

Neurónové siete Inžiniersky prístup (2. diel)

Peter Sinčák

Katedra kybernetiky a umelej inteligencie

Elektrotechnická fakulta

Technická Univerzita Košice

e-mail: sincak@ccsun.tuke.sk

&

Gabriela Andrejková

Katedra matematickej informatiky

Prírodovedecká fakulta

Univerzita P.J. Šafárika, Košice

e-mail: andrejk@kosice.upjs.sk

24. septembra 2002

PodĎakovanie

Týmto vyjadrujeme **verejné podĎakovanie** študentom 4. a 5. ročníka FEI Katedry kybernetiky a umelej inteligencie TU Košice (šk. roku 1995–1996) a študentom Katedry matematickej informatiky Prírodovedeckej fakulty za **morálnu podporu a inšpirácie** pri príprave tejto publikácie. Nekonečné diskusie a **nadšenie študentov** pre neurónové siete nám pomáhalo pri príprave a dokončovaní tejto knihy.

Touto cestou Ďakujeme Ing. Rudolfovi Jakšovi za dokončovacie práce pri príprave knihy, Ing. Marekovi Hatalovi za pomoc pri konečnej úprave a počítačovej sadzbe a Editke Šutakovej za nakreslenie obrázkov. Súčasne Ďakujeme Prof. Ing. Jánovi Sarnovskému, CSc., vedúcemu Katedry kybernetiky a umelej inteligencie, ktorý vytvoril podmienky pre rozvoj tejto disciplíny na TU Košice.

Záverom Ďakujeme všetkým tým, ktorí **nám držali palce**, aby táto publikácia bola konečne dokončená.

Autori

Obsah

1	Rekurentné NN	1
2	Kontrolované učenie na RC NN	3
2.1	Hopfieldove siete	3
2.1.1	Problém obchodného cestujúceho (TSP)	5
2.2	Metóda spätného šírenia chyby (BP) na RC NN	10
2.2.1	Odvodenie adaptačného pravidla	13
3	Nekontrolované učenie na RC NN	19
3.1	Teoretický popis metód Adaptive Resonance Theory (ART) 20	
3.1.1	Úvod do ART	20
3.1.2	Siete triedy ART pre spracovanie binárnych dát	21
3.1.3	Štruktúra a dynamika sietí ART1	21
3.1.4	Teória sietí ART1	25
4	Modulárne neurónové siete	31
4.1	Štruktúra modulárnej siete	33
4.2	Pravdepodobnostná funkcia	35
4.3	Učiaci algoritmus pre regresiu	37
4.3.1	Úprava váh expertných modulov	38
4.3.2	Úprava váh bránového modulu	40
4.4	Učiaci algoritmus pre klasifikáciu	43
4.5	EM algoritmus	45
5	Učenie NN podľa stavu systému	49
5.1	Topológia NN	49
5.2	Adaptačné pravidlo pre ASE	51
5.3	Adaptačné pravidlo pre ACE	51

5.4	Repetitóriium č. 4	54
6	Záver	57
6.1	Stručné zhodnotenie zámeru autorov	57
6.2	Informácie o NN na sieti Internet	57
6.3	Knihy a časopisy o NN	58
6.4	Organizácia záujemcov o NN na Slovensku	59
	Literatúra	61
	Register	63

Používané skratky

znak	význam	definované na strane
NN	neurónová sieť (neural network)	1. diel
SV	synaptická váha	1. diel
FF	dopredná (feed-forward)	1. diel
RC	rekurentná (recurent)	1. diel
BP	metóda spätného šírenia chyby (Error Backpropagation)	1. diel
TSP	problém obchodného cestujúceho (The Travelling Salesman Problem)	5
in_i	vstup do neurónu "i"	1. diel
x_i	aktivačná hodnota neurónu "i"	1. diel
ou_i	výstupná hodnota neurónu "i"	1. diel
θ_i	prah neurónu "i"	1. diel
$J(t)$	chybová funkcia	1. diel
ev_i	očakávaná hodnota aktivácie neurónu "i"	1. diel
γ	učiaci pomer	1. diel
w_{ij}	hodnota SV smerujúcej od neurónu "j" k neurónu "i"	1. diel
e_i	chybový rozdiel $e_i = ev_i - x_i$	1. diel
δ_i	chybový signál pri BP učení	1. diel
α	momentum parameter pri BP momentum učení	1. diel
Λ	funkcia susednosti pri Kohonenových NS	1. diel

Rekurentné NN

V nasledujúcich kapitolách je poskytnutý prehľad základných metód učenia pre rekurentné NN. Pod rekurentnými NN (ďalej RC NN) rozumieme NN, ktoré majú synapsie orientované **rôznymi** smermi. V takejto štruktúre, môže byť neurón **súčasne vstupný a výstupný**. V terminológii NN v takom prípade hovoríme o **duálnom neuróne**. V [15] je poskytnutý základný koncepčný pohľad o používaní RC NN. Otázka **”kedy použiť RC NN ?”** je dosť zložitá. Vo všeobecnosti sa dodržiava pravidlo, že k RC NN siahneme až potom, ak :

- je zrejmé, že sme neuspeli s FF NN
alebo
- činnosť FF NN je neefektívna.

Vo všeobecnosti môžeme konštatovať , že RC NN predstavujú **dynamickejši** systém ako FF NN. Problémy neurodynamiky sú veľmi aktuálne z hľadiska GS a konvergencie RC NN. Tak isto ako pri FF NN rozdeľujeme metódy učenia aplikované na RC NN topológie do dvoch základných skupín:

- kontrolované učenie na RC NN (príkladom môže byť Hopfiedova sieť ako simulácie pamäte, BP pre RC NN, BP pre Elmanove RC NN)
- nekontrolované učenie na RC NN (v tejto oblasti sú dominantné hlavne metódy učenia ART, ktoré sú veľmi dôležité v rôznych aplikáciach.

V rámci RC NN ešte existuje jedna podskupina čiastočne rekurentných sietí (ďalej **partialy RC NN**). U čiastočne rekurentných sietí sa predpokladá existencia skupín neurónov, ktoré nespĺňajú vlastnosť rekurencie. Špeciálne topológie RC NN navrhli **Elman a Jordan**. Sú určené väčšinou na spracovanie sekvenčných vzoriek ako celku.

Kontrolované učenie na RC NN

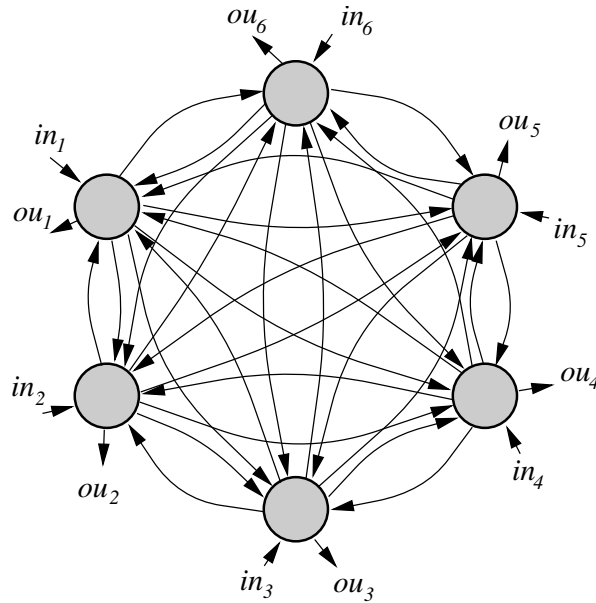
Je zrejmé, že pod kontrolovaným učením rozumieme proces, kedy ponúkame na vstup NN nejaké vzorky a k nim príslušné výsledky očakávame na výstupe. Očakávaný výstup sa môže rovnať vstupu, hlavne keď ide o simuláciu pamäte. Na rozdiel od nekontrolovaného učenia BP, nám tu ide o výstup, kým v spomínanom BP prístupe nás zaujímali výsledky na skrytej vrstve.

2.1 Hopfieldove siete

Hopfieldove siete (ďalej HS) predstavujú najjednoduchšiu verziu architektúry RC NN. Je potrebné pripomenúť, že táto sieť sa skladá z **duálnych** neurónov (viď obrázok 2.1). Ďalej je potrebné si uvedomiť, že v prípade HS ide o asynchrónnu činnosť celej HS, teda jednotlivé neuróny menia svoje stavy **asynchrónne**. Tiež je nutné poznamenať, že zmeny SV sa realizujú podľa Hebbovho pravidla. Využitie takejto siete môže byť napr. pri simulácii asociatívnej pamäte. Teda HS budeme učiť nejaké vzorky si uchovať a potom po naučení takejto siete ak dáme na vstup nejakú vzorku, očakávame, aby na výstupe bola tá istá, tj. HS ju mala v pamäti. Celý proces teda môžeme rozdeliť do dvoch fáz a to

- učenie (zmena SV)
- testovanie

Vstup do jednotlivých neurónov je vstup z externého sveta a vstup z ostatných neurónov.



Obr. 2.1: Jednoduchá Hopfieldova sieť so 6 neurónmi

$$in_i(t) = \sum_{j=1 \& j \neq i}^N w_{ij}x_j(t) + I_i(t) \quad (2.1)$$

kde

$$I_i(t) = in_{ext}(t) + \theta_i \quad (2.2)$$

adaptačné pravidlo je **Hebbovho typu** teda sa priamo prenáša aj na stavy jednotlivých neurónov, o ktorých môžeme napísať, že

$$x_i(t+1) = \begin{cases} +1 & ak \ in_i(t) > T \\ -1 & ak \ in_i(t) < T \\ x_i(t) & inak \end{cases} \quad (2.3)$$

teda v konečnom dôsledku sa prejaví samotné rozšírené Hebbovo pravidlo, ktoré hovorí, že SV sa zosilní, ak pred- a postsynaptický neurón

sú rovnakého znamienka a sa zoslabí, ak sú rôzneho znamienka. Teda¹

$$\Delta w_{ij}(t) = \gamma x_i(t)x_j(t) \quad (2.4)$$

Po naučení p – *prvkov* má SV tvar

$$w_{ij} = \begin{cases} \sum_{p=0}^{N-1} x_i^p x_j^p & \text{ak } i \neq j \\ 0 & \text{inak} \end{cases} \quad (2.5)$$

Po určitej dobe dôjde k GS HS. Ak budeme predpokladať symetrické SV teda $w_{ij} = w_{ji}$ a ak si zvolíme Ljapunovovu funkciu pre energie HS v tvare

$$E = -\frac{1}{2} \sum_i \sum_j x_i x_j w_{ij} - \sum_i \theta_i x_i \quad (2.6)$$

ak chceme aby HS našla svoju GS tak ΔE musí byť záporná a prejavovať klesajúci charakter, čo sa dá aj dokázať vid' [5]. Problémom HS je malé množstvo uchovateľných vzoriek. Jedná sa zhruba o 0.15N vzoriek zo všetkých možných vzoriek.

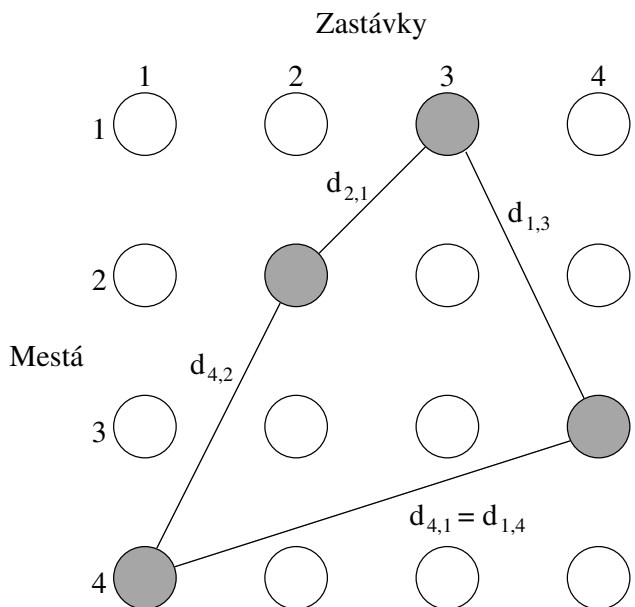
2.1.1 Problém obchodného cestujúceho (TSP)

Majme danú množinu n miest a vzdialenosť $d_{i,j}$ pre všetky dvojice miest i, j . Potrebujeme zistiť, v akom poradí má obchodný cestujúci prechádzať mestá tak, aby sa vrátil do mesta, z ktorého vyrazil a zároveň každé mesto na trase navštívil práve jedenkrát. Podmienkou však je, že vzdialenosť, ktorú precestoval, je najkratšia možná. Riešenie tohto problému je prípustné, ak sú splnené nasledujúce obmedzenia :

- obchodný cestujúci prechádza každým mestom práve jedenkrát,
- jeho cesta v grafe tvorí cyklus.

Prípustnosť riešenia je teda daná nájdením hamiltonovskej kružnice a mieru optimality riešenia popisuje súčet vzdialeností medzi mestami na hamiltonovskej kružnici.

Neurónová sieť, ktorá tento problém rieši, musí vyjadriť pre dané mesto, ktoré mesto mu predchádza a ktoré mesto za ním nasleduje. Toto je vyjadriteľné v sieti s n^2 neurónmi pri maticovom usporiadaní uvedenom na obr. 2.2



Obr. 2.2: Návrh neurónovej siete pre TSP pri $n = 4$. Kvôli prehľadnosti nie sú zakreslené prepojenia neurónov. Riešenie je zobrazené pomocou čiar. Riešením je postupnosť miest 4-2-1-3, prípadne jej cyklická permutácia.

Je zrejmé, že usporiadanie v tomto tvare pomáha pri pochopení riešenia, inak je nepodstatné. Vzťah pre energiu siete je možné upraviť pomocou zavedenia dvoch indexov pre jeden neurón. Jedná sa len o technickú úpravu vzťahu, význam zostane nezmenený.

