoritmy a struktury neuropočítačů

ASN – P5

Vícevrstvé sítě – dopředné a Elmanovy, MLNN s učením BPG

- dopředné propojení a šíření signálu
- heteroasociativní sítě s učitelem (je požadována znalost cílových hodnot)
- **off-line učení** – synaptické váhy a prahy pro všechny tréninkové vzorky,
batch training
 - modifikace vah až po přivedení všech vzorků do sítě,
 - počítá se aktuální gradient chyby E pro úplnou množinu tréninkových vzorků
- **on-line učení** - váhy a prahy jsou modifikovány bezprostředně po přivedení každého tréninkového vzorku,
incremental training
 - proces konverguje stochasticky k lokálním minimům, nezaručuje dosažení absolutního minima

Princip:

Iterativní proces, hledání minima chyby mezi skutečnou (výstupní) hodnotou a požadovanou hodnotou.

- 1) Na vstup NS je přiváděn vektor resp. matice vstupních parametrů -**
 - číselné hodnoty pro konkrétní hodnoty fyzikálních veličin
 - kategoriální data (přiřazení kategorie určité vlastnosti),
 - číselná hodnota udává váhu dané vlastnosti vzhledem k ostatním.
- 2) Po průchodu sítí je spočítán výstup z každého neuronu, výsledek je porovnán s požadovanou hodnotou.**
- 3) Je spočítána chyba - zpětně se přepočítává do předchozích vrstev,**
 - synaptické váhy představující paměť jsou opraveny.
- 4) Do opravené sítě je znovu přiveden vstupní vektor resp. matice,**
 - proces se opakuje.

Nevýhoda: velká citlivost na relevantnost vstupních dat a na inicializaci synaptických vah.

minimalizace rozhodovací funkce (energetická funkce, *cost function*) gradientní metoda

- **rozhodovací funkce** - střední kvadratická chyba mezi požadovaným a skutečným výstupem (*MSE – mean squared error*)
 - suma kvadratických chyb (*SSE – sum squared error*)

tvár plochy chybové funkce závisí - na parametrech sítě (váhy, přenos. funkce)
- na vstupních vektorech

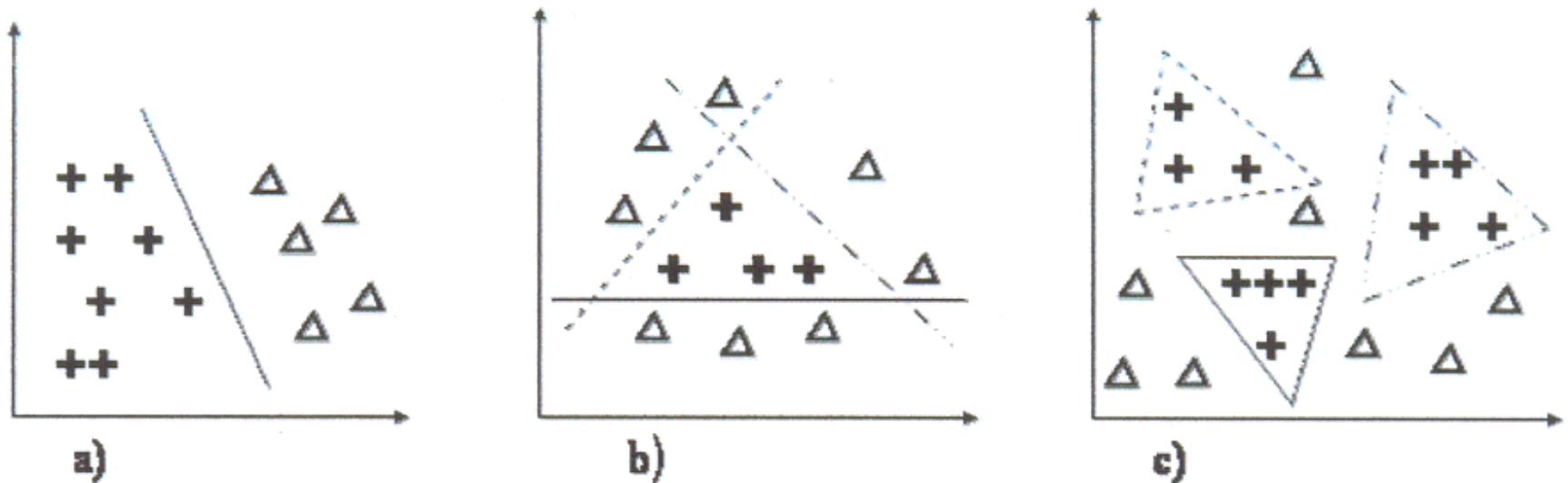
předzpracování vstupů sítě - zmenšení vstupní dimenze vektorů
- potlačení nepodstatných vlastností vstupních vektorů
- zdůraznění podstatných vlastností

