

7. Samoorganizujúce sa mapy

Igor Farkaš¹

7.1. Úvod

V tejto kapitole sa budeme venovať ďalšiemu zo základných modelov umelých neurónových sietí, známemu pod názvom *SamoOrganizujúca sa Mapa* (SOM), ktorej autorom je Teuvo Kohonen [29]. Ako vyplýva už z názvu, SOM patrí do kategórie modelov, ktoré sa učia *bez učiteľa* (samoorganizovane), t.j. algoritmus učenia nemá informáciu o požadovaných aktivitách výstupných neurónov v priebehu tréningu (ako napr. v algoritme back-propagation), o ktoré by sa mohol "opierať". Ako v iných algoritmoch samoorganizácie, i tu adaptácia váh odzrkadľuje štatistické vlastnosti tréningovej množiny, ktorá je sieť prezentovaná vo forme vstupných vzorov (vektorov).

Špecifickou črtou SOM je to, že (pri splnení istých podmienok) umožňuje realizovať *zobrazenie zachovávajúce topológiu a zobrazíť tak charakteristické príznaky (črty)* tréningovej množiny dát. Za týmto účelom sú neuróny zoradené v pravidelnej, zväčša dvojrozmernej alebo jednorozmernej štruktúre (mriežka alebo reťaz). Takto uvažované usporiadanie neurónov predstavuje výstupný priestor, v ktorom vzdialenosť dvoch neurónov je obyčajne daná euklidovskou vzdialenosťou vektorov ich súradníc v uvažovanej štruktúre. Zobrazenie zachovávajúce topológiu, ktoré vznikne po natrénovaní SOM, má dôležitú vlastnosť: ľubovoľné dva vzory blízke vo vstupnom priestore evokujú v sieťi odozvy na neurónoch, ktoré sú tiež fyzicky blízke (vo výstupnom priestore).

Fenomén topologického zobrazenia príznakov (angl. *feature mapping*) má výrazné zastúpenie v biologických neurónových sieťach, konkrétne v mozgoch vyšších cicavcov i človeka [22]. *Topografické mapy*, ktorých existencia bola zistená v jednotlivých častiach mozgu, hlavne mozgovej kôry, predstavujú efektívny spôsob reprezentácie dôležitých parametrov vstupných dát. Jedná sa o mapy, ktoré buď priamo reprodukovujú periférnu reprezentáciu, tzv. projekčné oblasti (napr. mapa povrchu tela), alebo reprezentované parametre sú nejakým spôsobom vypočítavané. Ako príklady možno uviesť vizuálne mapy (napr. mapa orientácie čiarových stimulov), sluchové mapy (napr. mapa frekvencií a amplitúd akustických stimulov), a tiež mapy v motorickej oblasti (napr. riadenie pohybu očí). Komplexnejším príkladom je mapa pozícií zdroja zvuku, ktorá sa "počíta" z jednoduchších sluchových máp. Z toho vyplýva, že mozgová kôra je do značnej miery priestorovo organizovaná a že je pre ňu charakteristická *lokálnosť odoziev* na vstupné podnety.

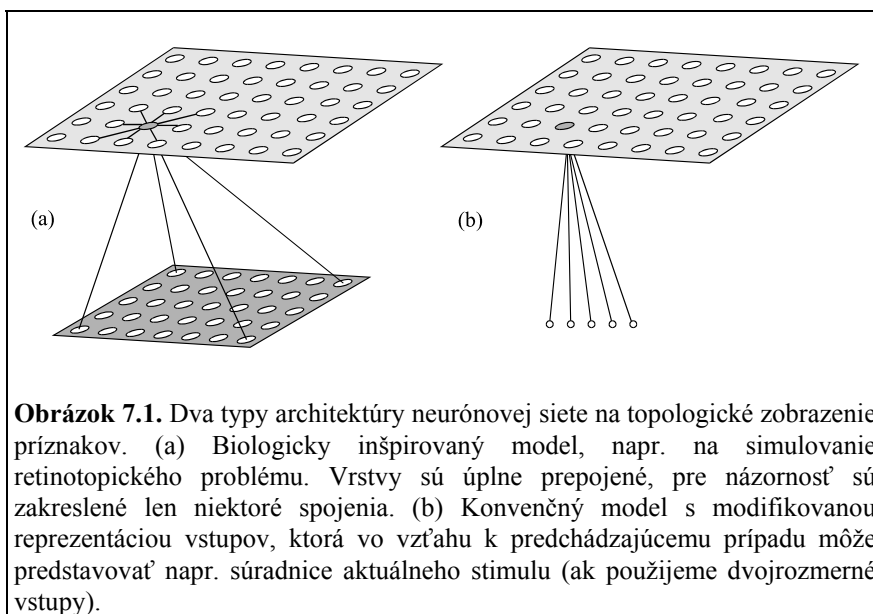
Ďalšou skutočnosťou potvrdenou experimentálne je fakt, že topografické mapy nie sú úplne vyvinuté už pri narodení, ale formujú sa v počiatočných štádiách vývoja v dôsledku zmyslovej skúsenosti. Inými slovami, hoci usporiadanie jednotlivých častí mozgu a ich funkcie sú dané geneticky, je tu priestor i pre *modifikovateľnosť* týchto štruktúr. Navyše je evidentné, že proces modifikácie prebieha na základe senzorickej podnetov prichádzajúcich z okolitého prostredia, a teda z pohľadu spôsobu učenia prebieha samoorganizovane.

Uvedené fakty boli inšpiračným zdrojom pre snahu simulovať proces samoorganizácie pomocou výpočtového modelu. Navyše, ako sa neskôr ukázalo, model SOM vďaka svojej relatívnej jednoduchosti a ilustratívnosti našiel uplatnenie pri riešení rôznych praktických problémov, kde sa využíva topologické zobrazenie, ako napr. pri rozpoznávaní vzorov (najmä hlások reči), v robotike (transformácia súradníc, generovanie modelu prostredia), kompresii obrazov, riadení procesov v priemysle, optimalizačných úlohách, či spracovaní prirodzeného jazyka. Zoznam aplikácií možno nájsť v [30], v posledných rokoch k nim pribudli ďalšie.