Teraz sformulujeme túto úlohu pomocou novej sústavy dvojstavových premenných tak, aby hľadanie prípustného riešenia mohlo byť vyjadrené ako minimalizácia funkcie týchto nových premenných. Zadefinujeme maticu \mathbf{M} typu $n \times n$ s prvkami \mathbf{M}_{ij} , ktoré nadobúdajú hodnoty 0 alebo 1, pre $i, j \in \langle 1, n \rangle$. $\mathbf{M}_{ij} = 1$ vtedy, keď obchodný cestujúci prechádza mestom i v j -tom kroku. V opačnom prípade $\mathbf{M}_{ij} = 0$. V podstate hodnoty \mathbf{M}_{ij} sú ekvivalentom hodnotám neurónov x_{ij} . Prípustnému riešeniu pôvodnej úlohy teraz zodpovedá stav matice, ktorý

¹Pozor tu ide o Δw a nie w !!!

sa dá popísať nasledovne :

- v každom riadku je najviac jedna jednička, tj. pre x -te mesto platí $\mathbf{M}_{xj}\mathbf{M}_{xl} = 0$, ak $j \neq l$,
- v každom stĺpci je najviac jedna jednička, tj. pre x -ty krok obchodného cestujúceho platí $\mathbf{M}_{ix}\mathbf{M}_{kx} = 0$, ak $i \neq k$,
- v matici je práve n jedničiek, tj. $\sum_i^n \sum_j^n \mathbf{M}_{ij} = n$,
- súčet vzdialeností medzi mestami je určený maticou vzdialeností, pričom celková dĺžka cesty je $\frac{1}{2} \sum_i^n \sum_{k \neq i}^n \sum_y^n d_{ki} \mathbf{M}_{ky} (\mathbf{M}_{i,y-1} + \mathbf{M}_{i,y+1})$

Teraz vytvoríme funkcie E_a, E_b, E_c a E_d , ktoré nadobúdajú minimálne hodnoty pri splnení predchádzajúcich podmienok. Funkcie sú vytvárané na základe vzťahov uvedených v týchto podmienkách.

$$E_a = \frac{A}{2} \sum_x^n \sum_j^n \sum_{l \neq j}^n \mathbf{M}_{xj} \mathbf{M}_{xl} \quad (2.7)$$

$$E_b = \frac{B}{2} \sum_i^n \sum_x^n \sum_{i \neq k}^n \mathbf{M}_{ix} \mathbf{M}_{kx} \quad (2.8)$$

$$E_c = \frac{C}{2} \left(\sum_i^n \sum_j^n \mathbf{M}_{ij} - n \right)^2 \quad (2.9)$$

$$E_d = \frac{D}{2} \sum_i^n \sum_{k \neq i}^n \sum_y^n d_{ki} \mathbf{M}_{ky} (\mathbf{M}_{i,y-1} + \mathbf{M}_{i,y+1}) \quad (2.10)$$

Vo vzťahoch (2.7) - (2.10) A, B, C a D sú voliteľné parametre. Takže riešenie je optimálne, ak $E = E_a + E_b + E_c + E_d$ nadobúda svoje minimum. Na tomto mieste je vhodné pripomenúť si, ako vyzerá energetická funkcia pre diskretný model Hopfieldovej siete pri maticovom očíslovaní neurónov:

$$E_{hop} = -\frac{1}{2} \sum_i^n \sum_j^n \sum_k^n \sum_l^n T_{ij,kl} \mathbf{M}_{ij} \mathbf{M}_{kl} + \sum_i^n \sum_j^n \Theta_{ij} \mathbf{M}_{ij} \quad (2.11)$$

Prepísaním (2.7),(2.8),(2.9) a (2.10) na tvar energetickej funkcie (2.11) dosiahneme to, že tvar "našej" funkcie E bude ekvivalentný tvaru funkcie E_{hop} , ktorá sa počas konvergencie siete minimalizuje (základná vlastnosť Hopfieldovej siete). Po tomto prepísaní nám teda bude stačiť porovnať výsledky s (2.11) a ľahko vypočítame nastavenie váh a prahov. Nasledujúce úpravy spočívajú v zavedení symbolu Croneckerovo delta, tj. $\delta_{ij} = 1$, ak $i = j$, inak $\delta_{ij} = 0$, čo nám umožní doplniť do niektorých výrazov ďalšie sumy.

Z (2.7) vyplýva

$$E_a = \frac{A}{2} \sum_i^n \sum_j^n \sum_k^n \sum_l^n \delta_{ik}(1 - \delta_{jl}) \mathbf{M}_{ij} \mathbf{M}_{kl} \quad (2.12)$$

Výraz $\delta_{ik}(1 - \delta_{jl})$ je rovný 0, ak $i \neq k$ alebo $j = l$.

$$E_a = -\frac{1}{2} \sum_i^n \sum_j^n \sum_k^n \sum_l^n -A\delta_{ik}(1 - \delta_{jl}) \mathbf{M}_{ij} \mathbf{M}_{kl} \quad (2.13)$$

Tento výraz je vlastne výraz (2.11), kde

$$T_{ij,kl}^a = -A\delta_{ik}(1 - \delta_{jl}) \quad (2.14)$$

Analogicky z (2.8) dostaneme

$$T_{ij,kl}^b = -B\delta_{jl}(1 - \delta_{ik}) \quad (2.15)$$

Z (2.9) dostaneme

$$E_c = \frac{C}{2} \left(\sum_i^n \sum_j^n \sum_k^n \sum_l^n \mathbf{M}_{ij} \mathbf{M}_{kl} - 2n \sum_i^n \sum_j^n \mathbf{M}_{ij} + n^2 \right) \quad (2.16)$$

Po prenasobení výrazu v zátvorke výrazom $\frac{C}{2}$ dostaneme

$$E_c = -\frac{1}{2} \sum_i^n \sum_j^n \sum_k^n \sum_l^n -C\mathbf{M}_{ij} \mathbf{M}_{kl} + \sum_i^n \sum_j^n -Cn\mathbf{M}_{ij} + \frac{C}{2}n^2 \quad (2.17)$$

Pretože posledný člen $\frac{C}{2}n^2$ nezmení polohu minima vyššie uvedenej funkcie E_c , zanedbáme ho a dostávame

$$T_{ij,kl}^c = -C, \quad \Theta_{ij} = -Cn. \quad (2.18)$$

Zo (2.10) vyplýva

$$E_d = \frac{D}{2} \sum_i^n \sum_k^n \sum_y^n d_{ki} \mathbf{M}_{ky} \mathbf{M}_{i,y-1} + \frac{D}{2} \sum_i^n \sum_k^n \sum_y^n d_{ki} \mathbf{M}_{ky} \mathbf{M}_{i,y+1} \quad (2.19)$$

pričom platí, že $d_{ik} = 0$, ak $i = k$. Zavedením Croneckerovho symbolu δ_{lx} a nahradením indexu y indexom l vo výraze pre E_d dostávame

$$E_d = \frac{D}{2} \sum_i^n \sum_x^n \sum_k^n \sum_l^n d_{ki} \delta_{lx} \mathbf{M}_{kl} \mathbf{M}_{i,x-1} + \frac{D}{2} \sum_i^n \sum_x^n \sum_k^n \sum_l^n d_{ki} \delta_{lx} \mathbf{M}_{kl} \mathbf{M}_{i,x+1} \quad (2.20)$$

$E_d = 0$, ak $l \neq x$.

Položíme $j = x - 1$ v prvej časti výrazu a $j = x + 1$ v druhej časti a dostaneme

$$= \frac{D}{2} \sum_i^n \sum_j^n \sum_k^n \sum_l^n (d_{ki} \delta_{l,j+1} \mathbf{M}_{kl} \mathbf{M}_{ij} + d_{ki} \delta_{l,j-1} \mathbf{M}_{kl} \mathbf{M}_{ij}) \quad (2.21)$$

$$= -\frac{1}{2} \sum_i^n \sum_j^n \sum_k^n \sum_l^n -D d_{ki} (\delta_{l,j+1} + d_{ki} \delta_{l,j-1}) \mathbf{M}_{kl} \mathbf{M}_{ij} \quad (2.22)$$

$$T_{ij,kl}^d = -D d_{ki} (\delta_{l,j+1} + \delta_{l,j-1}) \quad (2.23)$$

Takže nakoniec pre nastavenie váh prepojení v sieti medzi neurónmi x_{ij} a x_{kl} bude platiť

$$T_{ij,kl} = T_{ij,kl}^a + T_{ij,kl}^b + T_{ij,kl}^c + T_{ij,kl}^d \quad (2.24)$$

$$T_{ij,kl}^d = -A \delta_{ik} (1 - \delta_{jl}) - B \delta_{jl} (1 - \delta_{ik}) - C - D d_{ki} (\delta_{l,j+1} \delta_{l,j-1}) \quad (2.25)$$

Po odvodení vyššie uvedených vzťahov môžeme uviesť nasledujúci algoritmus:

ALGORITMUS pre TSP:**Krok 1. Priradenie váh prepojeniam**

$$T_{ij,kl} = -A\delta_{ik}(1 - \delta_{jl}) - B\delta_{jl}(1 - \delta_{ik}) - C - Dd_{ki}(\delta_{l,j+1} + \delta_{l,j-1})$$

pre $1 \leq i, j, k, l \leq n$,

- A,B,C,D sú zvolené parametre siete (sú meniteľné),

Krok 2. Inicializácia

$$\mathbf{M}_{ij}(0) = x_{ij}, \text{ pre } 1 \leq i, j \leq n,$$

V tejto formule $\mathbf{M}_{ij}(t)$ je výstup vrcholu „ij“ v čase $t = 0$ a x_{ij} je náhodná premenná, ktorá nadobúda hodnotu 0 alebo 1.

Krok 3. Iterácia pokiaľ sieť neskonvergovala

$$\mathbf{M}_{ij}(t+1) = g_h[\sum_{k=1}^n \sum_{l=1}^n T_{ij,kl} \cdot \mathbf{M}_{kl}(t)]$$

pre $1 \leq i, j \leq n$.

Krok končí, ak sieť skonvergovala, tj. jej stav sa už nemení. Tu môže dôjsť k zacykleniu siete.

Krok 4. Opakovanie od kroku 2.

Ak došlo k zacykleniu siete, je potrebný nový výpočet začínajúci nastavením nových počiatkových hodnôt siete. Tiež je možné zmeniť parametre siete a nastaviť nové váhy, tj. začať krokom 1.

2.2 Metóda spätného šírenia chyby (BP) na RC NN

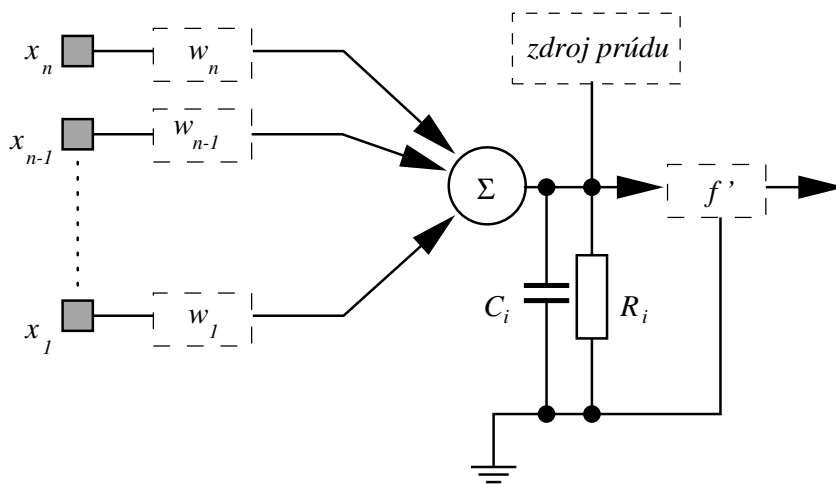
Zovšeobecnenie metódy BP na RC NN bolo prvýkrát prezentované v [16]. Logika celého procesu BP je veľmi podobná ako pre FF NN. Pri RC NN nás však do veľkej miery zaujíma dynamika činnosti RC NN z dôvodu existencie rekurentných synapsii. Teda už pri šírení signálu sa musia vytvárať podmienky GS a až potom sa môže šíriť späť samotná chyba. Pod dynamikou NN systému budeme rozumieť nejakú funkciu $D(x(t))$, kde $x(t)$ je stav neurónu v čase t , teda

$$D(x(t)) = \frac{\partial x(t)}{\partial t} \quad (2.26)$$

Teda v prípade metódy BP na RC NN musíme hľadať také adaptačné pravidlo zmeny SV, ktoré :

- bude minimalizovať chybu v zmysle BP prístupu
- bude mať podmienky na dosiahnutie stavu stability v bode $x^*(t)$ kedy je $D(x^*(t)) = 0$.

Teda **hlavným rozdielom** medzi BP na RC NN a FF NN je problém neurodynamiky. RC NN musí vykazovať stabilné vlastnosti pri šírení signálu **dopredu ako aj pri spätnom šírení chyby**. Inak môžu vzniknúť **nekontrolovateľné procesy** a celý systém bude nefunkčný. Je zrejmé, že exaktný popis modelu neurónu a jeho dynamiky bude mať význam pri štúdiu dynamiky celého systému. V literatúre týkajúcej sa NN sa používa tzv. **Resistance-Capacity model** neurónu ako východisko pre popis RC NN vid' obrázok 2.3.



Obr. 2.3: Resistance-Capacity model neurónu

Ak tento model budeme považovať za jednoduchý elektrický obvod, potom v bode **V** bude platiť Kirchhoffov zákon o súčte prúdov. Tento model má aj svoju analógiu v biologických systémoch, kde

- w_{ij} vodivosť medzi neurónmi "i" a "j"
- $x_i(t)$ priemerný potenciál neurónov na základe vstupu a výstupu z neurónov

- R_i odpor neurónovej membrány neurónu "i"
- C_i kapacita neurónovej membrány neurónu "i"
- $f(in_i(t))$ je prahová funkcia neurónu "i"

Teda potom podľa Kirchoffovho zákona dostaneme :

$$C_i \frac{\partial x_i(t)}{\partial t} + \frac{x_i}{R_i} = \sum_j w_{ij} x_j + I_i \quad (2.27)$$

kde $x_i(t) = f(in_i(t))$. Táto rovnica vlastne **determinuje** podmienky rovnováhy v neuróne, t.j. $D(x_i(t)) = 0$ na uvedenom modeli neurónu. Rovnicu (2.27) je možné upraviť do tvaru (využijeme fyzikálne zákony)

$$\tau \frac{\partial x_i}{\partial t} = -x_i(t) + \sum_j w_{ij} x_j(t) + I_i \quad (2.28)$$

kde τ je funkcia R, C . Ešte raz je potrebné zdôrazniť, že (2.28) predstavuje podmienky takej zmeny stavu neurónu x_i , aby došlo k GS. Riešením tejto diferenciálnej rovnice je bod stability $x^*(t)$ v stavovom priestore. Ak sa teraz vrátíme k reálnej topológii RC NN na obr. 2.4, tak celkovo môžeme vyjadriť vstup do ľubovoľného neurónu ako

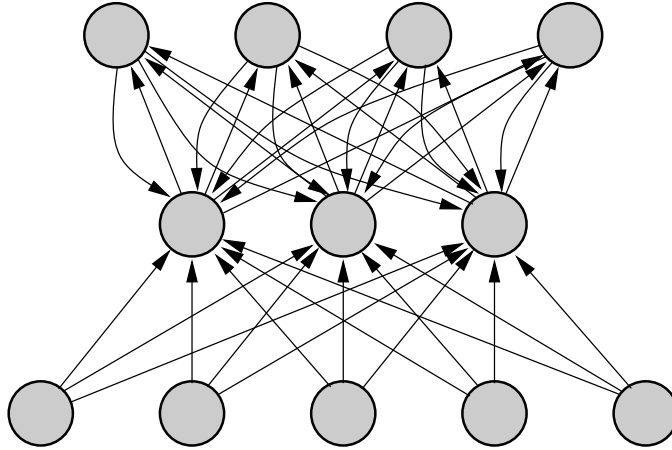
$$in_i(t) = \sum_j w_{ij} ou_j(t) + I_i \quad (2.29)$$

kde I_i je vstup do neurónu z externého sveta (θ_i , samotný vstup). Ďalej môžeme definovať chybový rozdiel (pri predpoklade identickej výstupnej funkcie $x_i(t) = ou_i(t)$) ako

$$e_i(t) = \begin{cases} (ev_i(t) - x_i^*(t)) & \text{pre vstupn neurón} \\ 0 & \text{inak} \end{cases} \quad (2.30)$$

kde $x_i^*(t)$ je spomínaný stav výstupného neurónu "i" pri dosiahnutí **rovnovážneho stavu** a $ev_i(t)$ je očakávaná hodnota neurónu v zmysle kontrolovaného učenia. Teda, ako u klasickej metódy BP deklarujeme chybovú funkciu v tvare

$$J(t) = \frac{1}{2} \sum_k e_k(t)^2 \quad (2.31)$$



Obr. 2.4: Topológia RC NN

kde k sa mení od 1 po počet neurónov vo výstupnej vrstve. V konečnom dôsledku dostaneme všeobecný výraz pre adaptačné pravidlo pre spojenie medzi ľubovoľnými "s" a "r" neurónmi:

$$\Delta w_{rs} = -\gamma \frac{\partial J}{\partial w_{rs}} \quad (2.32)$$

2.2.1 Odvodenie adaptačného pravidla

Ak chceme pravidlo exaktne vyjadriť, môžeme urobiť nasledovné kroky:

