

SAMOORGANIZUJÚCE SA NEURÓNOVÉ SIETE

Metóda hlavných komponentov

Igor Farkaš

1 Samoorganizácia

Pod samoorganizáciou rozumieme proces učenia sa neurónovej siete v závislosti od prostredia, bez pomoci učiteľa. Možno povedať, že modelovanie samoorganizujúcich sa neurónových sietí (SONS) podstatne viac pripomína princípy neurobiologických štruktúr, než je to v prípade modelov umelých NS učiacich sa s učiteľom. Je to preto, lebo proces samoorganizácie je fundamentálnym javom v organizácii mozgu. Pravdepodobne sa uplatňuje na rôznych úrovniach spracovania informácie, počnúc perцепčnými subsystémami a končiac kortikálnymi oblasťami, ktoré spracovávajú komplexné informácie.

Samoorganizácia umožňuje **adaptívnosť** neurónovej siete. Vďaka nej nemusí byť sieť úplne geneticky predurčená, ale jej funkcia sa môže prispôbiť, napr. v prípade perцепčných subsystémov, podľa aktuálneho prostredia.

Samoorganizujúca sa neurónová sieť môže mať rôzne **architektúry**:

- (a) vstupnú a výstupnú vrstvu neurónov, s doprednými prepojeniami (váhami) medzi vrstvami,
- (b) ako v prípade a) plus laterálne prepojenia medzi neurónmi vo výstupnej vrstve,
- (c) viacvrstvovú architektúru s doprednými prepojeniami medzi susednými vrstvami neurónov.

Vo všetkých prípadoch učenie spočíva v iteratívnej modifikácii všetkých parametrov siete (váh) v závislosti od predkladaných vstupov, až kým sieť nenadobudne nejakú užitočnú finálnu konfiguráciu. Odpoveď, prečo taká finálna konfigurácia môže vzniknúť, možno nájsť v pozorovaní, ktoré sa vzťahuje na mozog ako aj na umelé NS (Turing, 1952):

Globálny poriadok môže vzniknúť z lokálnych interakcií.

Inými slovami, súčinnosť mnohých lokálnych, spočiatku náhodných interakcií medzi susednými neurónmi môže viesť ku **koherentnému správaniu** celej siete, čo je podstatnou črtou samoorganizácie.

Organizácia v NS sa odohráva na dvoch úrovniach, ktoré interagujú pomocou spätnej väzby:

- (a) úroveň aktivít neurónov – generovanie odpovedí siete na základe vstupných podnetov,
- (b) úroveň konektivity – modifikácia váh siete v závislosti od aktivít neurónov.

Aby sa dosiahla samoorganizácia, spätná väzba medzi zmenami váh a zmenami aktivít neurónov musí byť **kladná**.

1.1 Základné princípy samoorganizácie

V SONS možno pozorovať nasledujúce fenomény (von der Malsburg, 1990):

1. modifikácie váh majú tendenciu samozosilnenia,
2. obmedzenie zdrojov NS vedie k “súpereniu” medzi váhami, čo vyvoláva posilňovanie niektorých váh a utlmovanie ostatných (vďaka **synaptickej plasticite**), čím sa zabezpečuje stabilita systému,
3. modifikácie susedných váh majú tendenciu spolupracovať.

Uvedené 3 princípy sa vzťahujú na samotnú NS. Aby však SONS bola schopná naučiť sa nejaké užitočné vstupno-výstupné zobrazenie, je nutné, aby vstupné dáta predkladané sieti boli **redundantné** (Barlow, 1989). V terminológii teórie informácie to možno vyjadriť tak, že informačný obsah

vstupných kanálov musí byť menší než ich prenosová kapacita, rozdiel týchto dvoch veličín predstavuje redundanciu.