- **aktivační funkce** - spojitá diferencovatelná nelineární monotónně rostoucí funkce (např. sigmoida nebo hyperbolická tangenta)

Vícevrstvé neuronové sítě

(*Multilayer NN, MLNN*)

- vstupní data do MLNN nejsou omezena požadavkem na lineární separovatelnost
- možnost klasifikace do více tříd



Separace podmnožin dat pomocí

a) jednoduchého perceptronu

b) dvojvrstvé NS bez skryté vrstvy

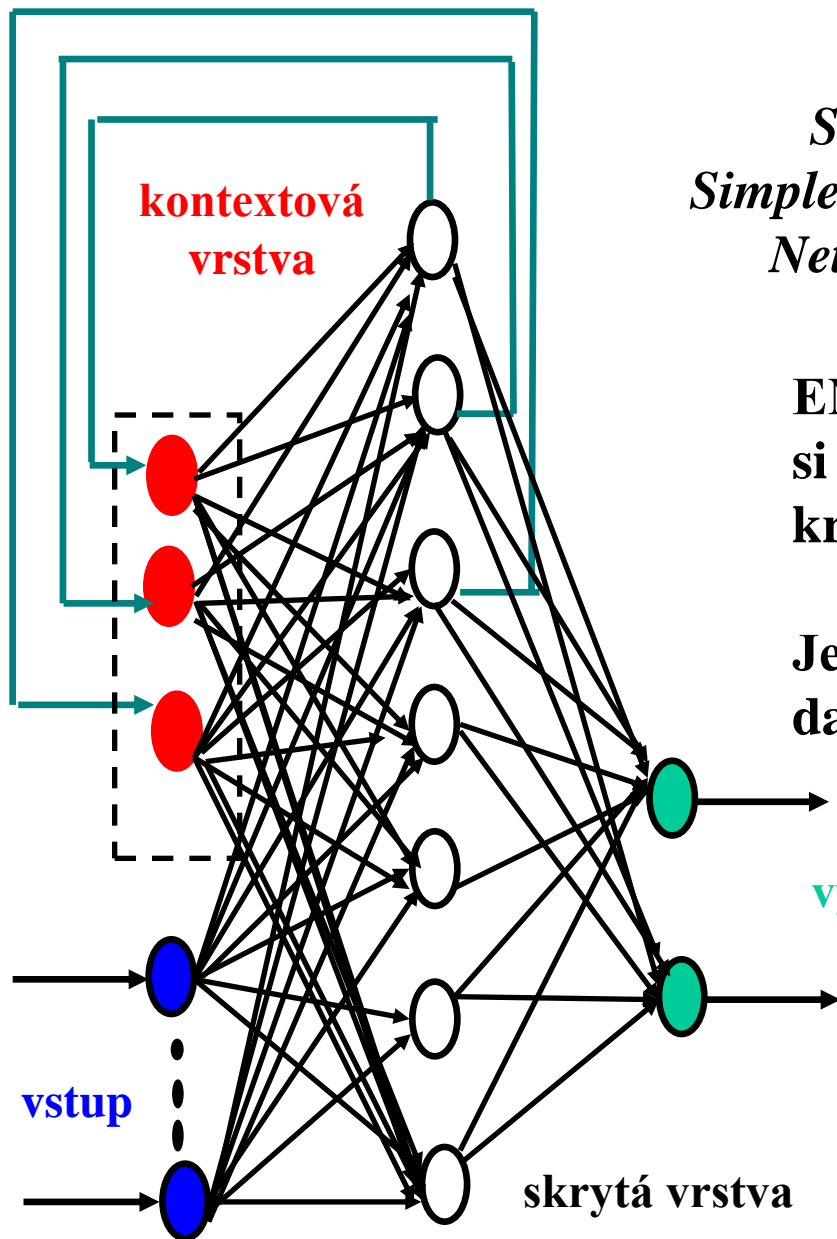
c) třívrstvé NS (s 1 skrytou vrstvou)