1. môžeme vyjadriť, že

$$\frac{\partial J(t)}{\partial w_{rs}} = 0.5 \sum_k \frac{\partial (e v_k(t) - x_k^*(t))^2}{\partial w_{rs}} = -e_i \sum_k \frac{\partial x_k^*(t)}{\partial w_{rs}} \quad (2.33)$$

teda

$$\Delta w_{rs} = \gamma e_i \sum_k \frac{\partial x_k^*(t)}{\partial w_{rs}} \quad (2.34)$$

týmto sa náš problém zúžil na výpočet $\frac{\partial x_k^*}{\partial w_{rs}}$

2. vieme, že všeobecne môžeme napísať

$$x_i^*(t) = f(in_i(t)) \tag{2.35}$$

kde

$$in_i(t) = \sum_j w_{ij} x_j^*(t) + I_i \tag{2.36}$$

z toho ale vyplýva, že

$$\frac{\partial x_i^*(t)}{\partial w_{rs}} = \frac{\partial x_i^*(t)}{\partial in_i(t)} \frac{\partial in_i(t)}{\partial w_{rs}} \tag{2.37}$$

3. prvý člen rovnice (2.37) z pravej strany je zrejmy lebo

$$\frac{\partial x_i^*(t)}{\partial in_i(t)} = f'(in_i(t)) \tag{2.38}$$

4. druhý člen v (2.37), môžeme pomocou (2.36) upraviť:

$$\frac{\partial in_i(t)}{\partial w_{rs}} = \sum_j \left\{ \frac{\partial w_{ij}}{\partial w_{rs}} x_j^*(t) + w_{ij} \frac{\partial x_j^*(t)}{\partial w_{rs}} \right\} \tag{2.39}$$

kde prvý člen pravej strany, vzhľadom na to, že SV sú od seba nezávislé, môžeme vyjadriť pomocou indexovej funkcie $\mathcal{C}\nabla\lambda\|\|\|\nabla\lambda\subseteq\lambda - \|\|\downarrow\sqcup\rightarrow^2$ v tvare δ_{ij}

$$\frac{\partial w_{ij}}{\partial w_{rs}} = \delta_{ir} \delta_{js} \tag{2.40}$$

kde potom $\sum_j \delta_{js}$ pre $j = s$ je rovné 1 a pre všetky ostatné je 0. Vzťah (2.39) môžeme prepísať do tvaru

$$\frac{\partial in_i(t)}{\partial w_{rs}} = \delta_{ir} x_s^* + \sum_j w_{ij} \frac{\partial x_j^*(t)}{\partial w_{rs}} \tag{2.41}$$

² $\delta_{ij} = \begin{cases} 1 & \text{pre } i = j \\ 0 & \text{ak } i \neq j \end{cases}$

5. teraz môžeme pomocou indexovej funkcie δ_{ij} napísať

$$\frac{\partial x_i^*(t)}{\partial w_{rs}} = \sum_j \delta_{ij} \frac{\partial x_j^*(t)}{\partial w_{rs}} \quad (2.42)$$

Ak je zrejmé že $x_i = f(in_i)$, tak môžeme pomocou predchádzajúceho vzorca napísať, že

$$\sum_j \delta_{ij} \frac{\partial x_j^*(t)}{\partial w_{rs}} = \frac{\partial x_i^*(t)}{\partial in_i(t)} \frac{\partial in_i(t)}{\partial w_{rs}} \quad (2.43)$$

Pravú stranu tejto rovnice môžeme prepísať za pomoci (2.38) a (2.41) do tvaru

$$\sum_j \delta_{ij} \frac{\partial x_j^*(t)}{\partial w_{rs}} = f'(in_i(t)) \left\{ \delta_{ir} x_s^* + \sum_j w_{ij} \frac{\partial x_j^*(t)}{\partial w_{rs}} \right\} \quad (2.44)$$

po úprave tohto vzorca dostaneme nasledovný výraz

$$\delta_{ir} f'(in_i(t)) x_s^* = \sum_j \left\{ \underbrace{\delta_{ij} - f'(in_i(t)) w_{ij}}_{L_{ij}} \right\} \frac{\partial x_j^*(t)}{\partial w_{rs}} \quad (2.45)$$

kde teda môžeme skrátene napísať

$$\sum_j L_{ij} \frac{\partial x_j^*(t)}{\partial w_{rs}} = \delta_{ir} f'(in_i(t)) x_s^*(t) \quad (2.46)$$

kde $i = 1, \dots, N$. Odtiaľ potom môžeme prejsť k vektorovému vyjadreniu pomocou vektorov \mathbf{L} a \mathbf{x}^* a to nasledovne

$$\mathbf{L} \frac{\partial \mathbf{x}^*(t)}{\partial w_{rs}} = \begin{pmatrix} \delta_{1r} f'(in_1) \\ \delta_{2r} f'(in_2) \\ \vdots \\ \delta_{Nr} f'(in_N) \end{pmatrix} \mathbf{x}_s^*(t) \quad (2.47)$$

z (2.47) môžeme vyjadriť $\frac{\partial \mathbf{x}^*(t)}{\partial w_{rs}}$:

$$\frac{\partial \mathbf{x}^*(t)}{\partial w_{rs}} = (\mathbf{L}^{-1}) \left\{ \begin{array}{c} \delta_{1r} f'(in_1) \\ \delta_{2r} f'(in_2) \\ \vdots \\ \delta_{Nr} f'(in_N) \end{array} \right\} x_s^*(t) \quad (2.48)$$

Teraz keď sa vrátíme ku skalárnemu vyjadreniu pre "k"-ty riadok dostaneme nasledovný výraz

$$\frac{\partial x_k^*(t)}{\partial w_{rs}} = (\mathbf{L}_{kr}^{-1}) f'(in_r(t)) x_s^*(t) \quad (2.49)$$

kde (\mathbf{L}_{kr}^{-1}) je kr-tý element v inverznej matici. Teda konečne môžeme napísať pomocou (2.34), že

$$\Delta w_{rs} = \gamma \sum_k e_i(\mathbf{L}_{kr}^{-1}) f'(in_r(t)) x_s^*(t) \quad (2.50)$$

Vzťah (2.50) môžeme prezentovať v obvyklom zápise pre BP v tvare³

$$\Delta w_{rs} = \gamma \delta_r x_s^*(t) \quad (2.51)$$

kde

$$\delta_r = f'(in_r) \sum_k e_i(\mathbf{L}^{-1})_{kr} \quad (2.52)$$

Nevýhodou výrazu (2.51) je, že vyžaduje výpočet $(\mathbf{L})^{-1}$, čo je globálna a nie lokálna operácia. Toto je možné vyriešiť nasledovným postupom

1. vo vzťahu (2.52) zavedieme substitúciu teda

$$\delta_r^* = f'(in_r(t)) y_r^*(t) \quad (2.53)$$

kde je zrejmé, že $y_r^*(t)$ je vlastne (z (2.30))

$$y_r^*(t) = \sum_k e_i(\mathbf{L}^{-1})_{kr} \quad (2.54)$$

³Pozor na rozdiel medzi indexovou funkciou δ_{ij} a chybovým signálom δ_r

2. teraz keď vzorec (2.54) upravíme, resp. ho rozpíšeme a vyberieme z neho sadu rovníc po eliminácii (\mathbf{L}^{-1}), tak potom dostaneme e_i :

$$e_i(t) = \sum_r \mathbf{L}_{ri} y_r^*(t) \quad (2.55)$$

Ak zo vzťahu (2.45) vyberieme časť pre \mathbf{L} , tak potom

$$e_i(t) = \sum_r \{ \delta_{ri} - f'(in_r(t)) w_{ri} \} y_r^*(t) \quad (2.56)$$

3. ale vieme, že

$$\sum_r \delta_{ri} y_r^*(t) = y_i^*(t) \quad (2.57)$$

a teda vzťah (2.56) môžeme prepísať do tvaru

$$e_i = y_i^*(t) - \sum_r f'(in_r(t)) w_{ri} y_r^*(t) \quad (2.58)$$

Ak zameníme index "r" za "j", môžeme dostať po úprave tvar

$$0 = - y_i^*(t) + \sum_j f'(in_j^*(t)) w_{ji} y_j^*(t) + e_i(t) \quad (2.59)$$

4. ak predpokladáme, že skutočne $y_i^*(t)$ je bodom stability⁴ tak potom musí platiť $\frac{\partial y_i(t)}{\partial t} = 0$, resp. potom

$$\frac{\partial y_i(t)}{\partial t} = - y_i(t) + \sum_j f'(in_j^*) w_{ji} y_j(t) + e_i(t) \quad (2.60)$$

Rovnica (2.60) vyjadruje neurodynamiku systému, a preto tam je $y_i(t)$ a nie ako v rovnici (2.59) $y_i^*(t)$.

Teda ak rovnica (2.28) vyjadrovala formu dopredného šírenia cez synapsie \mathbf{w}_{ij} tak vzťah (2.60) vyjadruje dynamiku spätného šírenia chyby po synapsii \mathbf{w}_{ji} . Ak porovnáme tieto vzťahy, tak zistíme, že \mathbf{w}_{ij} bol nahradený vo vzťahu (2.60) výrazom $f'(in_j(t)) w_{ji}$. Teda celkový postup môžeme zhrnúť do nasledovných bodov:

⁴ $y_i^*(t)$ je odvodený od $x_i^*(t)$ cez vzťah (2.54)

- pre vstup do NN je nutné určiť jednotlivé θ_i a potom stabilizovať RC NN v zmysle (2.28) výpočtom pevných bodov $\mathbf{x}^*(\mathbf{t})$
- vypočítame chybový signál e_i podľa (2.30)
- vypočítať zo vzťahu (2.60) pevný bod $\mathbf{y}^*(\mathbf{t})$ pre stabilné spätné šírenie chyby
- potom na základe (2.51) a (2.53) vypočítame nové hodnoty zmien SV Δw_{rs} .

Z hľadiska porovnania autor v [5] komentuje zvýšenie účinnosti učenia a skrátenie času pri použití RC NN. Na druhej strane BP na RC NN je citlivejší na zmenu parametrov samotného učenia γ .

Nekontrolované učenie na RC NN

Skupina metód, ktoré charakterizujú nekontrolované učenie na RC NN, predstavujú tzv. prístup typu "Adaptive Resonance Theory" (ďalej ART)¹. Vo všeobecnosti ich môžeme charakterizovať ako metódy s cieľom zhlukovania dát. Tento prístup bol ovplyvnený samotným konkurenčným učením a impulzom vyplývajúcim zo sledovania biologických systémov. Problémom zostáva neexistencia dôkazu GS pri konkurenčnom učení, čo môže spôsobiť problémy v oblasti stability. Určitým východiskom je riešenie problému postupnou redukciou učiaceho parametra na $\rightarrow 0$. Potom určite nastane GS, ale na druhej strane NN nebude tak **adaptívna**², teda nebude vedieť reagovať na nové typy dát. Tento problém prvýkrát zdôradnil Stefan Grossberg a nazval to **dilema stability-plasticity**.

A práve prístup ART, kde **resonance** znamená súlad medzi vstupom a doteraz už identifikovanými zhlukmi, sa pokúša problém nejako vyriešiť. Niekedy v literatúre sa stretávame s prívlastkom u týchto metód **stabilné metódy** konkurenčného učenia. Boli vyvinuté ART1 pre binárne vstupy, ďalej analógová verzia ART2 ako aj ďalšia verzia ART-MAP, ktorá predstavuje **kontrolované učenie** typu ART.

¹V NN literatúre sa nazývajú aj **Cluster discovery networks**

²plastická

3.1 Teoretický popis metód Adaptive Resonance Theory (ART)

3.1.1 Úvod do ART

Výraznou črtou ľudskej pamäti je jej schopnosť učiť sa nové veci bez toho, aby zabúdala predtým nadobudnuté vedomosti. Bolo by veľmi výhodné, ak by sa podarilo vytvoriť umelé neurónové siete s podobnými vlastnosťami. Ale väčšina dosiaľ používaných neurónových sietí túto schopnosť nemá a skôr má sklon zabúdať staré znalosti v prípade, že sa pokúšame doplniť nejaké znalosti nové. **Napríklad, ak sme sieť typu FF NN s učiacim algoritmom backpropagation naučili 10 vecí a chceme doplniť jednu novú, sieť musí byť trénovaná odznova použitím všetkých 11 vzoriek. Tento problém je potrebné riešiť, v prípade, že chceme vytvoriť systém schopný autonómne sa prispôbiť nečakaným zmenám vonkajšieho prostredia v reálnom čase.** Ukázalo sa, že systém vyhovujúci pre daný problém by mal optimálne riešiť spomínanú dilemu stability a plasticity.

Vlastnosti stability a plasticity môžu byť definované takto:

Plasticita: Systém v priebehu učenia generuje rozpoznávacie kódy ako odozvu na postupnosť vstupných vzoriek z okolitého prostredia. Pritom sa postupne vytvárajú takzvané vzorky kritických črt, čo znamená, že každý rozpoznávací kód si ponechá len tie črty, ktoré ho odlišujú od kódov ostatných tried.

Stabilita: V priebehu učenia sa vytvárajú stabilné stavy, ktoré zabezpečujú priamu odozvu na už známu vzorku, pričom sa vnútorne rozhoduje o tom, či sa pre prezentovanú vzorku vytvorí nová kategória, alebo či bude iba upravený kód niektorej už existujúcej kategórie.

Potom dilema stability a plasticity môže byť popísaná sériou otázok:

- Ako zabezpečiť, aby systém zostal dostatočne adaptívny (plastický) v odozve na výrazne novú vstupnú vzorku, a pritom bol nemenný (stabilný) v prípade, že vstup je nedôležitý ?

- Ako má systém vedieť, kedy sa má prepnúť do stabilného a kedy do plastického módu, tzn. ako dosiahnuť stabilitu bez neflexibility a plasticitu bez vzniku chaosu v systéme ?
- Ako zaistiť, aby si systém zachoval skôr naučené vedomosti a zároveň sa učil nové veci ?

Názov ART súvisí so spôsobom, akým v sieti prebieha učenie a hľadanie víťaza. Vstupná informácia osciluje vo forme výstupných hodnôt medzi dvomi vrstvami neurónov, až dokiaľ sa nedosiahne rezonancia. V tom momente nastáva učenie, čiže adaptácia váh. Rezonancia môže nastať dvoma rôznymi spôsobmi, a to v prípade, že sieť už v minulosti spracovávala takú istú alebo veľmi podobnú vzorku, vtedy rezonancia nastáva okamžite; alebo v prípade, že vstupná vzorka je odlišná od všetkých predošlých, spúšťa sa proces prehľadávania naučených kódov a porovnáva sa ich podobnosť s prezentovanou vzorkou, pričom je definovaná určitá prahová hodnota, ktorá určuje minimálnu prípustnú podobnosť víťaznej triedy. Ak tento prah nespĺňa žiadna z už známych tried, systém vytvorí triedu novú, identickú s prezentovanou vzorkou. Takto sa zároveň dosahuje stabilita (sieť rezonuje v prípade známeho vstupu) aj plasticita (zachovanie schopnosti učiť sa nové neznáme vzorky).