1.2 Využitie samoorganizujúcich sa neurónových sietí

Z pohľadu učenia majú SONS oproti modelom učiacim sa s učiteľom značnú výhodu v tom, že sa lepšie umožňujú zvládnuť **problém škálovania** (nárast času tréovania pri zvyšovaní počtu parametrov siete). Je to vďaka tomu, že učiace pravidlá pre SONS majú jednoduchý tvar a zväčša **lokálny** charakter. V prípade viacvrstvových modelov SONS je možné učiť jednotlivé vrstvy nie naraz (ako napr. v back-propagation), ale postupne jednu za druhou.

SONS možno použiť v podstate na riešenie týchto typov úloh:

- PCA – nájdenie tendencií (hlavných smerov) v dátach, t.j. optimálne lineárne kódovanie vzhľadom na strednú kvadratickú chybu rekonštrukcie,
- klastrovanie dát – zatriedenie do kategórií (vektorová kvantizácia),
- topologické zobrazenie príznakov vstupných dát,
- kódovanie dát s maximálnym prenosom informácie na výstup (minimálna redundancia výstupnej reprezentácie).

Použitie SONS sa obyčajne sprevádzané **redukciou dimenzie** vstupných dát, ktorá súvisí s **extrakciou príznakov**.

2 Metóda hlavných komponentov

Metóda hlavných komponentov (Principal Component Analysis - PCA) je zrejme najstaršou (Pearson, 1901) a najznámejšou metódou používanou v multivariačnej analýze. Umožňuje efektívne transformovať dáta zo (viacrozmerného) vstupného priestoru do tzv. priestoru *príznakov* s nižšou dimenziou, čím sa dáta premietnu do najdôležitejších lineárnych smerov a tie najmenej podstatné sa zanedbajú. PCA je teda sprevádzaná dvoma atribútmi dôležitými pri analýze dát – **extrakciou príznakov** a **redukciou dimenzie** dát.

Majme n -rozmerný náhodný vektor \mathbf{x} , ktorý popisuje vstupné dátum. Predpokladáme, že množina vstupných dát nulovú strednú hodnotu, t.j.

$$\langle \mathbf{x} \rangle = 0 .$$

Nech \mathbf{u} označuje jednotkový vektor, $\|\mathbf{u}\| = (\mathbf{u}^T \mathbf{u})^{1/2} = 1$, tiež dimenzie n , na ktorý premietneme vektor \mathbf{x} . Táto projekcia je definovaná ako

$$a = \mathbf{x}^T \mathbf{u} = \mathbf{u}^T \mathbf{x} .$$

Projekcia a (súradnica v 1D priestore generovaného vektorom \mathbf{u}) je náhodná premenná so strednou hodnotou a varianciou, ktoré závisia od štatistiky vektora \mathbf{x} . Keďže $\langle \mathbf{x} \rangle = 0$, potom

$$\langle a \rangle = \mathbf{u}^T \langle \mathbf{x} \rangle = 0 .$$

Variancia a je potom

$$\sigma^2 = \langle a^2 \rangle = \langle (\mathbf{u}^T \mathbf{x})(\mathbf{x}^T \mathbf{u}) \rangle = \mathbf{u}^T \langle \mathbf{x} \mathbf{x}^T \rangle \mathbf{u} = \mathbf{u}^T \mathbf{R} \mathbf{u} , \quad (1)$$

kde $\mathbf{R}[n \times n]$ je **korelačná matica** vstupných dát. Je vidieť, že \mathbf{R} je symetrická, t.j. $\mathbf{R}^T = \mathbf{R}$. Z tejto vlastnosti vyplýva, že ak máme nejaké n -rozmerné vektory \mathbf{a} a \mathbf{b} , potom

$$\mathbf{a}^T \mathbf{R} \mathbf{b} = \mathbf{b}^T \mathbf{R} \mathbf{a} . \quad (2)$$