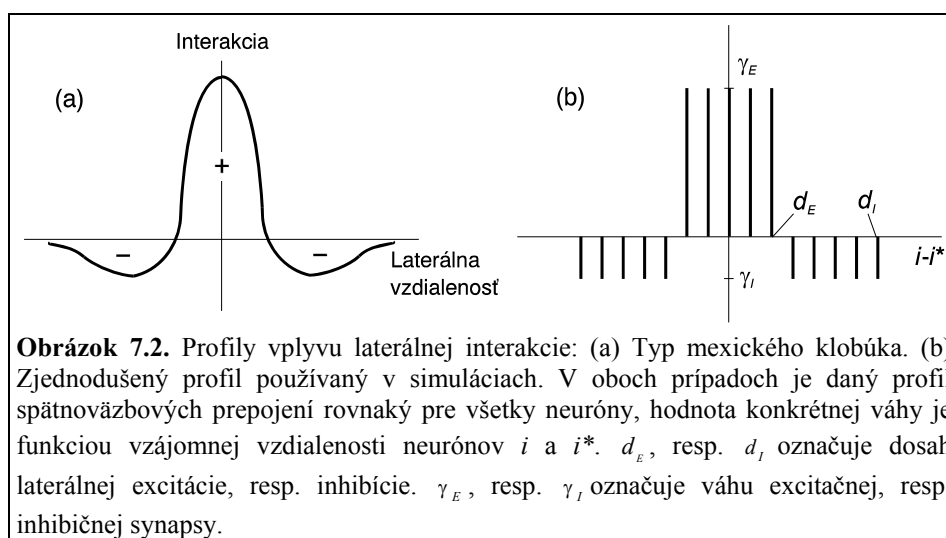
7.1.1 Prvé biologicky inšpirované modely

Jednými z prvých, ktorí sa zaoberali problémom topologického mapovania pomocou neurónových sietí, boli Willshaw a von der Malsburg [46]. V snahe porozumieť biologicky zaujímavému problému – mechanizmu projekcie zo sietnice na mozgovú kôru (retinotopický problém) – navrhli model neurónovej siete s architektúrou ako vidieť na obr. 7.1a.

¹ Kapitola z knihy: Kvasnička, V., Beňušková L., Pospíchal J., Farkaš I., Tiňo P., Král' A. Úvod do teórie neurónových sietí. Iris, Bratislava, 1997.



Receptívne bunky spodnej, vstupnej vrstvy (reprezentujúcej sietnicu oka, tzv. retinu) sú spojené s neurónmi mozgovej kôry vo výstupnej vrstve, a to spôsobom každý s každým. Okrem toho, neuróny vo výstupnej vrstve obsahujú nemenné *laterálne (bočné) prepojenia*, ktoré sa vzájomne privádzajú na vstup neurónov v rámci laterálneho dosahu. Sila vplyvu týchto prepojení, reprezentovaná synaptickými váhami, sa so vzdialenosťou od neurónu mení podľa profilu tvaru mexického klobúka (obr. 7.2).



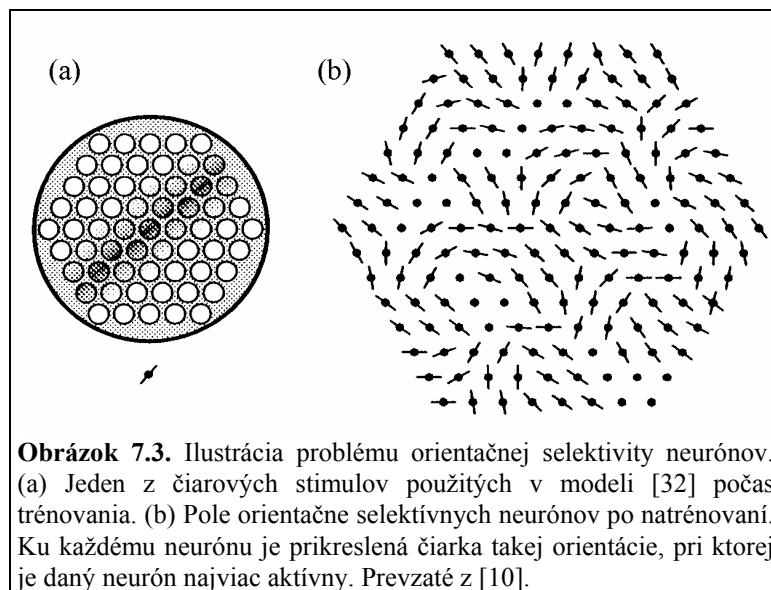
Ako vidieť na obr. 7.2, laterálna interakcia má *excitačný* účinok pre blízke neuróny, a v menšej miere *inhibičný* účinok pre navzájom vzdialenejšie neuróny. Vstup každého neurónu vo výstupnej vrstve je teda súčtom dvoch váhovaných zložiek: doprednej (od neurónov vstupnej vrstvy) a spätnoväzbovej (z výstupov okolitých neurónov v mape, váhovaných podľa spomínaného profilu). Vstupnými vzormi boli rôzne tzv. dipólové stimuly, t.j. dva susedné receptory na retine aktívne a ostatné neaktívne. Učenie bolo založené na štandardnom Hebbovom pravidle² s následným normovaním váh. Výsledkom tréningu na dipólových stimuloch bolo získanie projekcie typu pozícia versus pozícia, s dodržanou vlastnosťou topologického usporiadania odoziev vo výstupnej vrstve. Treba spomenúť, že očakávaný efekt topologického usporiadania by sa nebol dostavil, keby bol len jeden receptor naraz aktívny. Simultánna aktivita minimálne dvoch susedných receptorov predstavuje

²Hebbovo pravidlo je založené na princípe zvyšovania hodnoty synaptických váh medzi neurónmi, ktoré sú synchronne aktívne, t.j. $\Delta w_{ij} \propto y_i \cdot y_j$ (v spomínanom modeli namiesto vstupného neurónu figuruje receptor). Adaptujú sa všetky spojenia, pričom miera adaptácie teda závisí od korelácie medzi výstupmi oboch neurónov.

redundanciu a umožňuje sieti zachytiť korelačné, priestorové vzťahy na vstupe, čo je nutnou podmienkou pre dosiahnutie správnej samoorganizácie.