3.1.2 Siete triedy ART pre spracovanie binárnych dát

Teória sietí ART bola pôvodne vypracovaná iba pre binárne dáta. Prvá z tejto triedy sietí sa dnes nazýva ART1 a slúži pre nekontrolované učenie. Jej rozšírená modifikácia pre kontrolované učenie sa nazýva ART-MAP a umožňuje kontrolované učenie klasifikačných tried s možnosťou zovšeobecnenia v množine vzoriek ako aj v množine tried. Uvedenými dvoma typmi sietí sa zaoberá nasledujúca časť.

3.1.3 Štruktúra a dynamika sietí ART1

Základná architektúra sietí ART1 je tvorená tromi skupinami neurónov (viď Obr. 3.1):

1. Vstupná vrstva (označovaná F_0) slúži iba na zaznamenanie hodnôt vstupnej vzorky.

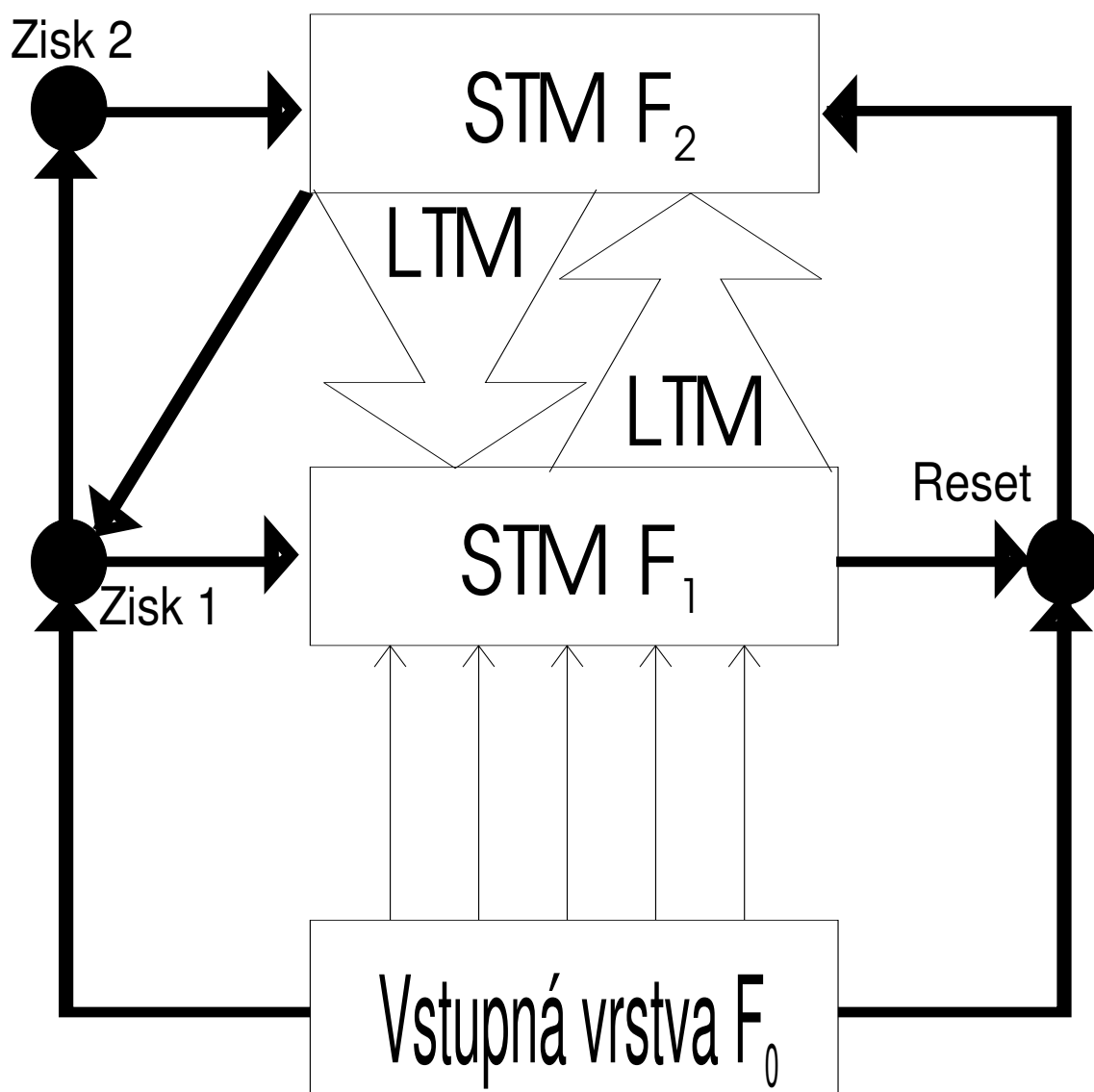
2. Porovnávacia vrstva (označovaná F_1), v ktorej sa zisťuje, či je víťazná kategória dostatočne podobná vstupnej vzorke.
3. Rozpoznávacia vrstva (označovaná F_2), aktivácia neurónov ktorej určuje víťaznú triedu v danom kroku hľadania.

Vrstvy F_1 a F_2 tvoria tzv. krátkodobú pamäť (STM - short term memory), pretože sa v nich uchovávajú informácie týkajúce sa iba jednej prezentácie vzoriek, zatiaľ čo linky spájajúce tieto dve vrstvy úplným obojsmerným prepojením, tvoria dlhodobú pamäť (LTM - long term memory), keďže uchovávajú informácie o všetkých doteraz naučených kategóriách. Medzi vrstvami F_1 a F_2 oscilujú signály vyvolané prezentáciou vzorky na vstupe, a v prípade, že porovnávacia vrstva vyhodnotí, že odozva rozpoznávacej vrstvy reprezentuje prezentovanú vzorku dostatočne, nastáva rezonančný stav a následne adaptácia liniek medzi víťazným rozpoznávacím neurónom a všetkými neurónmi v komparačnej vrstve F_1 .

Okrem týchto troch vrstiev neurónov (F_0 , F_1 a F_2) sieť obsahuje niekoľko ďalších špeciálnych neurónov slúžiacich k riadeniu jej činnosti. Neuróny *Zisk1* a *Zisk2* slúžia výlučne k synchronizácii siete. Ich úlohou je rozhodovať, kedy sa môžu aktivovať vrstvy F_1 a F_2 . Riadiaci neurón *Reset* má úlohu definovanú priamo v algoritme učenia siete. Tento neurón rozhoduje na základe porovnania vstupnej a komparačnej vrstvy o tom, či nájdená víťazná kategória v rozpoznávacej vrstve spĺňa definované kritéria. V prípade, že kritérium podobnosti splnené nebolo, signál z tohto neurónu zmrazí aktuálneho víťaza vo vrstve F_2 na celý zvyšok prezentácie danej vzorky, čím umožní pokračovať v hľadaní lepšej triedy, prípadne vytvoriť triedu novú.

Na obrázku sú informačné linky označené bielymi a riadiace linky čiernymi šípkami. Každá čierna čiara, aj keď to na obrázku nie je vyznačené, predstavuje linky od všetkých, resp. ku všetkým neurónom vrstvy, z alebo do ktorej smeruje.

Základnou bázou, na ktorej sú siete ART postavené, je konkurenčné učenie (competitive learning), ktoré sa uskutočňuje medzi vrstvami F_1 a F_2 . Neuróny vrstvy F_2 majú okrem vstupov z vrstvy F_1 aj záporné prepojenia medzi sebou a kladnú spätnú väzbu na seba samých, čo spôsobí, že neurón, ktorý má najväčšiu hodnotu aktivácie potláča aktiváciu neurónov ostatných, až kým sa po niekoľkých propagáciách aktivácia



Obr. 3.1: Architektúra neurónovej siete ART1

všetkých ostatných neurónov neustáli na nule. Linky vedúce k víťaznému neurónu sa potom nastaví na takú hodnotu, že pri opakovanej prezentácii danej vzorky bude aktivácia víťaza ešte väčšia. Takéto správanie sa siete sa nazýva tiež **víťaz berie všetko (winner takes all)**.

Učiaci cyklus sa začína spracovaním vstupnej vzorky vo vrstve F_0 . Neurón *Zisk1* v ďalšom kroku zabezpečí, že vzorka sa nezmenená dostane do porovnávacej vrstvy. Tento neurón má jednotkovú hodnotu na výstupe iba v prípade, že na vstupe je nenulová hodnota a zároveň nie je aktivovaný ani jeden neurón vo vrstve F_2 . A keďže neuróny vo vrstve F_1 sa aktivujú podľa tzv. *Pravidla 2/3*, ktoré vraví že neurón je aktívny práve vtedy, ak sú aktívne dva z jeho troch vstupov (*Zisk1*, F_0 , F_2), za daného stavu (vrstva F_2 neaktívna, *Zisk1* aktívny, vrstva F_0 obsahuje vstup) sa vstup nezmenený skopíruje do vrstvy F_1 .

Vrstva F_1 je v smere zdola nahor plne prepojená s vrstvou F_2 . Týmto prepojením je vzorka propagovaná ďalej k porovnávacím neurónom. A práve tu nastáva konkurenčné učenie, takže sa nájde práve jeden víťazný neurón, ktorého výstupná hodnota sa nastaví na 1, výstupné hodnoty ostatných sa vynulujú.

Týmto bola predbežne určená trieda, o ktorej sieť „predpokladá“, že by do nej mala byť zatriedená prezentovaná vzorka. Čiže, ak máme vo vrstve F_2 M neurónov, neaktívnych bude práve $M - 1$ z nich, a aktívny bude iba jediný víťaz. Táto aktivačná vzorka je spätne propagovaná cez linky spájajúce vrstvy F_2 a F_1 zhora nadol.

V tomto momente je už ale výstupná hodnota neurónu *Zisk1* nulová, keďže vo vrstve F_2 je už jeden neurón aktívny. Takže vo vrstve F_1 budú teraz aktivované iba tie neuróny, ktoré získali pozitívny signál zároveň od im zodpovedajúceho vstupného neurónu, ako aj od víťaza vo vrstve F_2 .

Na začiatku učenia sú váhy všetkých liniek inicializované na hodnotu 1. To znamená, že ak v rozpoznávacej vrstve vyhral predtým nepoužitý neurón, v porovnávacej vrstve sa opäť objaví nezmenená vstupná vzorka.

V prípade, že víťaz F_2 je už naučený na klasifikáciu nejakej inej vzorky, sú váhy niektorých od neho vedúcich liniek nulové, čo znamená, že zodpovedajúce neuróny vo vrstve F_1 sa neaktivujú, aj keď zo vstupu k nim prichádza jednotkový signál. Z toho vyplýva, že vektor re-

prezentovaný porovnávacou vrstvou tvorí vždy podmnožinu vstupného vektora.

Teraz musí sieť rozhodnúť o tom, či predpokladaná trieda skutočne klasifikuje prezentovanú vzorku. To sa deje porovnaním podobnosti vektorov F_0 a F_1 .

Ak sa zistí, že zhoda vektorov F_0 a F_1 je dostatočne veľká, nasleduje úprava váh medzi vrstvami F_1 a F_2 . Váhy liniek smerujúcich do F_2 sa nastaví tak, aby bola pri opakovanej prezentácii danej vzorky víťazná trieda zvolená priamo. Dolesmerujúce linky sa nastaví na rovnaké hodnoty, aké majú aktivácie neurónov vo vrstve F_1 . To znamená, že do týchto váh sa uloží vzorka tvoriaca prienik medzi aktuálnou vzorkou a pôvodným klasifikačným kódom víťaznej triedy.

V prípade, že podobnosť vektorov F_0 a F_1 nie je dostatočná, aktivuje sa *Reset*, ktorý spôsobí zmrazenie aktuálneho víťaza na zvyšok prezentácie, čo znamená, že jeho výstupná hodnota ostane až do jej konca nulová. Tým sa sieť dostala do stavu zhodného s počiatočným a celý cyklus sa opakuje aktiváciou neurónu *Zisk1*, skopírovaním vstupu do porovnávacjej vrstvy atď., až kým sa nenájde zodpovedajúca klasifikačná trieda, ktorá vygeneruje dostatočne podobný porovnávací vektor F_1 , alebo kým sa neodskúšajú všetky možné triedy, a ani jedna z nich nesplní kritérium podobnosti, čo znamená, že k dispozícii nie je dostatočné množstvo tried.

Stručne sa algoritmus učenia ART1 dá zhrnúť do týchto bodov:

1. prezentácia novej vzorky
2. nájdenie víťaza vo vrstve F_2
3. otestovanie navrhovaného víťaza vo vrstve F_1
4. v prípade, že test zlyhal a vo vrstve F_2 sú ešte neotestované neuróny, zmrazenie víťaza a prechod na bod 2, inak koniec
5. adaptácia váh medzi vrstvami F_1 a F_2

3.1.4 Teória sietí ART1

Zo slovného popisu dynamiky vyplývajú pre aktiváciu jednotlivých vrstiev neurónov tieto vzťahy:

- Neuróny vo vrstve F_1 sa správajú podľa *Pravidla 2/3*, čo sa dá zapísať:

$$x_i = \begin{cases} 1 & \text{ak } I_i + g_1 + \sum_{j=1}^M y_j w_{ji} > 1 + \tilde{w}, \\ 0 & \text{inak,} \end{cases} \quad (3.1)$$

kde I_i je vstup $F_0 \rightarrow F_1$, g_1 je výstupná hodnota neurónu *Zisk1* a $\sum y_j w_{ji}$ je suma výstupných hodnôt y_j vo vrstve F_2 vážená trénovateľnými váhami w_{ji} . \tilde{w} je konštanta z intervalu

$$0 < \tilde{w} < 1. \quad (3.2)$$

- Neurón *Zisk1* sa aktivuje podľa nasledovného pravidla:

$$g_1 = \begin{cases} 1 & \text{ak } \sum_i I_i > 0 \text{ a } \sum_j y_j \leq 0, \\ 0 & \text{inak,} \end{cases} \quad (3.3)$$

to znamená, že k tomu aby bolo g_1 aktívne, musia byť všetky neuróny vo vrstve F_2 neaktívne a na vstupe musí byť nenulová hodnota.

Takže, ak je F_2 neaktívne, rovnica (3.1) sa redukuje na:

$$x_i = \begin{cases} 1 & \text{ak } I_i = 1, \\ 0 & \text{inak,} \end{cases} \quad (3.4)$$

Inak je aktívny práve jeden neurón y_J vo vrstve F_2 . Vtedy sa rovnica (3.1) zmení na:

$$x_i = \begin{cases} 1 & \text{ak } I_i = 1 \text{ a } w_{Ji} > \bar{z}, \\ 0 & \text{inak,} \end{cases} \quad (3.5)$$

- Pre definíciu aktivácie neurónov vo vrstve F_2 sa zavádza premenná T_j , ktorá udáva vstup zo siete do j-tého neurónu, takto:

$$T_j = \frac{\sum_{i=1}^N x_i w_{ij}}{\alpha + \sum_{i=1}^N w_{ij}} \quad (3.6)$$

kde w_{ij} sú váhy $F_2 \rightarrow F_1$. Potom, za predpokladu, že existuje také j , že $T_j > 0$, ako víťazná bude zvolená trieda J , pre ktorú platí:

$$T_J = \max_{1 \leq j \leq M} \{T_j\} \quad (3.7)$$

Takto by sa mohlo stať, že zároveň niekoľko neurónov vo vrstve F_2 nadobudne rovnakú maximálnu hodnotu:

$$T_{J_1} = T_{J_2} = \dots = T_{J_k} = \max_{1 \leq j \leq M} \{T_j\},$$

čo by znamenalo nejednoznačného víťaza a nasledujúci postup by sa značne skomplikoval. Preto sa vždy uvažuje, že víťaz je jednoznačný, a to ten s najmenším indexom.