Z rovnice (1) je vidieť, že variancia projekcie σ^2 je funkciou jednotkového vektora \mathbf{u} , čo matematicky vyjadríme vzt'ahom

$$\sigma^2 = \psi(\mathbf{u}) = \mathbf{u}^T \mathbf{R} \mathbf{u} . \quad (3)$$

2.1 Štruktúra vlastných hodnôt PCA

Budeme hľadať taký smer \mathbf{u} , v ktorom funkcia $\psi(\mathbf{u})$ nadobúda extrém. Ako uvidíme, riešenie tohto optimalizačného problému závisí od štruktúry vlastných hodnôt korelačnej matice \mathbf{R} . Ak \mathbf{u} je taký 1-ový vektor, $\psi(\mathbf{u})$ nadobúda extrém, potom pre ľubovoľnú malú perturbáciu $\delta\mathbf{u}$ podľa Taylorovho rozvoja zanedbaním členov vyšších rádov dostaneme

$$\psi(\mathbf{u} + \delta\mathbf{u}) = \psi(\mathbf{u}) . \quad (4)$$

Potom podľa (3)

$$\psi(\mathbf{u} + \delta\mathbf{u}) = (\mathbf{u} + \delta\mathbf{u})^T \mathbf{R} (\mathbf{u} + \delta\mathbf{u}) = \mathbf{u}^T \mathbf{R} \mathbf{u} + 2(\delta\mathbf{u})^T \mathbf{R} \mathbf{u} + (\delta\mathbf{u})^T \mathbf{R} \delta\mathbf{u} .$$

Ak zanedbáme člen druhého rádu, využitím rovnosti (4) dostaneme podmienku

$$(\delta\mathbf{u})^T \mathbf{R} \mathbf{u} = 0 . \quad (5)$$

Súčasne musí platiť, že $\|\mathbf{u} + \delta\mathbf{u}\| = 1$, čo môžeme ekvivalentne zapísať ako $(\mathbf{u} + \delta\mathbf{u})^T (\mathbf{u} + \delta\mathbf{u}) = 1$, odkiaľ po rozpísaní, a po zanedbaní člena druhého rádu máme

$$(\delta\mathbf{u})^T \mathbf{u} = 0 . \quad (6)$$

Z toho vyplýva, že perturbácie $\delta\mathbf{u}$ musia byť kolmé na \mathbf{u} .

Keďže elementy 1-ého vektora sú fyzikálne bezrozmerné, tak ak chceme skombinovať rovnice (5) a (6), musíme k druhému členu pridať škálovaciu konštantu λ s takými rozmermi, aké majú prvky matice \mathbf{R} . Potom môžeme písať

$$(\delta\mathbf{u})^T \mathbf{R} \mathbf{u} - \lambda (\delta\mathbf{u})^T \mathbf{u} = (\delta\mathbf{u})^T (\mathbf{R} \mathbf{u} - \lambda \mathbf{u}) = 0 .$$

Odtiaľ vyplýva, že musí platiť

$$\mathbf{R} \mathbf{u} = \lambda \mathbf{u} ,$$

čo je rovnica známa ako *problém vlastných hodnôt*, s ktorým sa často stretávame v lineárnej algebre. Pre $\mathbf{u} \neq \mathbf{0}$ má tento problém netriviálne riešenia, λ sa nazývajú **vlastné čísla**, s nimi súvisiace vektory \mathbf{u} sú **vlastné vektory** korelačnej matice \mathbf{R} . Jej vlastné čísla majú nezáporné reálne hodnoty.