K zaujímavému výsledku sa následne dopracovali Takeuchi a Amari [43], ktorí analyzovali modifikovanú verziu spomínaného modelu v jednorozmernom prípade. Zistili, že k topologicky správnej organizácii na výstupe dochádza za predpokladu, že excitačná časť mexického klobúka je dostatočne široká v porovnaní s veľkosťou (šírkou) lokálnych vstupov. V opačnom prípade je konfigurácia nestabilná a prejavuje sa existenciou tzv. stĺpcových mikroštruktúr.

Projekcia typu pozícia versus pozícia nie je jediným typom, ktorý bol študovaný. Ešte skôr sa von der Malsburg [32] venoval príbuznému biologicky zameranému problému - simulácii formovania orientačnej selektivity neurónov vo vizuálnej kôre. Na rozdiel od predchádzajúceho modelu, tu išlo o projekciu typu uhol orientácie versus pozícia, pri ktorej boli stimulom čiary rôznych orientácií vo vstupnej vrstve (t.j. v poli receptorov), prechádzajúce jej stredom (obr. 7.3a). Obr. 7.3b ilustruje situáciu vo výstupnej vrstve po natrénovaní. Je vidieť, že preferovaná orientačná selektivita neurónov v mape sa mení spojite (s občasnými skokmi), čo je pozorované i v biologických sieťach.



Obrázok 7.3. Ilustrácia problému orientačnej selektivity neurónov. (a) Jeden z čiarových stimulov použitých v modeli [32] počas tréningu. (b) Pole orientačne selektívnych neurónov po natrénovaní. Ku každému neurónu je prikreslená čiarka takej orientácie, pri ktorej je daný neurón najviac aktívny. Prevzaté z [10].

7.1.2 Formovanie lokálnych odoziev vplyvom laterálnej spätnej väzby

Je užitočné ozrejmiť si, akú úlohu zohráva laterálna spätná väzba v spomínaných samoorganizujúcich sa modeloch pri formovaní výstupnej aktivity neurónov. Jej vplyv možno najlepšie ilustrovať na príklade. Pre jednoduchosť budeme uvažovať jednorozmerný prípad, t.j. receptorové bunky i neuróny vo výstupnej vrstve zoradené v reťazi. Nech vektor \mathbf{y} označuje distribúciu aktivít jednotlivých neurónov vo výstupnej vrstve (reťazi) a zložky vektora \mathbf{net} označujú analógie postsynaptických potenciálov. Dynamiku takejto siete možno popísať rovnicou

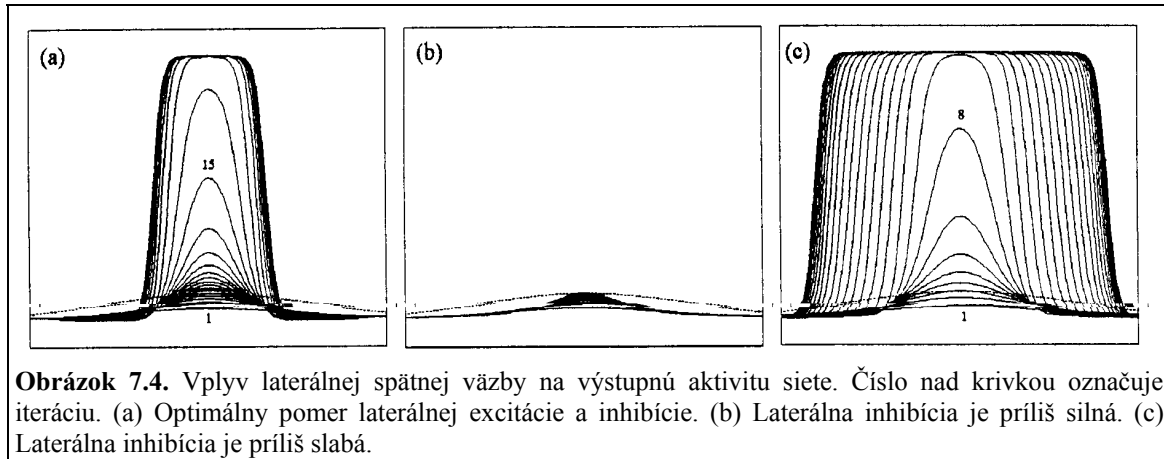
$$\mathbf{y}(t+1) = \mathbf{S}[\mathbf{net}] = \mathbf{S}[\mathbf{z} + \mathbf{L} \cdot \mathbf{y}(t)] \quad (7.1)$$

Na vstup sa teda privedie lokálny stimul \mathbf{x} (aktivita receptorového poľa), vypočíta sa počiatočná vnútorná aktivita výstupných neurónov $\mathbf{z} = \mathbf{W} \cdot \mathbf{x}$, kde \mathbf{W} je matica váh medzi receptormi a neurónmi, a laterálna spätná väzba sa nechá niekoľko iterácií pôsobiť. Je daná symetrickou maticou \mathbf{L} , koeficienty ktorej možno jednoducho získať z profilu na obr. 7.2b.³ Vektorová funkcia \mathbf{S} , pozostávajúca zo sigmoid, zabraňuje nekonečnému nárastu výstupnej aktivity.