- Neurón *Zisk2* má funkciu analogickú s neurónom *Zisk1*, a to blokovať všetky neuróny vo vrstve F_2 . V priebehu učenia ART1 je dôležité, aby bola táto podmienka platná iba pri prezentácii novej vzorky, takže neurón *Zisk2* sa aktivuje práve len v tomto momente.

Predtým, než budú popísané pravidlá pre učenie, si definujeme význam označenia, ktoré je ďalej používané. Ak \bar{a} je binárny vektor s L zložkami, čiže $\bar{a} = (a_1, a_2, \dots, a_L)$, potom *norma* tohoto vektora je definovaná nasledovne:

$$|\bar{a}| = \sum_{i=1}^L a_i, \quad (3.8)$$

čo značí, že $|\bar{a}|$ udáva počet jednotiek vo vektore \bar{a} . Ďalej si definujeme *prienik dvoch vektorov* \bar{a} a \bar{b} ako vektor $\bar{c} \equiv \bar{a} \cap \bar{b}$, kde

$$c_i = 1 \Leftrightarrow a_i = 1 \text{ a } b_i = 1 \quad (3.9)$$

a na záver pojem *podmnožiny vektora* ako

$$\bar{a} \subset \bar{b} \Leftrightarrow \bar{a} \cap \bar{b} = \bar{a}. \quad (3.10)$$

Potom pre učenie platí:

- Váhy $F_2 \rightarrow F_1$ sa menia podľa pravidiel, ktoré boli uvedené už v predošlej časti tejto kapitoly. Teraz ich rozšírime len o matematický popis. Pri inicializácii sa váhy všetkých týchto liniek nastavia na ich maximálnu hodnotu, tzn.:

$$\bar{w}_{ji}(0) = 1 \quad \forall 1 \leq j \leq M, 1 \leq i \leq N. \quad (3.11)$$

Úprava váh nastáva vždy až potom čo bol určený víťaz y_J vo vrstve F_2 a menia sa iba váhy liniek vedúcich z tohto neurónu. Formulovať to môžeme takto:

$$\frac{dw_{ij}}{dt} = y_j(x_i - w_{ij}(t - 1)), \quad (3.12)$$

z čoho vyplýva, že zmena sa deje prvýkrát iba ak daná linka vychádza z víťazného neurónu a druhýkrát iba ak existuje rozdiel medzi predtým nastavenou váhou linky a terajšiou aktiváciou i -tého neurónu x_i . A keďže nemôže nastať prípad, že $x_i = 1$ a zároveň $w_{ji} = 0$, váha w_{ji} sa buď nemení, alebo sa mení z 1 na 0, čím sa zabezpečuje stabilita systému. Zmenu váhového vektora víťazného neurónu potom môžeme zapísať:

$$\bar{w}_J^{nov} = \begin{cases} \bar{I} \cap \bar{w}_J^{star} & \text{ak } y_J \text{ už niekedy vyhral,} \\ \bar{I} & \text{ak } y_J \text{ vyhral prvýkrát} \end{cases} \quad (3.13)$$

- Váhy $F_1 \rightarrow F_2$ majú dôležitejšiu úlohu, pretože sa musia starať aj o to, aby sa nové triedy vytvárali systematicky, t.j. aby sa nepreskakovali niektoré nepoužité neuróny, a tiež aby sa vždy najprv otestovali už použité neuróny a až potom, čo sa tieto ukážu ako nezodpovedajúce kritériám, použije sa nový neurón. Toto je zabezpečené inicializáciou váh, pre ktoré platí:

$$\bar{w}_{ij}(0) = \alpha_j \quad \forall 1 \leq j \leq M, 1 \leq i \leq N \quad (3.14)$$

kde

$$0 < \alpha_M < \alpha_{M-1} < \dots < \alpha_1 \leq \frac{1}{\beta + N} \quad (3.15)$$

Samotná úprava váh je totožná s úpravou váh $F_2 \rightarrow F_1$ až na to, že tu sú nové váhy ešte normované na hodnotu menšiu ako 1, čiže

$$\bar{W}_J^{nov} = \frac{\bar{I} \cap \bar{w}_J^{star}}{\beta + |\bar{I} \cap \bar{w}_J^{star}|} \quad (3.16)$$

kde β je bezvýznamne malá hodnota.

- Ako posledný je popísaný test hypotézy, tzn. overenie či vífazná kategória určená rovnicami (3.1), (3.7) je uspokojivá z hľadiska parametra podobnosti ρ (vigilance parameter). Neurón *Reset* sa aktivuje, ak platí

$$Reset \Leftrightarrow \frac{|\bar{I} \cap \bar{Z}_J|}{|\bar{I}|} \leq \rho \quad (3.17)$$

V prípade, že podmienka neplatí, algoritmus sa dostáva do poslednej fázy-úpravy váh, popísanej rovnicami (3.13) a (3.16). Ak podmienka platí, vyšle sa signál reset, ktorý zmrazí aktuálneho vífaza y_j , následkom čoho sa zase aktivuje neurón *Zisk1* a celý cyklus sa opakuje.

Z poslednej rovnice vyplýva dôležité pravidlo určujúce, aké hodnoty parametra ρ volíme. Ak dáme $\rho \leq 0$, reset nenastane nikdy, čiže všetky vzorky sú klasifikované do tej istej triedy. Na druhej strane pre $\rho = 1$ každej odlišnej vzorke bude vytvorená nová kategória. Ak by ρ nadobudlo hodnotu väčšiu ako jedna, reset nastane v každom prípade, čo spôsobí zmrazenie siete, a to je nekorektné správanie.

Modulárne neurónové siete

Je zrejmé, že učenie je z hľadiska svojej dĺžky a presnosti zložitý proces, ktorý narastá so zložitou riešeného problému. Preto ak je možné rozdeliť celkovú úlohu na rad podúloh, tak je vhodné použiť tzv. **modulárne neurónové siete** MNN. Príkladom môže byť napr. problém aproximácie funkcie vyjadrenej vzťahom (4.1), pre ktorú je možné vytvoriť sieť s dvoma modulmi, ktoré sú naučené príslušným častiam tejto funkcie.

$$f(x) = \begin{cases} x & \text{ak } x > 0 \\ -x & \text{ak } x \leq 0 \end{cases} \quad (4.1)$$

Všeobecná definícia MNN môže byť nasledovná:

Neurónové siete sa nazývajú modulárne, ak samotné výpočty v NN je možné dekomponovať do 2 alebo viacerých modulov (subsystémov), ktoré môžu fungovať bez vzájomnej komunikácie. Výstupy z týchto modulov sú vstupnými signálmi do integrujúceho branového modulu, ktorý môže mať nasledovné funkcie:

- rozhodne, ktorý z modulov sa bude učiť – ktorými tréningovými vzorkami
- rozhodne, ako budú kombinované vstupy na konečný výstup

Modularita je teda preto chápana ako proces spracovania v zmysle metódy

rozdeľuj a panuj!

Výhody MNN by sme mohli zhrnúť do týchto bodov:

1. rýchlosť učenia na svojich dátach
2. variabilita učenia, teda každý subsystém môže mať svoje vlastné adaptačné pravidla
3. schopnosť riešiť komplikované problémy efektívnejšie ako jednoduché NN

Pri MNN poznáme dva základne subsystémy a to:

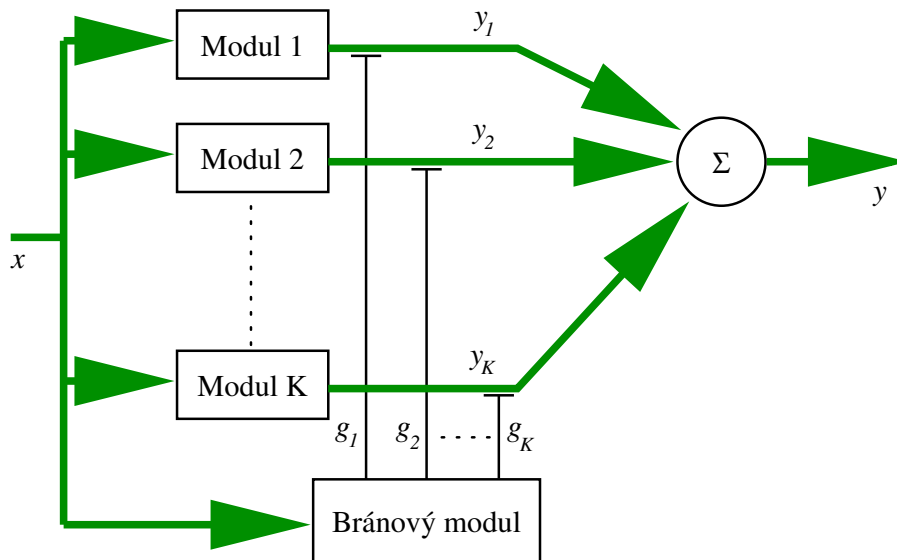
- **expertné siete** – **EN** (expert networks)
- **bránové siete** – **GN** (gating networks)

každá z týchto sietí má svoj vlastný adaptačný mechanizmus.

V modulárnej architektúre sa spájajú prvky kontrolovaného a nekontrolovaného učenia.

- *Kontrolované učenie* je charakterizované tým, že každá učiaci vzorka sa skladá z dvoch častí: vstupu a k nemu prisluchajúceho výstupu.
- *Nekontrolované učenie* je charakterizované modulmi, ktoré medzi sebou "súťažia" o možnosť produkovania správneho výstupu. O tom, ktorý modul sa bude učiť jednotlivé vstupné vzorky teda nerozhoduje externý učiteľ, ale schopnosť jednotlivých modulov produkovať výstup, ktorý je najbližšie k požadovanému.

Počas učenia sa uplatňuje efekt *kladnej spätnej väzby*. Modul, ktorý "odpovedal" najlepšie na daný vstup, prijíma najväčšie množstvo učiacej informácie v etape úpravy hodnôt váh. Moduly, ktoré produkovali horšie výsledky, prijímajú menšie množstvo učiacej informácie. Každý modul prijíma také množstvo učiacej informácie, ktoré je úmerné jeho schopnosti učenia. *Bránový modul* (gating module) plní úlohu klasifikátora nad celým definičným oborom funkcie. *Expertné moduly* pracujú ako aproximátory čiastkových funkcií v jednotlivých častiach definičného oboru. Vzájomnou konkurenciou sa po skončení učenia dosiahne špecializácia jednotlivých expertných modulov na jednotlivé časti definičného oboru aproximovanej funkcie.



Obr. 4.1: Základná bloková schéma modulárnej neurónovej siete pozostávajúcej z K expertných modulov a jedného bránového modulu.

4.1 Štruktúra modulárnej siete

Bloková štruktúra základného modelu modulárnej neurónovej siete je zobrazená na Obr. 4.1. Sieť pozostáva z K expertných modulov a jedného bránového modulu. Všetky moduly sú zložené z jednej skrytej vrstvy. Medzi vstupnou vrstvou a vrstvami jednotlivých modulov sú prepojenia typu „full connection“. Nech sú vstupné vzorky reprezentované vstupným vektorom \mathbf{x} s rozmerom p a výstupným vektorom \mathbf{d} s rozmerom q . Expertné moduly obsahujú q neurónov, bránová sieť obsahuje K neurónov. Každému expertnému modulu prislúcha jeden neurón bránového modulu. Vstupný vektor \mathbf{x} je zo vstupu privedený do všetkých modulov naraz. Každý expertný modul produkuje výstupný vektor \mathbf{y}_i , ktorý má q prvkov. Bránový modul produkuje vektor \mathbf{g} s K prvkami. Nech g_i je výstupná hodnota i -teho neurónu bránového mo-

dulu. Celkový výstupný vektor \mathbf{y} sa vypočíta ako súčet výstupov jednotlivých expertných modulov \mathbf{y}_i násobených príslušnými hodnotami g_i .

$$\mathbf{y} = \sum_{i=1}^K g_i \mathbf{y}_i \quad (4.2)$$

Druh úlohy, pre ktorý je určené použitie siete určuje typ aktivačnej funkcie neurónov v expertných sieťach. Pre potreby *regresie* sú určené lineárne aktivačné funkcie, pre potreby *klasifikácie* sú určené nelineárne neuróny so sigmoidálnou aktivačnou funkciou. Zatiaľ bude výklad venovaný opisu štruktúry siete určenej pre regresné úlohy. Výstupy jednotlivých expertných modulov sú zhodné s aktivačnými hodnotami neurónov v týchto moduloch a ich hodnoty výsledkom násobenia vstupného vektora \mathbf{x} vektormi váh \mathbf{w}_i prislúchajúcich k jednotlivým modulom

$$\mathbf{y}_i = \mathbf{x}^T \mathbf{w}_i \quad (4.3)$$

Aktivačná funkcia neurónov bránového modulu je tiež lineárna. Vektor \mathbf{v} je vektor váh bránovej siete. Vektor aktivačných hodnôt \mathbf{a} sa vypočíta podľa vzťahu

$$\mathbf{a} = \mathbf{x}^T \mathbf{v} \quad (4.4)$$

Keďže pri tvorbe základného modelu bol zvolený *štatistický prístup*¹, výstupné hodnoty g_i neurónov bránovej siete sú potom upravené tak, aby nadobúdali hodnoty z intervalu $\langle 0, 1 \rangle$ a ich súčet bol rovný jednej. Táto požiadavka je zabezpečená výstupnou funkciou typu *softmax* tvaru

$$g_i = \frac{\exp(a_i)}{\sum_{j=1}^K \exp(a_j)} \quad (4.5)$$

¹Podobnú tématiku sa zaoberali Szymanski a Lemmon z informačno-teoretického hľadiska.

Použitie funkcie softmax v tomto prípade dovoľuje interpretovať hodnoty g_i ako pravdepodobnosti. Hodnota g_i tak určuje apriórnu pravdepodobnosť, že i -ty modul generoval aktuálnu učiacu vzorku.

4.2 Pravdepodobnostná funkcia

Neurónovú sieť možno chápať aj ako prostriedok pre štatistické modelovanie a predikciu. Z tohto hľadiska je činnosť naučenej neurónovej siete úspešná, ak modeluje proces, pomocou ktorého boli učiace vzorky generované. Veľkosť chyby v procese učenia preto nemôže byť postačujúcim ukazovateľom kvality naučenia siete.