Označme vlastné čísla kovariančnej matice $\mathbf{R}[n \times n]$ ako $\lambda_1, \lambda_2, \dots, \lambda_n$ a vlastné vektory ako $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$. Potom možno písať

$$\mathbf{R} \mathbf{u}_j = \lambda \mathbf{u}_j , \quad j = 1, 2, \dots, n . \quad (7)$$

Nech sú vlastné čísla usporiadané zostupne, t.j. $\lambda_1 > \lambda_2 > \dots > \lambda_n$, čiže $\lambda_1 = \lambda_{max}$. Skonstruujme maticu \mathbf{U} ako

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n] .$$

Potom možno (7) prepísať do maticového tvaru

$$\mathbf{R} \mathbf{U} = \mathbf{U} \mathbf{\Lambda} , \quad (8)$$

kde $\mathbf{\Lambda}$ je diagonálna matica $\mathbf{\Lambda} = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_n]$. \mathbf{U} je *ortogonálna matica*, lebo jej stĺpce sú navzájom ortonormálne:

$$\mathbf{u}_i^T \mathbf{u}_j = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases}$$

Ekvivalentným zápisom je $\mathbf{U}^T \mathbf{U} = \mathbf{I}$, odkiaľ je zrejmé, že $\mathbf{U}^T = \mathbf{U}^{-1}$. Využitím tejto vlastnosti možno (8) prepísať do tvaru

$$\mathbf{U}^T \mathbf{R} \mathbf{U} = \mathbf{\Lambda}$$

resp. v rozpísanom tvare

$$\mathbf{u}_i^T \mathbf{R} \mathbf{u}_j = \begin{cases} \lambda_j, & j = i \\ 0, & j \neq i \end{cases}, \quad (9)$$

odkiaľ je vidieť, že

$$\psi(\mathbf{u}_j) = \lambda_j, \quad j = 1, 2, \dots, n$$

Vyššie odvodené vzťahy možno teda zosumarizovať v dvoch bodoch:

- Vlastné vektory korelačnej matice centrovaných dát sú reprezentované 1-vými vektormi \mathbf{u}_j , predstavujúcimi hlavné smery, v ktorých variancia $\psi(\mathbf{u}_j)$ nadobúda extrém.
- Asociované vlastné čísla definujú extrémny variancií $\psi(\mathbf{u}_j)$.

2.2 Základné reprezentácie dát

Pre n rôznych vlastných vektorov \mathbf{u}_j dostávame n rôznych projekcií vstupného vektora

$$a_j = \mathbf{u}_j^T \mathbf{x} = \mathbf{x}^T \mathbf{u}_j, \quad j = 1, 2, \dots, n$$

ktoré budeme nazývať **hlavnými komponentami**; majú tie isté fyzikálne rozmery ako vstupný vektor \mathbf{x} . Tento krok označujeme ako *analýza*.

Pri rekonštrukcii pôvodného vektora \mathbf{x} z projekcií a_j postupujeme tak, že skombinujeme projekcie a_j do vektora $\mathbf{a} = [a_1, a_2, \dots, a_n]^T$ a potom možno písať

$$\mathbf{a} = [\mathbf{u}_1^T \mathbf{x}, \mathbf{u}_2^T \mathbf{x}, \dots, \mathbf{u}_n^T \mathbf{x}]^T = \mathbf{U}^T \mathbf{x}.$$

Vynásobením oboch strán rovnice maticou \mathbf{U} zľava dostaneme vzťah pre rekonštrukciu

$$\mathbf{x} = \mathbf{U} \mathbf{a} = \sum_{j=1}^n a_j \mathbf{u}_j, \quad (10)$$

ktorý možno chápať ako *syntéza*. V tomto zmysle vektory predstavujú *bázu* priestoru dát. Ide teda transformáciu súradníc, podľa ktorej sa bod \mathbf{x} vo vstupnom priestore transformuje na bod \mathbf{a} v priestore *príznakov*.

2.3 Redukcia dimenzie

Redukcia dimenzie dát má z pohľadu štatistického rozpoznávania veľký význam. Umožní nám redukovat' počet príznakov (súradníc) potrebných na efektívny popis dát. Takto môžeme napr. ignorovať smery, v ktorých majú dáta len malú varianciu (rozlišiteľnosť) a všímať si len tie s veľkou varianciou (Oja, 1983).