Vplyv laterálnej väzby vyjadrený vzťahom (7.1) je v konkrétnom prípade ilustrovaný na obr. 7.4a-c. Vo všetkých troch prípadoch bola vstupným stimulom reťaze zloženej zo 100 neurónov aktivita \mathbf{x} gaussovského tvaru (znázornená čiarkovane) so zložkami generovanými podľa vzťahu $x(i) = 1 \cdot \exp\left(-\frac{(i-50)^2}{2 \cdot 30^2}\right)$, pre jednoduchosť $i=1,2,\dots,100$, t.j. dimenzia vstupu rovná počtu výstupných neurónov. Výstupná aktivita \mathbf{y} každého

³ V simuláciách sa používa zjednodušený profil laterálnej spätnej väzby, ktorý má však kvalitatívne rovnaký efekt ako pôvodný "mexický klobúk". Excitačné synaptické váhy sú pri zjednodušení reprezentované kladnou konštantou, inhibičné zápornou.

neurónu bola ohraničená sigmoidálnou funkciou tvaru $y = s(\text{net}) = 10 / (1 + \exp(-2 \cdot (\text{net} - 2.5)))$. V snahe poukázať len na vplyv spätnej väzby možno vzťah (7.1) zjednodušiť uvažovaním $\mathbf{W} = \mathbf{I}$, t.j. že aktivita receptorov sa priamo prenesie na vstup neurónov. Obrázky 7.4a-c odpovedajú trom kvalitatívne rôznym prípadom vplyvu spätnej väzby (v závislosti od veľkosti laterálnej inhibície) pre tie isté hodnoty parametrov: $d_E = 10$, $d_I = 30$, $\gamma_E = 0.032$ (podľa obr. 7.2b). V prípade (a) $\gamma_I = 0,009$; v prípade (b) 0,013; v prípade (c) 0,004.



Pri správnom pomere excitácie a inhibície (prípade a) má sieť tendenciu vytvárať "zhluk" aktivít na výstupe, ktorý sa vplyvom spätnej väzby zosilňuje. Podobne, v dvojrozmernom prípade možno pozorovať nárast "aktivačnej bubliny", ktorá vzniká na mieste, v ktorom mala počiatočná výstupná aktivita neurónov maximum. Platí, že veľkosť aktivačnej bubliny ("zhuku" aktivít) závisí od "pomery síl" excitačnej a inhibičnej spätnej väzby: čím väčší vplyv excitácie, tým väčšia bublina, a naopak. Ak je laterálna inhibícia príliš silná, k vytvoreniu bubliny nedôjde a výstupná aktivita sa utlmí (prípade b). Príliš slabá inhibícia zase znamená "predimenzovanie" výstupnej aktivity (prípade c).

7.2 Kohonenov algoritmus

Kohonenov model samoorganizujúcej sa mapy, na ktorý sa zameriame, predstavuje výpočtové zjednodušenie modelu Willshawa a von der Malsburga [46], a to v dvoch krokoch. Prvý sa týka náhrady laterálnej interakcie funkciou okolia neurónov, ktorá je zahrnutá v učiacom algoritme: spätnoväzbové spojenia, ktoré sú navyše časovo náročné na simuláciu, v Kohonenovom modeli SOM nejstávajú, zato však každý neurón má definovaných svojich fyzických susedov. Druhá úprava spočíva v uvažovaní reprezentácie vstupov vo forme N -rozmerných vektorov s reálnymi zložkami (obr. 7.1b).

7.2.1 ED verzia algoritmu

Spôsob akým sa zisťuje pozícia primárnej odozvy v SOM na aktuálny podnet sa nazýva *súťaženie*. Výsledkom súťaženia v každom kroku (po predložení konkrétneho vstupu) je *vítaný neurón*, ktorý najviac reaguje na daný vstup \mathbf{x} . Jednou možnosťou je hľadať maximum výstupu lineárneho neurónu, t.j. $i^* = \operatorname{argmax}(\mathbf{w}_i^T \mathbf{x})$, kde i^* je index víťazného neurónu. O tejto verzii sa bližšie zmienime v podkapitole 7.5. V základnej, ED (angl. *Euclidean Distance*) verzii algoritmu SOM figuruje iná miera podobnosti: hľadá sa neurón, ktorého váhový vektor je najbližšie k aktuálnemu vstupu v zmysle euklidovskej vzdialenosti:

$$i^* = \operatorname{argmin}_i \|\mathbf{x} - \mathbf{w}_i\| \quad (7.2)$$

Obe miery podobnosti však navzájom súvisia: hľadanie $\max(\mathbf{w}_i^T \mathbf{x})$ odpovedá hľadaniu $\min\|\mathbf{x} - \mathbf{w}_i\|$ za predpokladu, že v prvom prípade sú váhové vektory \mathbf{w}_i normované (ležia na povrchu hypergule). Vidieť to dobre z rovnosti

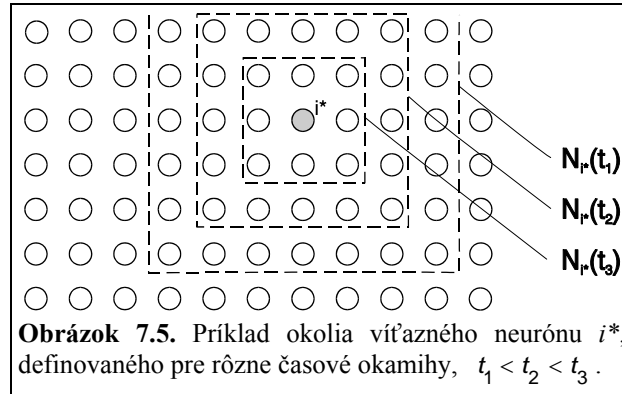
$$\|\mathbf{x} - \mathbf{w}_i\|^2 = \|\mathbf{x}\|^2 - 2\mathbf{w}_i^T \mathbf{x} + \|\mathbf{w}_i\|^2 \quad (7.3)$$

Keďže aktuálny vstup je nezávislý od i , a $\|\mathbf{w}_i\|^2$ je konštantné vďaka normovaniu, tak neurón, ktorého váhový vektor je najbližšie k vstupu \mathbf{x} , je súčasne neurónom, ktorého skalárny súčin $\mathbf{w}_i^T \mathbf{x}$ je najväčší.

Po nájdení víťaza nasleduje adaptácia váh - *učenie*. To zabezpečí, že váhové vektory víťazného neurónu a jeho topologických susedov sa posunú smerom k aktuálnemu vstupu podľa vzťahu

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t) \cdot h(i^*, i) \cdot [\mathbf{x}(t) - \mathbf{w}_i(t)] \quad (7.4)$$

Funkcia $\alpha(t) \in (0,1)$ predstavuje učiaci parameter (rýchlosť učenia), ktorý, podobne ako v iných algoritmoch pre neurónové siete, s časom klesá k nule (napr. podľa vzťahu $1/t$, resp. $\exp(-kt)$), čím sa zabezpečí ukončenie procesu učenia. Funkcia okolia $h(i^*, i)$ (obr. 7.5) definuje rozsah *kooperácie* medzi neurónmi, t.j. koľko váhových vektorov prislúchajúcich neurónom v okolí víťaza bude adaptovaných, a do akej miery.