Najvšeobecnejší a kompletný popis vzoriek učiacej množiny je možný podľa funkcie rozdelenia ich pravdepodobnosti $p(\mathbf{x}, \mathbf{d})$ [21]. Cieľom učiaceho algoritmu aplikovaného na danú architektúru je modelovanie pravdepodobnostného rozdelenia množiny učiacich vzoriek $\{\mathbf{x}, \mathbf{d}\}$ alebo aj maximalizácia funkcie rozdelenia pravdepodobnosti $l = p(\mathbf{x}, \mathbf{d})$. Podľa [21] môže byť maximalizácia logaritmickkej pravdepodobnostnej funkcie l interpretovaná ako minimalizácia chybovej funkcie J doprednej siete

$$J = -\ln l \quad (4.6)$$

Ak $p(\mathbf{x})$ predstavuje nepodmienujúcu pravdepodobnosť vstupu a $p(\mathbf{d}|\mathbf{x})$ je podmienená pravdepodobnosť výstupu, ktorá je podmienená vstupným vektorom \mathbf{x} , potom sa funkcia rozdelenia pravdepodobnosti vypočíta ako súčin hodnôt týchto dvoch pravdepodobností

$$p(\mathbf{x}, \mathbf{d}) = p(\mathbf{d}|\mathbf{x})p(\mathbf{x}) \quad (4.7)$$

Pri ďalšom odvodzovaní učiaceho algoritmu je výhodnejšie pracovať s prirodzeným logaritmom výrazu (4.7). Je tak možné urobiť, pretože funkcia logaritmus je monotónne rastúca na celom definičnom obore.

Logaritmická pravdepodobnostná funkcia je definovaná nasledovne:

$$l(\theta) = \ln p(\mathbf{d}|\mathbf{x})p(\mathbf{x}) \quad (4.8)$$

Hodnota pravdepodobnostnej funkcie závisí od množiny hodnôt voľných parametrov siete θ . Typ podmienenej pravdepodobnosti $p(\mathbf{d}|\mathbf{x})$ závisí od typu úlohy, pre ktorý je modulárna sieť určená.

- *Regresia:* Učiace údaje sú spojité z oboru reálnych hodnôt. Rozdelenie pravdepodobnosti výstupných hodnôt je možné opísať pomocou Gaussovho rozdelenia pravdepodobnosti

$$p(\mathbf{d}|\mathbf{x}) = \frac{1}{\sqrt[q]{2\pi} \sqrt{|\mathbf{\Lambda}|}} \exp\left(-\frac{1}{2}(\mathbf{d} - \mathbf{y})^T \mathbf{\Lambda}^{-1}(\mathbf{d} - \mathbf{y})\right) \quad (4.9)$$

\mathbf{d} - požadovaný výstupný vektor hodnôt

\mathbf{y} - výstupný vektor hodnôt expertnej siete

$\mathbf{\Lambda}$ - matica kovariancií

q - počet neurónov expertnej siete

- *Klasifikácia:* Výstupné hodnoty klasifikátora nadobúdajú diskkrétne hodnoty. Podľa [21] je v takom prípade potrebné pre ich opis použiť Bernoulliho rozdelenie pravdepodobnosti.

$$p(\mathbf{d}|\mathbf{x}) = \prod_{i=1}^q y_i^{d_i} (1 - y_i)^{1-d_i} \quad (4.10)$$

d_i - požadovaná výstupná hodnota i -teho neurónu expertnej siete

y_i - výstupná hodnota i -teho neurónu expertnej siete

q - počet neurónov expertnej siete

Za predpokladu, že kovariančná matica $\mathbf{\Lambda}$ vo výraze (4.9) je jednotková, je možné hodnotu argumentu funkcie *exp* v Gaussovskom rozdelení pravdepodobnosti vypočítať ako Euklidovskú normu vektora.

$$p(\mathbf{d}|\mathbf{x}) = \frac{1}{\sqrt[q]{2\pi}} \exp\left(-\frac{1}{2}\|\mathbf{d} - \mathbf{y}\|^2\right) \quad (4.11)$$

Dosadením výrazu (4.11) do (4.7) pre všetky K expertné siete sa získa vzorec pre výpočet celkovej podmienenej pravdepodobnosti.

$$p(\mathbf{d}|\mathbf{x}) = \frac{1}{\sqrt[q]{2\pi}} \sum_{i=1}^K g_i \exp\left(-\frac{1}{2}\|\mathbf{d} - \mathbf{y}_i\|^2\right) \quad (4.12)$$

Dosadením výrazu (4.12) do (4.8) pri zanedbaní konštanty $-\ln\sqrt[q]{2\pi}$ sa získa konečný tvar logaritmickej pravdepodobnostnej funkcie pre regresiu.

$$l_R(\theta) = \ln \sum_{i=1}^K g_i \exp\left(-\frac{1}{2}\|\mathbf{d} - \mathbf{y}_i\|^2\right) \quad (4.13)$$

Toto rozdelenie pravdepodobnosti je uvedené v [5] pod názvom *Gaussov zmesový model*. (Gaussian mixture model). Logaritmicкую pravdepodobnostnú funkciu pre klasifikáciu možno vyjadriť podobne v tvare

$$l_K(\theta) = \ln \sum_{i=1}^K g_i \prod_{j=1}^q y_{ij}^{d_{ij}} (1 - y_{ij})^{1-d_{ij}} \quad (4.14)$$

Pravdepodobnostné funkcie (4.13) a (4.14) budú ďalej základom pre odvádzanie učiacich algoritmov pre regresiu a klasifikáciu.

4.3 Učiaci algoritmus pre regresiu

Odvodenie učiaceho algoritmu je založené na metóde najstrmšieho *vzostupu* hodnoty logaritmickej pravdepodobnostnej funkcie $l_R(\theta)$. Cieľom učenia je maximalizácia hodnoty tejto funkcie.

Postup je analogický metóde najstrmšieho zostupu hodnoty chybovej funkcie $J(\Theta)$ doprednej siete. Jej veľkosť závisí od matice Θ , ktorá predstavuje množinu voľných parametrov doprednej siete. V procese učenia sa upravujú hodnoty prvkov matice Θ tak, aby sa hodnota chybovej funkcie $J(\Theta)$ znižovala. Nové hodnoty prvkov matice Θ sa v n -tom kroku učiaceho procesu vypočítajú podľa vzťahu

$$\Theta(n+1) = \Theta(n) + \Delta\Theta(n) \quad (4.15)$$

Hodnota $\Delta\Theta$ sa v n -tom kroku vypočíta podľa vzťahu

$$\Delta\Theta = -\gamma \frac{\partial J(\Theta)}{\partial \Theta} \quad (4.16)$$

V modulárnej architektúre opísanej v predošlej časti predstavujú množinu voľných parametrov vektory váh \mathbf{w}_i expertných modulov a \mathbf{v} bránového modulu. Od ich hodnôt závisí veľkosť logaritmickej pravdepodobnostnej funkcie $l_R(\theta)$.

4.3.1 Úprava váh expertných modulov

Modulárna sieť podľa schémy na Obr. 4.1 pozostáva z K expertných modulov a jedného bránového modulu. Každý expertný modul obsahuje jednu vrstvu pozostávajúcu z q lineárnych neurónov. Ich výstupné hodnoty tvoria vektor výstupných hodnôt \mathbf{y}_i i -teho modulu ($i = 1 \dots K$). Hodnoty výstupného vektora určené podľa vzťahu (4.3) závisia od matice vektorov hodnôt váh \mathbf{w}_i . Prvok \mathbf{w}_{ij} tejto matice predstavuje *vektor váh* prislúchajúci j -temu neurónu i -teho expertného modulu. Nové hodnoty tohto vektora sa v n -tom kroku učiaceho procesu vypočítajú analogicky ako vo vzťahoch (4.15) a (4.16).

$$\mathbf{w}_{ij}(n+1) = \mathbf{w}_{ij}(n) + \Delta\mathbf{w}_{ij}(n) \quad (4.17)$$

$$\Delta\mathbf{w}_{ij} = \gamma \frac{\partial l_R(\mathbf{w}_{ij})}{\partial \mathbf{w}_{ij}} \quad (4.18)$$

Rozdiel oproti metóde najstrmšieho zostupu chybovej funkcie je v znamienku na pravej strane výrazu (4.18). Cieľom pri metóde najstrmšieho vzostupu pravdepodobnostnej funkcie je zväčšovanie jej hodnoty. Deriváciu funkcie l_R podľa \mathbf{w}_{ij} možno podľa pravidla zretáženia prepísať do tvaru

$$\frac{\partial l_R}{\partial \mathbf{w}_{ij}} = \frac{\partial l_R}{\partial y_{ij}} \frac{\partial y_{ij}}{\partial \mathbf{w}_{ij}} \quad (4.19)$$

Výpočet zložitej derivácie sa rozdelil na výpočet dvoch jednoduchších. Derivácia l_R podľa y_{ij} sa vypočíta nasledovne

$$\frac{\partial l_R}{\partial y_{ij}} = h_i(d_j - y_{ij}) \quad (4.20)$$

y_{ij} - výstupná hodnota j -teho neurónu i -tej expertnej siete.

d_j - požadovaná hodnota j -teho prvku celkového výstupného vektora \mathbf{d}

h_i - aposteriórna pravdepodobnosť, že i -ty expertný modul generuje požadovaný výstupný vektor \mathbf{d} . Jej hodnota je daná vzťahom

$$h_i = \frac{g_i \exp(-\frac{1}{2}\|\mathbf{d} - \mathbf{y}_i\|^2)}{\sum_{j=1}^K g_j \exp(-\frac{1}{2}\|\mathbf{d} - \mathbf{y}_j\|^2)} \quad (4.21)$$

Zo vzťahu (4.21) vyplýva, že všetky $h_{1\dots K}$ splňajú podmienky

$$h_i \in \langle 0, 1 \rangle \quad (4.22)$$

$$\sum_{i=1}^K h_i = 1$$

Keďže aktivačné aj výstupné funkcie neurónov v expertných moduloch sú lineárne, hodnota y_{ij} sa podľa vzťahu (4.3) vypočíta nasledovne

$$y_{ij} = \mathbf{x}^T \mathbf{w}_{ij} \quad (4.23)$$

Deriváciou výrazu (4.23) podľa \mathbf{w}_{ij} sa získa druhá časť výrazu (4.19).

$$\frac{\partial y_{ij}}{\partial \mathbf{w}_{ij}} = \mathbf{x} \quad (4.24)$$

Dosadením (4.20) a (4.24) do (4.19) a dosadením do (4.18) sa získa vzorec pre výpočet hodnoty $\Delta \mathbf{w}_{ij}$

$$\Delta \mathbf{w}_{ij} = \gamma h_i(d_j - y_{ij})\mathbf{x} \quad (4.25)$$

kde γ predstavuje veľkosť učiaceho parametra. Výsledný vzorec pre výpočet novej hodnoty vektora váh \mathbf{w}_{ij} , ktorý prislúcha j -temu neurónu i -teho expertného modulu má tvar

$$\mathbf{w}_{ij}(n + 1) = \mathbf{w}_{ij}(n) + \gamma h_i(n)(d_j(n) - y_{ij}(n))\mathbf{x} \quad (4.26)$$

4.3.2 Úprava váh bránového modulu

Bránový modul obsahuje K nelineárnych neurónov s výstupnou funkciou typu softmax. Výpočet hodnôt g_i je definovaný vzťahom (4.5), ktorého dosadením do (4.13) je určené vyjadrenie pravdepodobnostnej funkcie pomocou vektora váh bránového modulu \mathbf{v}

$$l_R(\theta) = \ln \sum_{i=1}^K \exp(a_i) \cdot \exp\left(-\frac{1}{2}\|\mathbf{d} - \mathbf{y}_i\|^2\right) - \ln \sum_{j=1}^K \exp(a_j) \quad (4.27)$$

Hodnoty $a_{1...K}$ sú výstupnými hodnotami neurónov bránového modulu, ktoré sa vypočítajú podľa vzťahu (4.4).

Ďalší postup odvodenia úpravy hodnôt vektora váh \mathbf{v} je analogický postupu odvodenia úpravy váh pre expertné moduly. Rozdiel je v počte a type neurónov bránového modulu. Expertné moduly obsahovali q lineárnych neurónov. Bránový modul obsahuje K nelineárnych neurónov. Nové hodnoty vektora váh \mathbf{v}_i , ktorý prislúcha i -temu neurónu sa určia nasledovne

$$\mathbf{v}_i(n + 1) = \mathbf{v}_i(n) + \Delta \mathbf{v}_i(n) \quad (4.28)$$

$$\Delta \mathbf{v}_i = \gamma \frac{\partial l_R(\mathbf{v}_i)}{\partial \mathbf{v}_i} \quad (4.29)$$

Učiaci parameter γ bránovej siete môže byť zhodný s učiacim parametrom pre expertné siete. Parciálnu deriváciu pravdepodobnostnej funkcie je možné znova rozpísať pomocou pravidla zreťazenia.

$$\frac{\partial l_R}{\partial \mathbf{v}_i} = \frac{\partial l_R}{\partial a_i} \frac{\partial a_i}{\partial \mathbf{v}_i} \quad (4.30)$$

Derivácia pravdepodobnostnej funkcie podľa a_i vyjadrená pre i -ty neurón bránovej siete má tvar

$$\frac{\partial l_R}{\partial a_i} = h_i - g_i \quad (4.31)$$

Derivácia a_i podľa vektora váh prislúchajúceho i -temu neurónu bránovej siete má tvar

$$\frac{\partial a_i}{\partial \mathbf{v}_i} = \mathbf{x} \quad (4.32)$$

Dosadením (4.31) a (4.32) do (4.30) a následným dosadením do (4.28) sa získa vzorec pre výpočet hodnoty $\Delta \mathbf{v}_i$

$$\Delta \mathbf{v}_i = \gamma(h_i - g_i)\mathbf{x} \quad (4.33)$$

Výsledný vzorec pre výpočet novej hodnoty vektora váh prislúchajúceho i -temu neurónu bránového modulu má tvar

$$\mathbf{v}_i(n+1) = \mathbf{v}_i(n) + \gamma(h_i(n) - g_i(n))\mathbf{x} \quad (4.34)$$

Zhrnutie učiaceho algoritmu pre regresiu:

1. *Inicializácia.*

Hodnoty všetkých váh celej siete sa nastavujú na náhodnú hodnotu z malého intervalu, napr. $\langle -1, 1 \rangle$.