MLNN – univerzální aproximátory - jsou schopny aproximovat jakoukoliv měřitelnou funkci s požadovanou přesností

- **úspěšnost natrénování závisí na :** výběru algoritmu učení, dostatečném počtu skrytých neuronů, na existujícím vztahu mezi vstupními daty a požadovanými hodnotami
- **koeficient konvergence - roste s počtem skrytých neuronů**

rekurentní vazba

Elmanova částečně rekurentní NS



SRN
Simple Recurrent
Network

EN
Elman Networks

EN sítě mají schopnost zapamatovat si data vstupující do NS v předchozím kroku učení. Sít' "rozumí" kontextu.

Je vhodná pro úlohy, ve kterých jsou data závislá prostorově i časově (zpracování řečového signálu).

výstup

vstup

skrytá vrstva

ELMAN, J. L. **Distributed Representations, Simple Recurrent Networks, and Grammatical Structure.** *Machine Learning* 7, 1991, 195–225.

zpětná vazba - určitý typ vnitřní paměti

- umožňuje zpracovávat i vstupní posloupnosti variabilní délky
- pokročilejší varianty rekurentních NS – např. pro aplikace na syntaktickou analýzu přirozených jazyků

- vstupní data do MLNN nejsou omezena požadavkem lineární separovatelnosti
- možnost klasifikace do více tříd

Funkce Elmanovy neuronové sítě (ENN):

- 1) Do ENN vstupují současně vstupní data a výstupní data z předchozích stavů. Informace z předchozích stavů jsou přiváděny do tzv. **kontextové** (rekurentní) vrstvy.

- 2) Kontextová vrstva = část vstupní vrstvy,
propojení ze skryté do vstupní vrstvy
rekurence HIF (*Hidden-Input Feedback*)
- 3) Kontextová vrstva = část vstupní vrstvy,
propojení z výstupní do vstupní vrstvy
rekurence OIF (*Output-Input Feedback*)
- 4) Kontextová vrstva = část skryté vrstvy,
propojení z výstupní do skryté vrstvy
rekurence OHF (*Output-Hidden Feedback*)

Skrytá vrstva - obecně libovolný počet neuronů

!!! dobře natrénovaná síť řešící reálné problémy má mít větší počet skrytých neuronů, než MLNN **!!!**

Např. : úlohy rozpoznání řeči - kolem 150 – 200 neuronů.

Volba aktivačních funkcí – žádné speciální požadavky.

obvykle: rostoucí nelineární funkce

Zobecněná Elmanova neuronová síť - neuronová síť se zpětným šířením chyby v závislosti na čase

Back Propagation Through Time – BPTT

Werbos, P.J.: Backpropagation through time: What it does and how to do it. Proc. of the IEEE, vol.78, No 10, october 1990

Algoritmus - umožňuje řešit časové závislosti změny dat,
- používá se pro zpracování číslicových signálů,
- při rozpoznání řeči
- při predikci časových řad.

!!! Pracuje jen v dávkovém módu, je náročná na paměť !!!

Základní algoritmus učení zpětného šíření chyby BPG

Error Back-propagation Algorithm *Back-propagation of Gradient Algorithm*

- Modifikace vah ve směru negativního gradientu chybové funkce.

W_k je vektor vah a prahů (v daném časovém okamžiku – *current vector*),
 g_k je gradient a α_k je rychlost učení

- Učení je velmi pomalé.

Postup

- 1) Vytvoření trénovací a testovací množiny vzorů.

2) Vytvoření množiny požadovaných (cílových) hodnot.

- a) *Je-li UNS použita jako klasifikátor, pak všechny hodnoty požadovaných výstupů se blíží nule, vyjma hodnoty výstupu odpovídajícího třídě příslušného vstupu (hodnota výstupu je 1).*
- b) *Je-li UNS použita pro aproximaci funkcí, mohou být požadované hodnoty vyjádřeny analogovou hodnotou.*

3) Volba topologie sítě.