Najjednoduchšou používanou funkciou je pravouhlé okolie

$$h(i^*, i) = \begin{cases} 1, & \text{ak } d_M(i^*, i) \leq \lambda(t) \\ 0, & \text{v ostatných prípadoch} \end{cases} \quad (7.5)$$

pričom $d_M(i^*, i)$ predstavuje vzdialenosť typu "Manhattan" medzi neurónmi i^* a i v mriežke mapy (t.j. sumu absolútnych hodnôt rozdielov ich súradníc). Na základe numerických simulácií dospel Kohonen k záveru, že najlepšie výsledky sa dosiahnu, ak sa veľkosť okolia s časom diskkrétne znižuje (priemer okolia odpovedá hodnote $2\lambda(t)$). Z obr. 7.5 taktiež vidieť, že v blízkosti okrajov mapy okolie nie je symetrické (týka sa to najmä počiatočných fáz algoritmu, keď polomer okolia je väčší), čo má za následok, ako spomenieme v ďalšom, kontrakciu váhových vektorov na okrajoch mapy.

Druhou často používanou voľbou je gaussovské okolie, ktoré možno popísať rovnicou

$$h(i^*, i) = \exp\left(-\frac{d_E^2(i^*, i)}{\lambda^2(t)}\right) \quad (7.6)$$

kde $d_E(i^*, i)$ predstavuje euklidovskú vzdialenosť neurónov i^* a i v mriežke, t.j. $d_E(i^*, i) = \|\mathbf{r}_{i^*} - \mathbf{r}_i\|$, kde \mathbf{r}_i označuje vektor súradníc i -teho neurónu v SOM. Parameter $\lambda(t)$ klesá s časom k nule, čím sa zabezpečuje znižovanie okolia počas učenia.

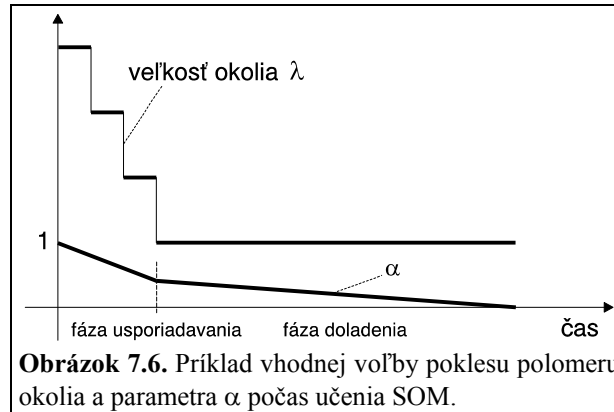
Algoritmus učenia teda spočíva v troch základných krokoch, ktoré sa opakujú po predložení každého vstupného vzoru, vybraného náhodne z tréningovej množiny vzorov (počiatočné hodnoty váh sú náhodné malé čísla):

- 1) nájdenie víťaza medzi neurónmi (vzťah 7.2),
- 2) adaptácia váh víťazného neurónu a jeho topologických susedov (vzťah 7.4),
- 3) aktualizácia parametrov učenia (h, α).

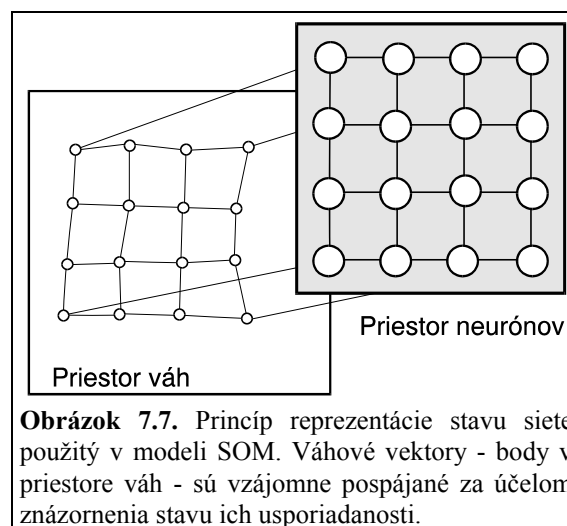
7.2.2 Voľba parametrov učenia

Spôsob, akým je vhodné upravovať veľkosť okolia λ i parametra učenia α je zhruba znázornený na obr. 7.6. Ako vidieť, v procese učenia možno rozlíšiť dve fázy. Počas prvej, nazývanej *fáza usporiadavania*, klesá veľkosť okolia diskkrétne s časom. Počas druhej fázy - *fázy doladenia* - možno ponechať najbližších susedov súčasťou okolia, až kým učenie neskončí. Na funkcii poklesu parametra učenia α v praxi až tak veľmi nezáleží (pozri podmienky (7.15) a (7.16)), dôležité je, aby to bola monotónne klesajúca funkcia z nejakej hodnoty blízkej 1, s malými hodnotami (rádovo 0,1-0,01) počas fázy doladenia. Možnou voľbou je napr. lineárna lomená funkcia (obr. 7.6), exponenciálna funkcia atď.

Na presnom počte iterácií takisto nezáleží. Kohonen uvádza empiricky získanú pomôcku, podľa ktorej počet iterácií má byť minimálne 500-násobok počtu neurónov v sieti. Bežne sa počet iterácií pohybuje v rozmedzí rádo 10000–100000. Dôležité je, aby počas fázy usporiadavania, pokiaľ je ešte parameter α relatívne veľký, sieť "stihla" správne zoradiť svoje váhové vektory, ktoré sa v zvyšnom čase lokálne doladia. Na základe simulácií sa takisto ukázalo, že je vhodné rozdeliť celkovú dobu tréningu tak, že na fázu doladenia sa ponechá viac času ako na prvú fázu.