2. *Úprava váh.*

Úprava váh sa uskutočňuje v N cykloch. V každom cykle sa privedú na vstup a výstup siete všetky vzorky tréningovej množiny. Každá vzorka je reprezentovaná dvojicou $\{\mathbf{x}, \mathbf{d}\}$. Pre jednotlivé indexy platí:

$$n \in \langle 0, 1, 2, \dots, N \rangle$$

$$i \in \langle 1, 2, \dots, K \rangle$$

$$j \in \langle 1, 2, \dots, q \rangle$$

(a)

$$a_i(n) = \mathbf{x}^T \mathbf{v}_i(n)$$

$$g_i(n) = \frac{\exp(a_i(n))}{\sum_{j=1}^K \exp(a_j(n))}$$

(b)

$$y_{ij}(n) = \mathbf{x}^T \mathbf{w}_{ij}(n)$$

(c)

$$h_i(n) = \frac{g_i(n) \exp(-\frac{1}{2} \|\mathbf{d} - \mathbf{y}_i(n)\|^2)}{\sum_{j=1}^K g_j(n) \exp(-\frac{1}{2} \|\mathbf{d} - \mathbf{y}_j(n)\|^2)}$$

(d)

$$\mathbf{w}_{ij}(n+1) = \mathbf{w}_{ij}(n) + \gamma h_i(n) (d_j(n) - y_{ij}(n)) \mathbf{x}$$

(e)

$$\mathbf{v}_i(n+1) = \mathbf{v}_i(n) + \gamma (h_i(n) - g_i(n)) \mathbf{x}$$

4.4 Učiaci algoritmus pre klasifikáciu

Výstupné hodnoty klasifikátora nadobúdajú diskkrétne hodnoty. Ich popis je možný pomocou Bernoulliho rozdelenia pravdepodobnosti. Pravdepodobnostnú funkciu $l_K(\theta)$ potom možno vyjadriť podľa vzťahu (4.14). Neuróny v expertných a bránovom module nie sú lineárne. Výstupná hodnota j -teho neurónu i -teho expertného modulu je určená sigmoidálnou funkciou

$$y_{ij} = \frac{1}{1 + \exp(-\mathbf{x}^T \mathbf{w}_{ij})} \quad (4.35)$$

Podobne je určená aj aktivačná hodnota i -teho neurónu bránového modulu

$$a_i = \frac{1}{1 + \exp(-\mathbf{x}^T \mathbf{v}_i)} \quad (4.36)$$

Výstupné hodnoty g_i neurónov bránového modulu sú určené nelinearitou typu softmax, ale bez funkcie *exp*

$$g_i = \frac{a_i}{\sum_{j=1}^K a_j} \quad (4.37)$$

Hodnoty y_{ij} a g_i spĺňajú nasledujúce podmienky:

$$y_{ij}, g_i \in \langle 0, 1 \rangle \quad (4.38)$$

$$\prod_{j=1}^q y_{ij} \in \langle 0, 1 \rangle$$

$$\sum_{i=1}^K g_i = 1$$

Podľa týchto ohraňení sa interpretuje činnosť bránového modulu ako klasifikácia nad celým definičným oborom aproximovanej funkcie. Taktó sa vstupný priestor vzoriek rozdelí na viacej oblastí. Činnosť expertných

modulov sa interpretuje ako klasifikácia vo vnútri jednotlivých oblastí vstupného priestoru vzoriek. Na tomto mieste je potrebné podotknúť, že výstupné vektory učiacich vzoriek sú kódované podľa pravidla 1-z-T. Ak aktuálna učiaca vzorka patrí do triedy t , výstupný vektor potom obsahuje jednu jednotku na pozícii t a zvyšné prvky vektora sú nulové. Hodnota g_i sa interpretuje ako pravdepodobnosť, že i -ty expertný modul generoval aktuálnu učiacu vzorku. Hodnota y_{ij} sa interpretuje ako pravdepodobnosť, že i -ty modul klasifikuje aktuálnu vzorku do triedy j .

Ďalší postup odvodenia učiaceho algoritmu je analogický postupu pre odvodenie učiaceho algoritmu pre regresiu. Rozdiel je pri výpočte aposteriorných pravdepodobností h_i pre potreby klasifikácie. ²

$$h_i = \frac{g_i \prod_{k=1}^q y_{ik}^{d_{ik}} (1 - y_{ik})^{1-d_{ik}}}{\sum_{j=1}^K g_j \prod_{k=1}^q y_{jk}^{d_{jk}} (1 - y_{jk})^{1-d_{jk}}} \quad (4.39)$$

Derivácie y_{ij} a a_i podľa vektorov váh, ktoré k nim prislúchajú, sú tvaru

$$\frac{\partial y_{ij}}{\partial \mathbf{w}_{ij}} = \frac{\exp(-\mathbf{x}^T \mathbf{w}_{ij})}{(1 + \exp(-\mathbf{x}^T \mathbf{w}_{ij}))^2} \mathbf{x} \quad (4.40)$$

$$\frac{\partial a_i}{\partial \mathbf{v}_i} = \frac{\exp(-\mathbf{x}^T \mathbf{v}_i)}{(1 + \exp(-\mathbf{x}^T \mathbf{v}_i))^2} \mathbf{x} \quad (4.41)$$

Vzťahy pre úpravu hodnôt váh expertných a bránového modulu sú podobné vzťahom (4.26) a (4.34). *Zhrnutie učiaceho algoritmu pre klasifikáciu:*

1. *Inicializácia.*

Hodnoty všetkých váh celej siete sa nastavujú na náhodnú hodnotu z malého intervalu napr. $\langle -1, 1 \rangle$.

2. *Úprava váh.*

Úprava váh sa uskutočňuje v N cykloch. V každom cykle sa pri-

²Pri programovej realizácii tohto algoritmu sa výpočet aposteriornej pravdepodobnosti expertného modulu nepočíta ako súčin hodnôt výstupných neurónov umocnených na požadovanú hodnotu. Experimenty ukázali, že je výhodnejšie použiť ich priemernú hodnotu.

vedú na vstup a výstup siete všetky vzorky trénovacej množiny. Každá vzorka je reprezentovaná dvojicou $\{\mathbf{x}, \mathbf{d}\}$.

(a)

$$a_i(n) = \frac{1}{1 + \exp(-\mathbf{x}^T \mathbf{v}_i(n))}$$

$$g_i(n) = \frac{a_i(n)}{\sum_{j=1}^K a_j(n)}$$

(b)

$$y_{ij}(n) = \frac{1}{1 + \exp(-\mathbf{x}^T \mathbf{w}_{ij}(n))}$$

(c)

$$h_i(n) = \frac{g_i(n) \prod_{k=1}^q y_{ik}^{d_{ik}} (1 - y_{ik})^{1-d_{ik}}(n)}{\sum_{j=1}^K g_j(n) \prod_{k=1}^q y_{jk}^{d_{jk}} (1 - y_{jk})^{1-d_{jk}}(n)}$$

(d)

$$\mathbf{w}_{ij}(n+1) = \mathbf{w}_{ij}(n) + \gamma h_i(n) (d_j(n) - y_{ij}(n)) \frac{\exp(-\mathbf{x}^T \mathbf{w}_{ij})}{(1 + \exp(-\mathbf{x}^T \mathbf{w}_{ij}))^2} \mathbf{x}$$

(e)

$$\mathbf{v}_i(n+1) = \mathbf{v}_i(n) + \gamma (h_i(n) - g_i(n)) \frac{\exp(-\mathbf{x}^T \mathbf{v}_i)}{(1 + \exp(-\mathbf{x}^T \mathbf{v}_i))^2} \mathbf{x}$$

4.5 EM algoritmus

Použitie modulárnej architektúry predpokladá zvládnutie problému rozdelenia jednotlivých učiacich vzoriek medzi expertné moduly a sformovanie celkového výstupu. Algoritmus *Expectation Maximisation* - EM je alternatívnym riešením tohto problému k už uvedenému spôsobu metódou najstrmšieho vzostupu pravdepodobnostnej funkcie.

EM algoritmus vznikol v roku 1977 a bol pôvodne využívaný pre nekontrolované učenie v kontexte zhlukovania. Veľmi rozšírené je jeho použitie v oblasti matematiky. Použitie pre kontrolované učenie spolu s modulárnou architektúrou uvádzajú Jacobs a Jordan v [8]. Ide o úpravu voľných parametrov siete za účelom maximalizácie pravdepodobnostnej funkcie pomocou algoritmu EM.

Nech \mathcal{Z} je množina binárnych vektorov $\mathbf{z}_{1\dots M}$, pričom M je počet vzoriek trérovacej množiny \mathcal{X} . Každý vektor množiny \mathcal{Z} je K -prvkový, pričom K je počet expertných modulov. Prvok $z_{ij} = 1$ ak je i -ta vzorka generovaná j -tým expertným modulom. Pridaním množiny \mathcal{Z} k trérovacej množine \mathcal{X} vznikne nová množina \mathcal{Y} . Takýmto rozšírením trérovacej množiny vznikne *kompletná* pravdepodobnostná funkcia $l_C(\theta; \mathcal{Y})$. Ak by bola množina \mathcal{Z} naozaj známa, optimalizácia kompletnej pravdepodobnostnej funkcie by bola jednoduchšia. Keďže množina \mathcal{Z} je neznáma, EM algoritmus je rozdelený na dve časti. Najprv sa nájde očakávaná hodnota kompletnej pravdepodobnostnej funkcie. V druhom kroku sa hľadajú také hodnoty parametrov θ , ktoré zväčšujú jej hodnotu. *E-krok*:

$$Q(\theta, \theta_n) = E[l_C(\theta; \mathcal{Y}) | \mathcal{X}] \quad (4.42)$$

M-krok:

$$\theta_{n+1} = \arg \max_{\theta} Q(\theta, \theta_n) \quad (4.43)$$

Jacobs a Jordan v [8] uvádzajú, že zväčšením hodnoty kompletnej pravdepodobnostnej funkcie sa dosiahne aj zväčšenie hodnoty pôvodnej pravdepodobnostnej funkcie oproti predošlému kroku výpočtu. Každá iterácia tak zväčší hodnotu $l(\theta; \mathcal{X})$

$$l(\theta_{n+1} | \mathcal{X}) \geq l(\theta_n | \mathcal{X}) \quad (4.44)$$

Hodnota pravdepodobnostnej funkcie $l(\theta; \mathcal{X})$ sa monotónne zväčšuje spolu s postupnosťou odhadov parametrov θ , ktoré sú generované EM algoritmom. To je podľa [8] príčinou konvergencie do *lokálneho maxima*. Napriek výhodám oproti metóde najstrmšieho vzostupu nie je EM algoritmus odolný voči takejto nežiadúcej konvergencii. Konkrétna implementácia pre úlohy klasifikácie je prevzatá z [20]. Architektúra siete je zhodná s architektúrou podľa Obr. 4.1. Všetky siete majú v

tomto prípade výstupnú funkciu typu softmax. Výstupná hodnota j -teho neurónu i -tej expertnej siete je určená vzťahom

$$y_{ij} = \frac{\exp(\mathbf{x}^T \mathbf{w}_{ij})}{\sum_{k=1}^q \exp(\mathbf{x}^T \mathbf{w}_{ik})} \quad (4.45)$$

Podobne sú určené výstupné hodnoty a_i neurónov bránovej siete.

$$a_i = \frac{\exp(\mathbf{x}^T \mathbf{v}_i)}{\sum_{j=1}^K \exp(\mathbf{x}^T \mathbf{v}_j)} \quad (4.46)$$

Pre všetky expertné moduly potom platí

$$\prod_{k=1}^q y_{ik} = 1 \quad (4.47)$$

Hodnota y_{it} sa môže interpretovať ako pravdepodobnosť, že i -ty modul klasifikuje aktuálnu učiacu vzorku do triedy t . Waterhouse v [20] uvádza, že ak aktuálna učiacia vzorka patrí do triedy t , potom platí

$$h_i = y_{it} \quad (4.48)$$

pričom h_i je aposteriórna pravdepodobnosť, že i -ty expertný modul klasifikuje aktuálnu učiacu vzorku do triedy t . Zanedbávajú sa tým hodnoty zvyšných neurónov expertného modulu. Je tak možné urobiť, ak sú výstupné vektory učiacich vzoriek kódované podľa pravidla 1-z- T . Pre úpravu hodnôt vektora váh, ktorý prislúcha j -temu neurónu i -teho expertného modulu platí

$$\mathbf{w}_{ij}(n+1) = \mathbf{w}_{ij}(n) + \gamma \frac{1}{\sum_{m=1}^M h_i(\mathbf{m})} \sum_{m=1}^M h_i(d_j - y_{ij}) \mathbf{x}(\mathbf{m}) \quad (4.49)$$

Pre úpravu hodnôt vektora váh, ktorý prislúcha i -temu neurónu bránového modulu platí

$$\mathbf{v}_i(n+1) = \mathbf{v}_i(n) + \gamma \frac{1}{\sum_{m=1}^M h_i(\mathbf{m})} \sum_{m=1}^M h_i(h_i - a_i) \mathbf{x}(\mathbf{m}) \quad (4.50)$$

Z uvedených vzťahov vyplýva, že úprava hodnôt váh sa vykonáva iba raz na konci každého učiaceho cyklu. Pri metóde najstrmšieho vzostupu sa hodnoty váh upravujú pri prezentácii každej vzorky. Učenie pomocou EM algoritmu je preto z hľadiska behu v reálnom čase v porovnaní s touto metódou rýchlejšie. Podľa predpokladov by učenie pomocou EM algoritmu malo byť rýchlejšie aj z hľadiska počtu učiacich cyklov potrebných na dosiahnutie minimálnej chyby učenia.

Učenie NN podľa stavu systému

V predchádzajúcich kapitolách tohto dielu sme sa venovali, buď **kontrolovanému** alebo **nekontrolovanej učeni**u na RC NN. Ak použijeme NN na riadenie nejakého systému, tak v podstate nám ide v o to, aby správanie **riadeného systému** bolo podľa stanovených požiadaviek. Teda v takýchto prípadoch **výstup** riadeného systému, by mal ovplyvňovať samotnú činnosť NN. Na takejto logike je založené **učenie podľa stavu riadeného systému** v svetovej literatúre označované ako **reinforcement learning**. Tieto metódy učenia si vyžadujú špeciálnu topológiu NN, ktorá obsahuje dve subsiete. Z globálneho pohľadu, nie je možné vhodne kategorizovať túto metódu učenia, a preto sa jej venuje špeciálna kapitola. Celá filozofia je postavené na **odmene alebo pokute**¹. Ak napríklad, chceme aby riadený systém ostal v nejakej množine parametrov α , tak potom za pomoci tzv. **reinforcement signálu "r"**, ktorý nadobúda hodnoty

$$r = \begin{cases} 0 & \forall s \in \alpha \\ -1 & \text{inak} \end{cases} \quad (5.1)$$

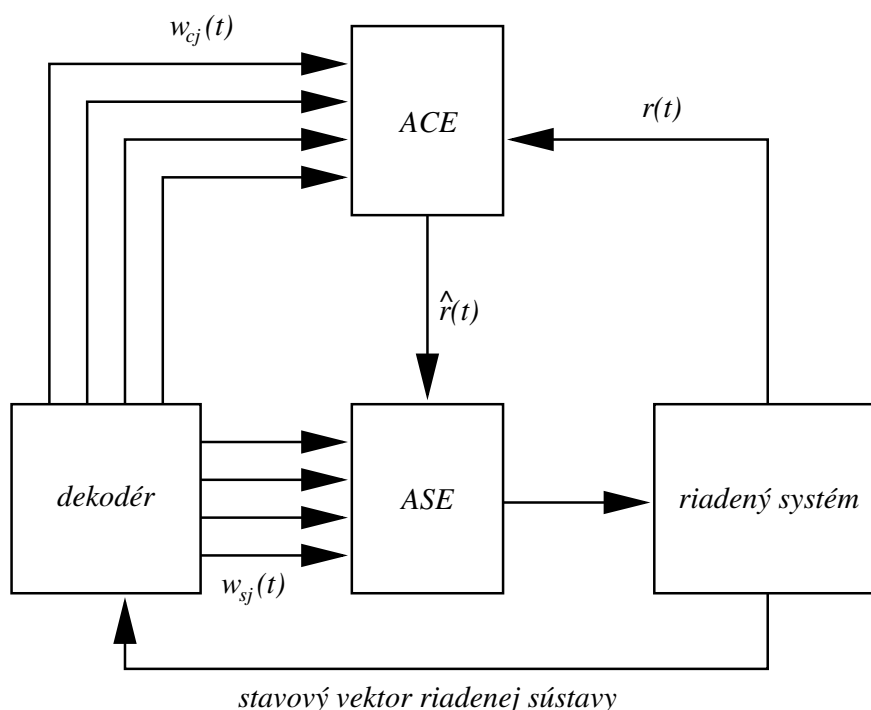
kde $r = -1$ predstavuje signál pokuty.