Nech $\lambda_1, \lambda_2, \dots, \lambda_p$ označujú p najväčších vlastných čísel korelačnej matice \mathbf{R} . Rekonštrukciu vektora \mathbf{x} potom možno aproximovať ako

$$\mathbf{x}' = \mathbf{U} \mathbf{a} = \sum_{j=1}^p a_j \mathbf{u}_j, \quad p < n \quad (11)$$

Chybu aproximácie možno potom vyjadriť ako

$$\mathbf{e} = \mathbf{x} - \mathbf{x}' = \sum_{j=p+1}^n a_j \mathbf{u}_j .$$

Jednoducho sa dá ukázať, že chybový vektor \mathbf{e} je kolmý na aproximáciu \mathbf{x}' :

$$\mathbf{e}^T \mathbf{x}' = \sum_{i=p+1}^n a_i \mathbf{u}_i \sum_{j=1}^p a_j \mathbf{u}_j = \sum_{i=p+1}^n \sum_{j=1}^p a_i a_j \mathbf{u}_i^T \mathbf{u}_j = 0 .$$

Celkovú varianciu p komponentov náhodného vektora \mathbf{x} možno podľa (3) a podľa prvého riadku (9) vyjadriť ako

$$\sum_{j=1}^n \sigma_j^2 = \sum_{j=1}^n \lambda_j ,$$

kde σ_j^2 je variancia j -tej hlavného komponentu a_j . Analogicky, celková variancia m elementov aproximácie \mathbf{x}' je rovná

$$\sum_{j=1}^m \sigma_j^2 = \sum_{j=1}^m \lambda_j .$$

$\lambda_{p+1}, \dots, \lambda_p$ je $n - p - 1$ najmenších vlastných čísel matice \mathbf{R} . Odpovedajú členom eliminovaným v aproximácii (11). Čím sú tieto vlastné čísla menšie, tým efektívnejšia je redukcia dimenzie z pohľadu množstva zachovanej informácie o vstupných dátach.

Aby sme teda redukovali dimenziu dát pomocou PCA, potrebujeme *vypočítať vlastné čísla a vlastné vektory ich korelačnej matice, a potom urobiť ortogonálnu projekciu do podpriestoru generovaného vlastnými vektormi odpovedajúcimi najväčším vlastným číslam*. Táto reprezentácia dát patrí do triedy metód označovaných ako *dekompozícia podpriestoru* (Oja, 1983).

3 Hebbovo učenie - jeden lineárny neurón

Kanadský psychológ Donald Hebb postuloval pravidlo modifikácie váh vo svojej priekopníckej knihe *The Organization of Behavior* (1949) nasledovne:

Keď axón bunky A je dostatočne blízko, aby excitoval bunku B a opakovane sa podieľa na vyvolaní jej výstupnej aktivity, potom by aspoň v jednej z týchto dvoch buniek mal prebehnúť proces rastu alebo nejaké metabolické zmeny, aby sa zvýšila efektivita bunky A pri vyvolávaní aktivity bunky B.

V analytickom vyjadrení to znamená, že spojenie (synapsa) w_{ba} vedúce z bunky a do bunky b sa má modifikovať podľa vzťahu $dw_{ba}/dt = \alpha y_a y_b$, kde y_a je presynaptická aktivita, y_b postsynaptická aktivita a α udáva veľkosť kroku adaptácie.