4) Inicializace vah a prahů v jednotlivých vrstvách.

Může být explicitní (méně časté) nebo malými náhodnými čísly (vhodné zejména při malé znalosti problematiky).

5) Trénování: výpočet výstupní hodnoty, adaptace vah a prahů.

Pro každou uspořádanou dvojici (X_k, Y_k) , $k = 1, 2, \dots, n$ platí, že

- a) hodnoty vzorů X_k vstupují do neuronů vrstvy L_1 a aktivují hodnotami vah a prahů výstupy z vrstvy w_{ih} matice W_1 . Spočítají se výstupy ze skryté vrstvy L_2**

$$h_i = \sigma \left(\sum_{h=1}^n x_h w_{ih}^1 + \Theta_i^1 \right) \quad i = 1, 2, \dots, S_1$$

h_i je aktivní hodnota i-tého neuronu ve vrstvě L_1 ,

Θ_i je i-tá prahová hodnota v i-tém neuronu ve vrstvě L_1 ,

$$\sigma(u) = \left(1 + e^{-u/T} \right)^{-1}$$

je sigmoida,

$1/T$ je strmost funkce

- b) aktivní výstupy z vrstvy L_2 , se filtrují prostřednictvím matice vah W_2 výstupní vrstvy L_3 podle vztahu**

$$y_j = \sigma \left(\sum_{i=1}^p h_i w_{ji} + \Theta_j^2 \right) \quad j = 1, 2, \dots, S_2$$

c) chyba na výstupu mezi aktuální a požadovanou hodnotou se spočítá podle vztahu

$$E_j = y_j (1 - y_j) (t_j^k - y_j) \quad j = 1, 2, \dots, S_2$$

d) vypočítá se chyba každého neuronu ve skryté vrstvě z každé chyby ve výstupní vrstvě

$$e_i = h_i (1 - h_i) \sum_{j=1}^{S_2} w_{ji} E_j, \quad i = 1, 2, \dots, S_1$$

e_i je i -tá chyba ve skryté vrstvě

e) adaptace vah mezi skrytou vrstvou L_2 a výstupní vrstvou L_3

$$\Delta w_{ji}^2 = \alpha h_i E_j \quad i = 1, 2, \dots, S_1, \quad j = 1, 2, \dots, S_2$$

Δw_{2ji} udává změnu vah mezi skrytou vrstvou L_2 a výstupní vrstvou L_3 , α je kladná konstanta, která řídí rychlost učení

f) nastavení prahů na výstupu

$$\Delta \Theta_j^2 = \alpha E_j \quad j = 1, 2, \dots, S_2$$

Θ_{2j} je přírůstek ze změny prahů ve výstupní vrstvě

g) adaptace vah mezi vstupní vrstvou L_1 a skrytou vrstvou L_2

$$\Delta w_{iS_1}^1 = \beta x_h e_i \quad i = 1, 2, \dots, S_1$$

Δw_{ih} je přírůstek ze změny vah mezi vstupní vrstvou L_1 a skrytou vrstvou L_2 , β je kladná konstanta řídící rychlost učení

$$\Delta \Theta_i^1 = \beta e_i \quad i = 1, 2, \dots, S_1$$

6) Opakujeme předcházející krok tak dlouho, dokud není splněna podmínka pro dovolenou chybu.

Mírou naučenosti sítě je velikost globální chyby neboli hodnota energie.

$$E_g = 0.5 \sum_{j=1}^m (y_j - t_j)^2$$

m je počet výstupů ze sítě, y_j je j -tý výstup ze sítě, t_j je j -tý požadovaný výstup (cílová hodnota)

Testování:

vypočet výstupních hodnot (v pořadí: skrytá vrstva, výstupní vrstva)
pomocí matic vah W_1, W_2

Problémy:

- nevhodná volba rychlosti učení (α, β) - může dojít k „přeskočení“ malých lokálních minim a k oscilacím
- není vždy možné dosáhnout lepšího průběhu chybové funkce náhodným výběrem pořadí vstupních trénovacích vektorů (např. při zpracování řečových signálů pořadí vstupních vektorů udává časovou posloupnost, která musí být dodržena)

Řešení: modifikace základního algoritmu