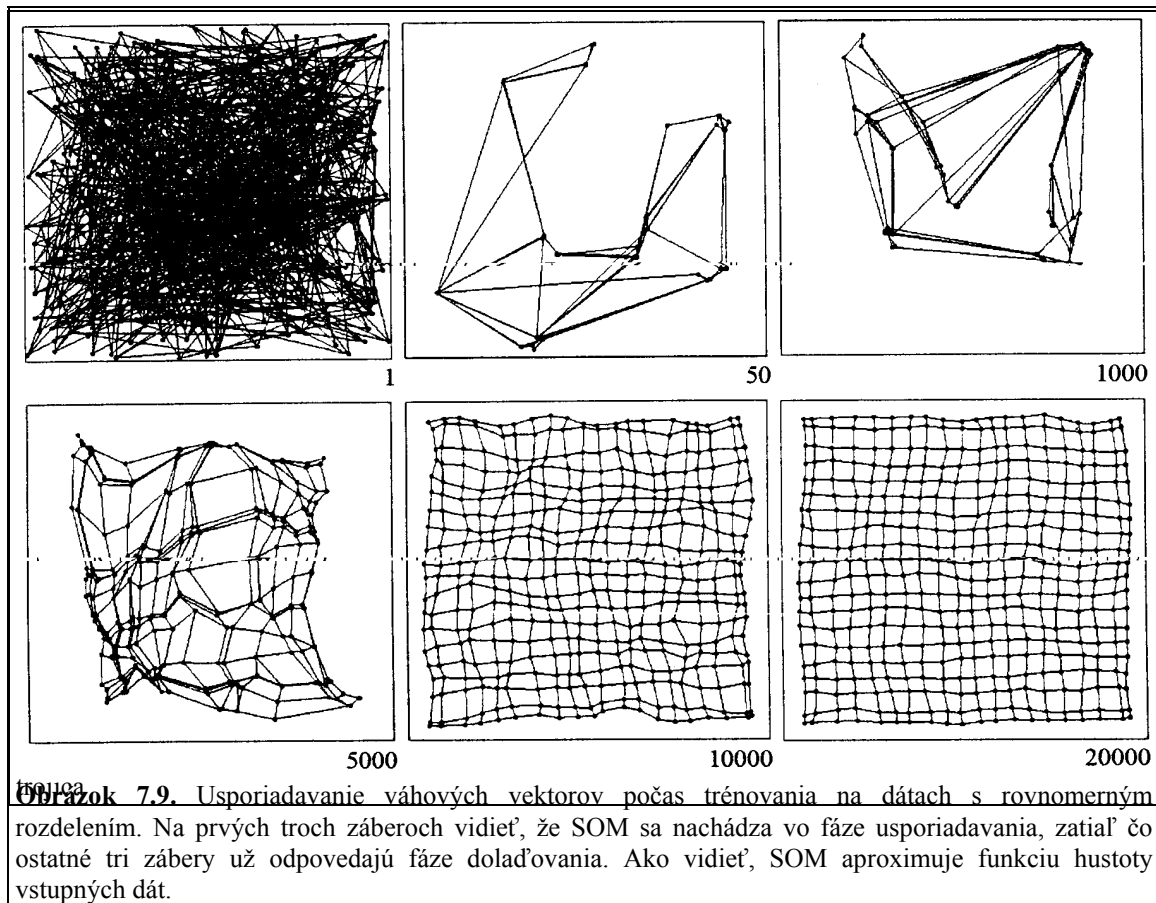
7.3 Príklady jednoduchých zobrazení



Z matematického hľadiska predstavuje SOM zobrazenie z množiny vstupných vzorov na diskretnú množinu neurónov, t.j. $\xi: \mathbf{X} \subset \mathcal{R}^N \rightarrow \mathfrak{S} = \{i, 1, 2, \dots, n\}$, kde N je dimenzia vstupov, a n počet neurónov v sieti⁴. Takto máme do činenia s dvoma priestormi – priestorom vstupov \mathbf{X} , ktorý obyčajne predstavuje časť euklidovského priestoru, a priestorom neurónov, ktorý je definovaný topológiou ich usporiadania (mriežka, reťaz). Keďže dimenzia váh je zhodná s dimenziou vstupov (ako vyplýva i z rovnice 7.4), možno vstupný priestor chápať súčasne ako priestor váh. Stav SOM sa vyjadruje váhovými vektormi a tie možno zobrazit' ako body v priestore váh. Situáciu znázorňuje obr. 7.7.

⁴ Ako vidieť, v modeli SOM je výstupná informácia reprezentovaná inak ako v ostatných modeloch ako napr. viacvrstvový perceptrón, či nejaká lineárna dopredná sieť. Namiesto výstupnej aktivity siete tu nás zaujíma len pozícia víťazného neurónu. V snahe nazrieť na SOM ako na systém realizujúci transformáciu $\varphi: \mathbf{X} \rightarrow \mathbf{Y}$ i tu je možné vypočítavať výstupnú aktivitu i -tého neurónu v tvare $y_i = f(\|\mathbf{x} - \mathbf{w}_i\|)$, kde f je monotónne klesajúca funkcia. Tento alternatívny prístup, v ktorom je výstupná informácia SOM reprezentovaná celkovou aktivitou všetkých neurónov, t.j. vektorom \mathbf{y} (a teda nie pozíciou víťaza i^*) je výpočtovo náročnejší, ale biologicky prijateľnejší [17].