Uvažujme najprv, že máme jeden lineárny neurón s n vstupmi, ktorého výstup sa počíta ako

$$y = \sum_{j=1}^n w_j x_j = \mathbf{w}^T \mathbf{x} .$$

Ak by sme priamo uplatnili Hebbov postulát, pravidlo učenia by malo tvar

$$w_j(t+1) = w_j(t) + \alpha y(t) x_j(t) \quad j = 1, 2, \dots, n$$

kde parameter α je rýchlosť učenia. Je však zrejmé, pri takejto adaptácii by váhy divergovali do nekonečna, čo je fyzikálne neprípustné. Priamym riešením tohto problému je explicitná normalizácia váh po každom kroku (Oja, 1982)

$$w_j(t+1) = \frac{w_j(t) + \alpha y(t)x_j(t)}{(\sum_{j=1}^n [w_j(t) + \alpha y(t)x_j(t)]^2)^{1/2}},$$

ktorá vnáša do algoritmu súperenie váh o limitované zdroje (2. princíp samoorganizácie). Za predpokladu, že α je malé, možno vzťah prepísať pomocou rozvoja ako

$$w_j(t+1) = w_j(t) + \alpha y(t)[x_j(t) - y(t)w_j(t)] + O(\alpha^2),$$

kde $O(\alpha^2)$ v sebe zahŕňa členy druhého a vyšších rádov. Pre malé α možno tento člen zanedbať, čím dostávame pravidlo učenia (Oja, 1982)

$$w_j(t+1) = w_i(t) + \alpha y(t)[x_j(t) - y(t)w_j(t)] = w_j(t) + \alpha y(t)x_j(t) - \alpha y^2(t)w_j(t), \quad (12)$$

kde prostredný člen predstavuje Hebbov člen (1. princíp samoorganizácie) a posledný je stabilizačný člen. Je vidieť, že tvar učiaceho pravidla (12) je z pohľadu vstup-výstup opačný k delta pravidlu. Ak urobíme substitúciu

$$x'_j(t) = x_j(t) - y(t)w_j(t),$$

kde x'_j možno interpretovať ako *efektívny vstup* j -tej synapsy, potom

$$w_j(t+1) = w_j(t) + \alpha y(t)x'_j(t),$$

čo je vzťah podobný pôvodnému tvaru modifikácie, avšak tu už vo vstupe x'_j je skrytá kombinácia kladnej a zápornej spätnej väzby.

3.1 Správanie modelu

O lineárnom neuróne s učiacim pravidlom podľa 12 možno vysloviť nasledovné tvrdenia (ak $\langle \mathbf{x} \rangle = 0$):

1. váhový vektor \mathbf{w} konverguje do smeru, ktorý je vlastným vektorom korelačnej matice vstupov $\mathbf{C} = \langle \mathbf{xx}^T \rangle$.
Ak uvažujeme, že po ustálení je stredná hodnota zmien váhových zložiek nulová, môžeme písať $0 = \langle \Delta w_j \rangle = \langle y(x_j - yw_j) \rangle = \langle \sum_k w_k x_k (x_j - \sum_k w_k x_k w_j) \rangle = \langle \sum_k w_k x_k x_j - \sum_{kl} w_k x_k w_l x_l w_j \rangle = \sum_k c_{jk} w_k - (\sum_{kl} w_k c_{kl} w_l) w_j$, kde $c_{jk} = \langle x_j x_k \rangle$ a $c_{kl} = \langle x_k x_l \rangle$ sú kovariančné koeficienty matice \mathbf{C} . Vo vektorovom zápise $\mathbf{0} = \langle \Delta \mathbf{w} \rangle = \mathbf{Cw} - (\mathbf{w}^T \mathbf{Cw})\mathbf{w} \implies \mathbf{Cw} = \lambda \mathbf{w}$.
2. váhový vektor \mathbf{w} po konvergencii maximalizuje $\langle y^2 \rangle$.
Táto vlastnosť je dôsledkom vlastnosti 1: $\langle y^2 \rangle = \langle (\mathbf{w}^T \mathbf{x})^2 \rangle = \langle \mathbf{w}^T \mathbf{xx}^T \mathbf{w} \rangle = \mathbf{w}^T \mathbf{Cw}$, a kvadratická norma $\mathbf{w}^T \mathbf{Cw}$ nadobúda maximálnu hodnotu $\iff \mathbf{w}$ leží v smere jej hlavného vlastného vektora.
3. po konvergencii norma $\|\mathbf{w}\| = 1$.
Z odvodenia pri vlastnosti 1 vyplýva, že $\lambda = \mathbf{w}^T \mathbf{Cw} = \mathbf{w}^T \lambda \mathbf{w} = \lambda \|\mathbf{w}\|^2 \implies \|\mathbf{w}\| = 1$.