5.1 Topológia NN

Celá topológia sa skladá z dvoch subsietí (viď Obr. 5.1) a to

¹typ úlohy sa tiež nazýva - credit assignent problem

- Associative search network ASE² - táto časť stanovuje vzťah medzi vstupom a výstupom do systému. Do subsiete ASE vstupujú **vstupy**, ktoré predstavujú stavový vektor "s" riadeného systému.
- Adaptive Critic Element ACE³ - na základe výstupu zo systému, stanovuje tzv. predikčnú hodnotu reinforcement parametra (\hat{r}) v čase $(t + 1)$ na základe parametrov $r(t)$ a $r(t - 1)$.



Obr. 5.1: Topológia NN určená pre učenie podľa stavu riadeného systému

Samotné bloky ASE a ACE môžu predstavovať buď subsieť ale aj samostný neurón. Okrem topológie na Obr. 5.1 je možné, ešte použiť topológiu, kde máme jednu subsieť, ktorá predstavuje **model** riadeného

²pridružená pre hľadanie optimálneho parametra riadenia

³zhodnocovacia subsieť, ktorá vyhodnocuje výstup zo systému

systému. Teda v konečnom dôsledku pôjde o riadenie pomocou modelu [6].

5.2 Adaptačné pravidlo pre ASE

Samotný blok ASE produkuje binárne výstupy $\{0, 1\}$ a vstupom do ASE je stavový vektor "s" riadeného systému. Teda vstup do ASE môžeme vyjadriť ako

$$in_{s_{ASE}}(t) = \sum_{j=1}^M w_{sj}x_j(t) + \theta_j \quad (5.2)$$

kde však hodnota prahu θ_j má pravdepodobnostný charakter podľa normálneho rozdelenia s nulovou strednou hodnotou. Aktivačná funkcia vzhľadom na to, že výstupná funkcia je jednotková má tvar :

$$out_{ASE}(t) = \begin{cases} 1 & ak \ in_{s_{ASE}} > 0 \\ 0 & inak \end{cases} \quad (5.3)$$

. Samotné adaptačné pravidlo pre zmenu SV $w_{s_{ASE}}$ má charakter Hebových SV, teda $out(t)x(t)$ ale ešte predtým, je to prepočítané, cez tzv. **koeficient spoľahlivosti** es_j a váhované predikčným signálom \hat{r} . Teda potom

$$\Delta w_{sj}(t) = \alpha \hat{r} es_j(t) \quad (5.4)$$

kde potom

$$es_j(t) = \beta es_j(t-1) + (1 - \beta) out(t)x_j(t) \quad (5.5)$$

kde β je zostupný pomer spoľahlivosti. Stavový vektor vstupuje do ASE veľmi často cez tzv. dekodér, ktorý vyberá z množstva možných stavov len ten aktuálny.

V práci [1] je dokázaná konvergencia takéhoto adaptačného pravidla.

5.3 Adaptačné pravidlo pre ACE

Subsieť ACE je taktiež nazývaná ako hodnotiacia sieť. Vstupom do ACE je stavový vektor riadenej sústavy a reinforcement signál "r".

Výstupom z ACE je tzv. predikcia "r" v tvare \hat{r}^4 . Hlavnou úlohou je teda vyprodukovať predikciu signálu v tvare \hat{r} na základe doposiaľ získaných skúsenosti. V podstate môžeme definovať hodnotiacu funkciu na základe reinforcement signálov "r" v tvare strednej hodnoty:

$$J(t) = E \left\{ \sum_{k=0}^t \chi^k r(k+1) \right\} \quad (5.6)$$

kde χ je tzv. pomerový parameter zľavy⁵ a je z intervalu $(0, 1)$. Z toho môžeme odvodiť tvar $J(t)$ v prípade korektného učenia ako

$$J(t) = r(t-1) + \chi J(t-1) \quad (5.7)$$

a potom v konečnom dôsledku môže byť predikčný signál definovaný ako

$$\hat{r}(t) = r(t) + \chi J(t) - J(t-1) \quad (5.8)$$

Ak budeme uvažovať pod hodnotiacou funkciou jednotlivé SV w_{cj} , tak potom ich zmena bude mať prírastok Hebbovského typu

$$\Delta w_{cj}(t) = \gamma \hat{r}(t) h_j(t) \quad (5.9)$$

kde γ je učiaci pomer a koeficient $h_j(t)$ má tvar

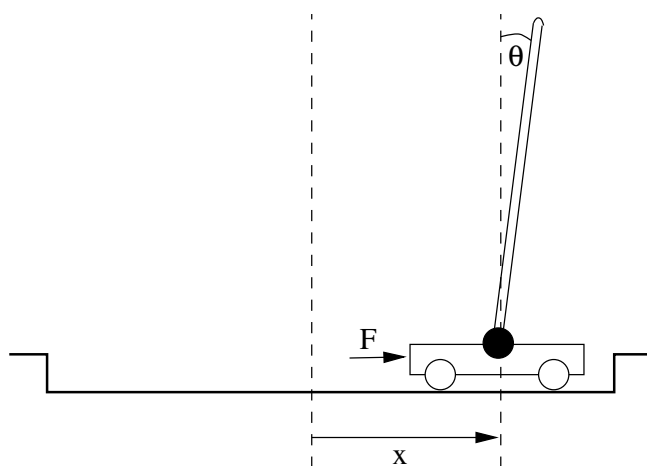
$$h_j(t) = \tau h_j(t-1) + (1 - \tau) x_j(t-1) \quad (5.10)$$

Teda samotný vzorec (5.9) predstavuje adaptačný mechanizmus SV pre subsieť ACE.

Táto metóda učenia na NN so špecializovanými topológiami sa vo veľkej miere využíva a má veľké ambície využitia pri riadení v reálnom čase. Najčastejším demonštratívnym príkladom je riadenie vozíka so vspríamenou tyčou upevnenou v otočnom kĺbe na pohybujúcom sa vozíku. Cieľom je riadiť vozík v reálnom čase tak, aby tyč bola stále vzpriamená (viď Obr. 5.2).

⁴Niekedy sa taktiež nazýva tzv. heuristický reinforcement signál

⁵discount rate parameter



Obr. 5.2: Riadený systém

5.4 Repetitorium č. 4

Tematické okruhy pre štúdium

- **5. tutoriál**

1. Vysvetlite základnu topológiu Hopfieldovej siete.
2. Aké sú aplikačné možnosti Hopfieldovej siete?
3. Stručne charakterizujte TSP problém.
4. Vysvetlite význam RC-modelu neurónu.
5. Vysvetlite základny princíp BP na RC sieti, význam $x(\infty)$ a $y(\infty)$.
6. Čo predstavuje indexová funkcia δ_{ij} a čo δ_i .
7. Charakterizujte postup činnosti BP na RC NN.
8. Stručne popíšte topológiu ART, je to kontrolované učenie na RC ?
9. Aké sú základné kroky jej činnosti?
10. Aké existujú iné varianty ART-učenia?
11. Popíšte základné charakteristiky učenia podľa stavu systému **reinforcement learning**.
12. Aké základné časti má táto NN? Aké majú funkcie?
13. Aké sú adaptačné pravidlá pre túto NN?
14. Charakterizujte modulárne NN, akú stratégiu používajú?
15. Aké sú základné typy časti modulárnych NN?
16. Charakterizujte jednoduchšiu a zložitejšiu MNN.

Poznámky

Poznámky

Záver

6.1 Stručné zhodnotenie zámeru autorov

Cieľom knihy bolo podať základne informácie z oblasti teoretických základov neurónových sietí. Výklad by mal byť dostatočne detailný na to, aby čitatelia vnikli do problematiky NN natoľko, aby mohli buď pokračovať v štúdiu teórie základov NN, alebo mohli tieto základy použiť pri aplikačných projektoch. Bola snaha o výklad, ktorý je blízky inžinierskym disciplinám. Inšpiráciou bola hlavne publikácia [5] vydaná v roku 1994, ktorá prezentuje inžiniersky prístup k problematike NN.

Dominantnou časťou 1. dielu knihy sú dopredné NN. Kontrolované a nekontrolované učenie na dopredných NN patrí medzi základné problémy a ich pochopenie je pre celkové štúdium NN veľmi dôležité. Sme názoru, že zvladnutie tejto časti je nutné pre pochopenie ďalších problémov teórie NN.

V druhom diele knihy je venovaná pozornosť rekurentným NN a kontrolovanému ako aj nekontrolovanému učeniu na týchto sieťach. V závere knihy je naznačená problematika modulárnych NN.

6.2 Informácie o NN na sieti Internet

Okrem literárnych zdrojov je možné získať mnohé informácie na sieti **Internet**. Sú to napr.:

- FAQ - Frequently Asked Questions o NN. najdete v slovenskom jazyku na anonymnom servri `neuron-ai.tuke.sk` pod adresárom `/pub/FAQ/Slovak` alebo v anglickom jazyku pod adresárom

/pub/FAQ/English

- na USENETe existuje klub **comp.ai.neural-nets** v anglickom jazyku alebo taktiež na USENET serveri `ccnews.ke.sanet.sk` existuje klub **local.vyuka.neural**.
- existuje Internetovský diskusný klub

`cs - neuro - fuzzy@ccsun.tuke.sk`

na `listproc@ccsun.tuke.sk`, kde prebieha diskusia v slovenskom a českom jazyku. Prihlásiť sa do klubu môžete ako do riadneho diskusného klubu siete Internet.

- existuje obrovské množstvo WEB stránok napr:
 1. <http://www.neuronet.ph.kcl.ac.uk/>
 2. http://www-xdiv.lanl.gov/XCM/neural/neural_announcements.html
 3. <http://www1.cern.ch/NeuralNets/nnwInHepHard.html>
 4. <http://ftp.funet.fi/pub/sci/neural/>
 5. <http://www.emsl.pnl.gov:2080/docs/cie/neural/>
 6. <http://www.cs.utexas.edu/users/nn/>
 7. <http://engine.ieee.org/nnc/>
 8. <http://www.informatik.uni-stuttgart.de/ipvr/bv/projekte/sns/sns.html>

6.3 Knihy a časopisy o NN

Z časopisov hodno spomenúť niekoľko a to :

1. Neural Networks
2. IEEE on Neural Networks
3. Application of Neural Networks
4. Neural Networks World

5. Neural Computation

6. a ďalšie.

6.4 Organizácia záujemcov o NN na Slovensku

Na Slovensku existuje v rámci **Slovenskej Informatickej Spoločnosti** pracovná skupina záujemcov o neurónové siete (Slovak Neural Network Working Group (SNNWG)). Zaujemci sa môžu o aktivitách tohto združenia informovať na adrese **snnwg@ccsun.tuke.sk**. Kontaktní aktivisti tejto skupiny sú uvedení v tabuľke.

meno	E-mail adresa
Prof. Vladimír Kvasnička, DrSc.	kvasnic@cvt.stuba.sk
Dr. Ľubica Beňušková, CSc.	benus@dcs.elf.stuba.sk
Dr. Igor Farkaš, CSc.	umerfark@savba.savba.sk
Doc. Ing. Igor Mokriš, CSc.	mokris@valm.sk
Dr. Gabriela Andrejková, CSc.	andrejk@kosice.upjs.sk
Ing. Peter Sinčák, CSc.	sincak@ccsun.tuke.sk

Samozrejme existuje **európska** ako aj **svetová** organizácia NN, ktorá poriada veľké množstvo rôznych seminárov, sympózií a konferencií. Existuje aj **študentská organizácia NN vo svete**. Táto je súčasťou svetovej organizácie a napomáha finančnou pomocou nadaným študentom dostať sa na rôzne zaujímavé akcie o NN.

Literatúra

- [1] Barto A.G., Anandan P.: Pattern-recognizing stochastic learning automata, *IEEE Transaction on Systems, Man and Cybernetics* 15 (1985), 360-375.
- [2] Carpenter G.A., Grossberg A.: A Massively Parallel Architecture for Self-Organizing Neural Pattern Recognition Machine, *Computer Vision, Graphics and Image Processing*, 1987, vol.37, pp.54-115.
- [3] Funahashi K.: On the Aproximate Realization of Continuous Mapping by Neural Networks, *neural Networks*, Vol.2, pp. 183-192, 1989.
- [4] Halgamuge S.K. a kol. : Fast Perceptron Learning by Fuzzy Controlled Dynamic Adaptation of Network Parameters, in *Fuzzy Systems in Computer Science*, Editors R.Kruse at all, University of Amsterdam, 1994.
- [5] Haykin S.: *Neural Network - A Comprehensive Foundation* Macmillian Publishing, 1994.
- [6] Hertz J., Krogh A., Palmer R.: *Introduction to the Theory of Neural Computation* Addison Wesley publishing , 1991.
- [7] Jacobs R.A.: Increased rate of convergence through learning rate adaptation, *Neural Networks* 1, 1988, pp. 295-307.
- [8] Jacobs, R.A, Jordan, M.I.: Hierarchical mixtures of experts and the EM algorithm, *Neural Computation* 6, pp. 182-214, 1994.
- [9] Kohonen T.: *Self-Organization and Associative Memory* , Springer-Verlag Berlin, 1984.

- [10] Lippmann R.P.: Review of Neural Networks for Speech Recognition, Neural Computation , Volume 1-N1, 1989, pp. 1-38.
- [11] Lukasová A., Šarmanová J.: Metody shlukové analýzy, SNTL, Praha, 1985.
- [12] Minai A.A. Williams R.J.: Backpropagation heuristics: A study of the extended delta-bar-delta algorithm., International Joint Conference on NN, Vol 1, pp. 595-600, 1990 San Diego, CA - USA.
- [13] Novák V.: Fuzzy množiny a jejich aplikace, SNTL, Praha, 1990.
- [14] Pao Y.H.: Adaptive Pattern Recognition and Neural Network, Addison-Wesley, 1989.
- [15] Pearlmutter B.: Dynamic Recurrent Neural Networks , CMU-CS-90-196, 1990.
- [16] Pineda, F.J.: Recurrent Backpropagation and Dynamical Approach to Adaptive Neural Computation, Neural Computation, N.2, 1989.
- [17] Rosenblatt F.: Principles of Neurodynamics Perceptron and the Theory of Brain Mechanism, Spartan Press, Washington, 1962.
- [18] Sinčák, P. , Andras, D. , Reiss R.: Comparative Study of Fuzzy and Non-fuzzy Approach to Speeding-up Backpropagation Learning Procedure, Conference on Fuzzy Logic Budapest, 1996.
- [19] Sobajic, Pao Y.H.: Artificial Neural Net Based Security Assessment for Electrical Power Systems, IEEE Transactions on Power Systems, 1989, Vol3.
- [20] Waterhouse, S.R., Robinson, A.J.: Classification using hierarchical mixtures of experts, IEEE Workshop on Neural Networks for Signal Processing IV, pp. 177-186, 1994.
- [21] Ghahramani, Z., Jordan, M.I.: Learning from incomplete data, Massachusetts Institute of Technology, 1994.

Register

dilema stability & plasticity, 19

FAQ o NN, 57

indexová funkcia, 14

informácie

Internet, 57

knihy & časopisy, 58

organizácie o NN, 59

kontrolované učenie na RC NN,

1

neurón

duálny neurón, 1

RS model neurónu, 11

Neurónová sieť

Hopfieldove NNC, 3

modulárna, 31

bránová, 32

expertná, 32

rozdežuj a panuj, 32

rekurentná, 1

ART, 19

partialy RC NN, 2

učenie

kontrolované, 3

Backpropagation na RC NN,

10

TSP, 5

nekontrolované

na RC NN, 1, 19

reinforcement learning, 49

odmena a pokuta, 49