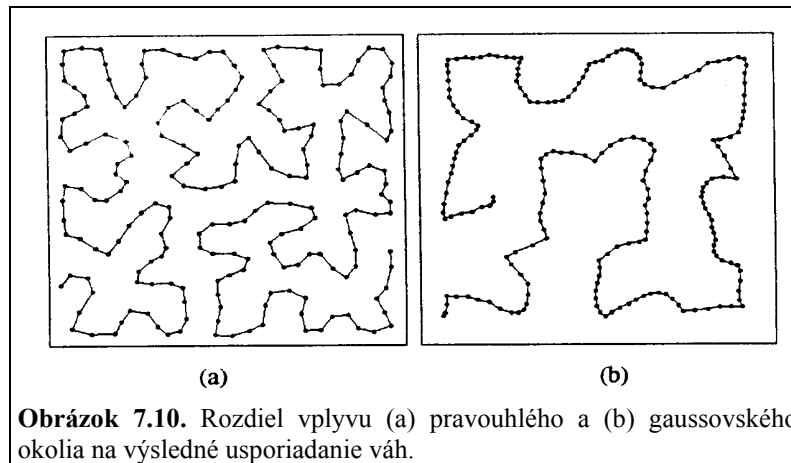
Spájajúce čiary medzi váhovými vektormi znázorňujú skutočnosť, že každé dva spojené váhové vektory prináležia dvojici neurónov, ktoré sú bezprostrednými susedmi v mriežke. Pri pohľade na zobrazené váhy takto dostávame vizuálnu informáciu o ich rozmiestnení i vzájomnom usporiadaní. Podobne vo výstupnom priestore, neuróny nie sú fyzicky spojené, spojnice len znázorňujú ich vzájomné topologické vzťahy, s ktorými súvisí definované okolie.



Ak budeme pre začiatok uvažovať, že máme dvojrozmerné vstupy $\mathbf{x} = [x_1, x_2]^T$, môžeme priestor váh priamo zobraziť v rovine. Na obr. 7.8 sú príklady portrétov váh SOM (dolná trojica obrázkov) získaných po natrénovaní na dátach s rovnomerným rozdelením (horná trojica obrázkov). V každom z príkladov boli počas tréningu jednotlivé vstupné vzory vybrané náhodne z tréningovej množiny. Je vidieť, že váhy majú tendenciu aproximovať funkciu hustoty tréningovej množiny.

Na sekvencii obrázkov 7.9 je ilustrovaný proces usporiadavania a doladovania váhových vektorov SOM tréningovej na "štvorcových" dátach. Je vidieť, že už približne po štvrtine celkového počtu iterácií (20000) sieť prechádza do fázy doladovania.

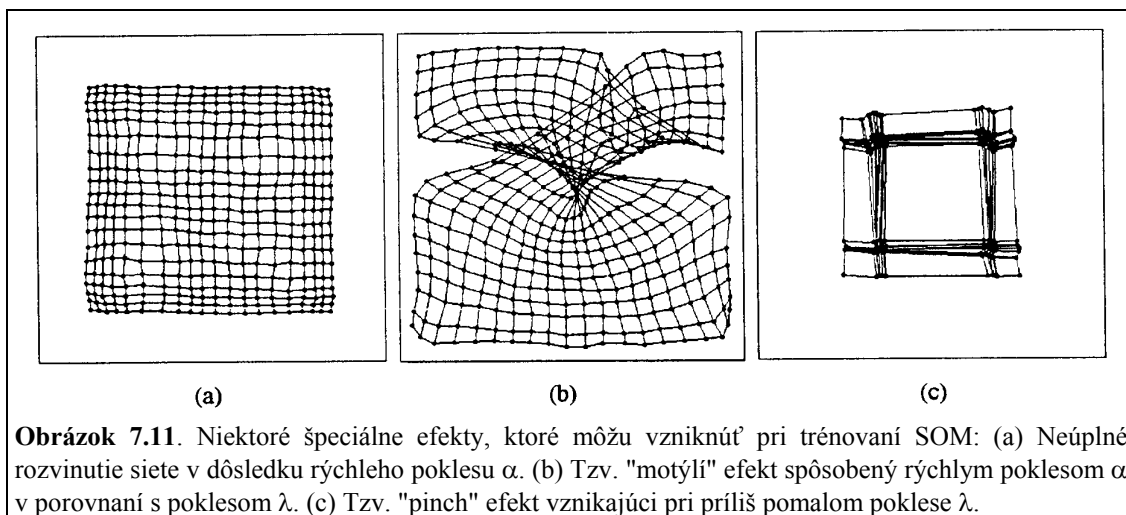
Ďalším zaujímavým fenoménom, ktorý je možné u SOM ilustrovať, je rozdielny vplyv pravouhlého (rovnicu 7.5) a gaussovského (7.6) okolia, parametrizovaných hodnotou λ . Väčší rozsah gaussovského okolia má za následok vznik "stiahnutejšej" reprezentácie pri rovnakom počte neurónov (obr. 7.10).



7.3.1 Niektoré špeciálne efekty

I keď proces samoorganizácie v SOM prebehne pri pomerne širokom rozpätí jednotlivých voliteľných parametrov, existujú podmienky, pri ktorých proces zlyhá. Na obr. 7.11a-c sú príklady nežiadúcich efektov, ktoré môžu vzniknúť. Všetky 3 simulácie boli nastavené na 20000 iterácií na dátach s rovnomerným rozdelením, pri použití siete 20×20 neurónov.

Efekt na obr. 7.11a vznikol v dôsledku príliš rýchleho poklesu parametra α (pomer dĺžok prvej a druhej fázy učenia bol 1:500), pri pravouhlom okolí so štandardným poklesom polomeru okolia λ . Na obr. 7.11b je tzv. "motýlí" efekt (dokonca kvázi dvojité)⁵, ktorý môže vzniknúť, ak necháme príliš rýchlo klesať λ v porovnaní s α (v simulácii bol pokles λ na hodnotu 1 pri gaussovskom okolí už po prvej stotine iterácií, zatiaľ čo α klesalo štandardne). Obr. 7.11c ilustruje tzv. "pinch" efekt, ktorý bol docielený vďaka pomalému poklesu λ (v simulácii mal parameter λ hodnotu 10 počas prvej polovice iterácií, potom jeho hodnota poklesla na 9).