Možno teda povedať, že lineárny neurón s pravidlom učenia 12 má tendenciu maximalizovať strednú hodnotu kvadrátu výstupu. Pri dátach s nulovou strednou hodnotou tým súčasne dochádza k nájdeniu hlavného komponentu kovariančnej matice vstupných dát.

4 Algoritmus GHA

Uvažujme, že máme NS s n vstupmi a p výstupmi, pričom $p < n$. Nech váha w_{ij} označuje spojenie i -teho neurónu s j -tym vstupom. Pravidlo učenia GHA (Generalized Hebbian Algorithm) navrhnuté Sangerom (1989) má tvar

$$\Delta w_{ij} = \alpha y_i (x_j - \sum_{k=1}^i y_k w_{kj}) \quad \text{pre } i = 1, 2, \dots, p. \quad (13)$$

O tomto pravidle učenia bolo dokázané, že extrahuje p hlavných komponentov korelačnej matice vstupov. Môže to byť zrejmejšie, keď si pravidlo prepíšeme do tvaru

$$\Delta w_{ij} = \alpha y_i (x'_j - y_i w_{ij}), \quad \text{kde } x'_j = x_j - \sum_{k=1}^{i-1} y_k w_{kj}.$$

Vo vektorovom zápise

$$\Delta \mathbf{w}_i = \alpha y_i (\mathbf{x}' - y_i \mathbf{w}_i) \quad \text{pričom } \mathbf{x}' = \mathbf{x} - \sum_{k=1}^{i-1} y_k \mathbf{w}_k.$$

Pri tomto tvare je vidieť podobnosť s pravidlom pre jeden Ojov neurón, pričom \mathbf{x}' tu vystupuje ako modifikovaný vstupný vektor. Z pohľadu výstupných neurónov vyzerá teraz situácia nasledovne: Pre neurón 1 je $\mathbf{x}' = \mathbf{x}$, t.j. má tendenciu premietat' vstupy do smeru hlavného komponentu. Pre neurón 2 je $\mathbf{x}' = \mathbf{x} - y_1 \mathbf{w}_1$, čo znamená, že tento neurón "vidí" vstup, od ktorého bola odpočítaná zložka odpovedajúca hlavnému vlastnému vektoru, preto premieta do smeru druhej hlavnej zložky, atď.

Výhody oproti štandardnej metóde PCA:

- (1) nie je explicitne nutné vopred počítat' korelačnú maticu,
- (2) je výpočtovo menej náročný najmä v prípadoch, keď $p \ll n$.

Vlastnosti algoritmu:

- (a) nájde hlavné komponenty,
- (b) tie sú usporiadané podľa veľkosti (variancie výstupov v jednotlivých ortogonálnych smeroch),
- (c) výsledky sú reprodukovateľné (až na znamienko),
- (d) vhodná aplikácia: kódovanie dát, kompresia.

4.1 Ojova verzia algoritmu

V aplikáciách, kde nie je nutné extrahovať samotné hlavné komponenty, ale stačí nájsť **hlavný podpriestor** (podpriestor generovaný hlavnými komponentami), možno použiť Ojovu verziu učiaceho pravidla v tvare

$$\Delta w_{ij} = \alpha y_i (x_j - \sum_{k=1}^p y_k w_{kj}) \quad \text{pre } i = 1, 2, \dots, p. \quad (14)$$

Vlastnosti algoritmu:

- (a) algoritmus nenájde hlavné komponenty, len podpriestor nimi generovaný,
- (b) výstupy nie sú usporiadané, variancia jednotlivých zložiek na výstupe je približne rovnaká,
- (c) výsledky nie sú reprodukovateľné, závisia od počiatočných podmienok a od poradia predkladania vstupov,
- (d) vhodná aplikácia v prípadoch, ak sa požaduje rovnomerne distribuovaná reprezentácia.

5 Algoritmus APEX

Kung & Diamantaras (1990) navrhli iný model na výpočet hlavných komponentov – algoritmus APEX (Adaptive Principal Component Extraction). Okrem dopredných spojení tu existujú i laterálne spojenia medzi výstupnými neurónmi, a to ku každému neurónu od ostatných neurónov s nižším indexovým číslom, t.j. i -ty neurón má laterálny váhový vektor

$$\mathbf{u}_i = [u_{i1}, u_{i2}, \dots, u_{i,i-1}]^T$$

a jeho výstup sa počíta podľa vzťahu

$$y_i = \mathbf{w}_i^T \mathbf{x} + \mathbf{u}_i^T \mathbf{y}_{i-1}$$

kde výstupný vektor $\mathbf{y}_{i-1} = [y_1, y_2, \dots, y_{i-1}]^T$. Dopredné váhy sa modifikujú podľa Sangerovho pravidla 13, a laterálne váhy podľa anti-Hebbovho pravidla (majú inhibičný účinok)

$$\Delta u_{ik} = -\alpha y_i (y_k + y_i u_{ik}) \quad (15)$$

O algoritme APEX bolo dokázané, že váhové vektory \mathbf{w}_i po poradí konvergujú do smerov vlastných vektorov korelačnej matice vstupov, zatiaľ čo laterálne váhové vektory \mathbf{u}_i konvergujú k nulovým vektorom. Na rozdiel od (takmer) simultánnej konvergenencie váh v algoritme GHA, tu konvergujú váhové vektory postupne.

Bolo navrhnutých viacero neuronálnych algoritmov na výpočet PCA. Možno ich rozdeliť do dvoch skupín (Becker & Plumbey, 1993):

- (a) **reestimačné algoritmy** (napr. GHA), v ktorých existujú len dopredné spojenia, a ktoré sa učia pomocou nejakej verzie Hebbovho pravidla, a
- (b) **dekorelačné algoritmy** (napr. APEX), v ktorých existujú aj laterálne spojenia, modifikované pomocou nejakej verzie anti-Hebbovho pravidla.

PCA možno realizovať i pomocou autoasociácie pri použití NS typu $n-p-n$ a učenia pomocou BP. V takom prípade k PCA reprezentácii dochádza na neurónoch v skrytej vrstve (Baldi & Hornik, 1989).

6 Použitie PCA algoritmov

PCA je vhodné použiť na predspracovanie dát, ktoré majú zhruba Gaussovské rozdelenie. Sprievodnými znakmi predspracovania pomocou PCA sú: (1) dekorelovanie vstupov, (2) zníženie dimenzie dát a (3) extrakcia lineárnych príznakov (hlavných komponentov). Napr. pri takto transformovaných dátach možno dosiahnuť podstatne rýchlejšiu konvergenciu BP vďaka dekorelovanosti súradníc vstupov.

Linsker (1988) ukázal, že PCA je ekvivalentná maximalizácii výstupnej informácie v prípade Gaussovského rozdelenia dát. I z toho vyplýva nevhodnosť predspracovania dát pomocou PCA, ak tie majú výrazne negaussovskú distribúciu, napr. pozostávajú zo zhlukov alebo v nich existujú bezvýznamné mimo ležiace prvky (outliers).

Literatúra

- Haykin S. (1994). *Neural Networks: A Comprehensive Foundation*. Macmillan College, New York.
- Hertz J., Krogh A. & Palmer R.G. (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, CA.