7.3.2 Hraničný efekt

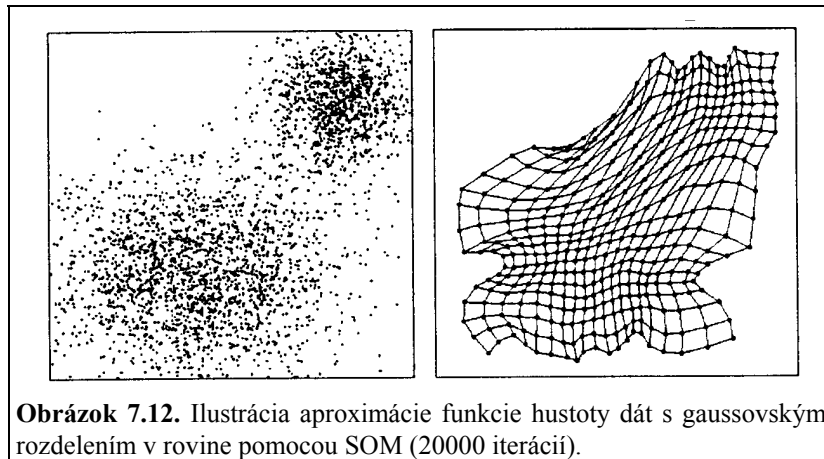
Pri pohľade na výsledné stavy sietí na obr. 7.8 vidieť, že ich okrajové časti sú mierne kontrahované smerom dovnútra. Tento okrajový defekt je dôsledkom asymetrie okolia (obr. 7.5), ktorá spôsobuje, že váhové vektory okrajových neurónov sú štatisticky v priemere viac adaptované smerom dovnútra siete. Ak je tento efekt nežiadúci, možno ho odstrániť tak, že sa okolie algoritmicky uzavrie do slučky. Potom nastane situácia, že všetky neuróny budú pozíčne ekvivalentné: v prípade reťaze bude mať *každý* neurón dvoch bezprostredných

⁵ Použitý termín nemá nijaký súvis s rovnomenným termínom používaným v teórii chaosu.

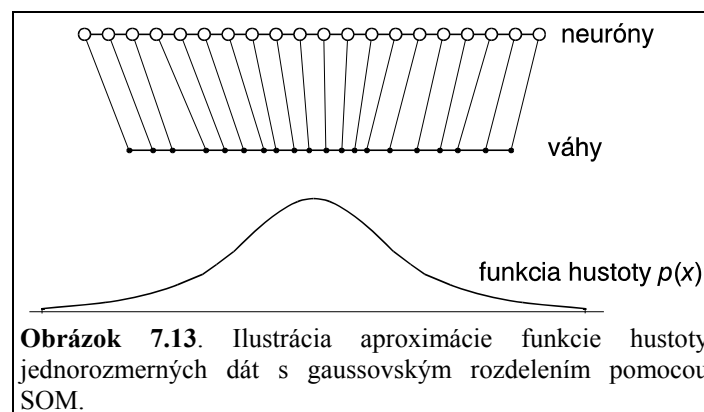
susedov, v prípade mriežky ôsmich. Podobne sa hraničný efekt prejavuje i v modeli s laterálnou inhibíciou, kde to možno obísť analogicky – uzavretím spätnej väzby do slučky.

7.3.3 Magnifikačný faktor

Na základe simulácií na dátach s rovnomerným rozdelením bolo vidieť, že váhové vektory mali tendenciu rovnomerne pokryť "trénovacia oblasť" (obr. 7.8). Inými slovami, SOM sa snažila aproximovať funkciu hustoty vstupných dát. Tento efekt sa prejavuje i v prípade nerovnomernej distribúcie vstupných dát, čoho výsledkom je odpovedajúce rozloženie váhových vektorov. Na obr. 7.12 je znázornená situácia po natrénovaní siete 20×20 neurónov na 2D dátach zložených z dvoch gaussovských rozdelení. Ako vidieť, v oblastiach centier zhlukov je zhustené rozloženie váhových vektorov siete. Takúto tendenciu algoritmu SOM možno interpretovať ako snahu o optimálne rozloženie svojich zdrojov.



Obrázok 7.12. Ilustrácia aproximácie funkcie hustoty dát s gaussovským rozdelením v rovine pomocou SOM (20000 iterácií).



Obrázok 7.13. Ilustrácia aproximácie funkcie hustoty jednorozmerných dát s gaussovským rozdelením pomocou SOM.

Túto vlastnosť algoritmu popisuje *magnifikačný faktor* (počet váhových vektorov pripadajúcich na jednotkovú plochu vstupného priestoru), ktorý nie je v tomto prípade konštantný, ale je funkciou pozície v mape. V jednorozmernom prípade sa dá povedať, že je zhruba úmerný funkcii hustoty dát $p(x)$, ako vidieť na obr. 7.13. Tým, že si sieť "vyhradí" proporcionálne väčší počet neurónov na reprezentáciu viac zahustených oblastí, tým je jej citlivosť na zmenu dx (zmena víťazného neurónu pri malej zmene x) v týchto miestach väčšia, a naopak.⁶

7.4 Niektoré aplikácie SOM

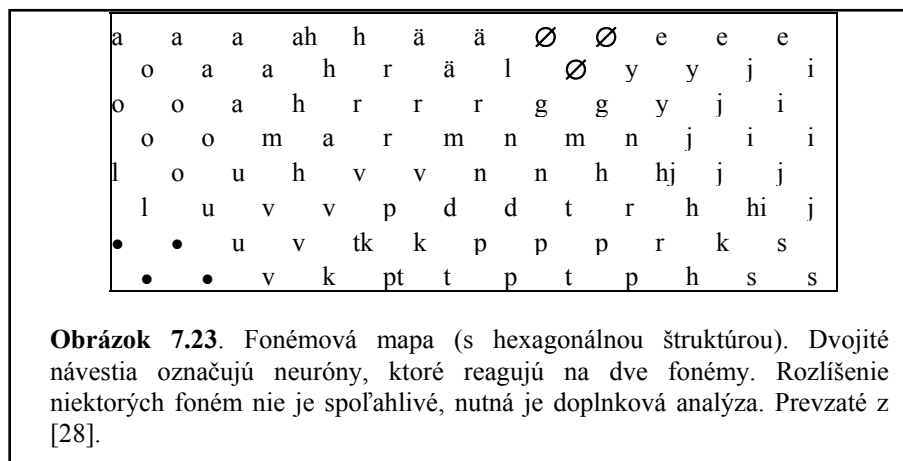
Aplikácii SOM je veľa, referencie na ne možno nájsť napr. v [30]. Tu spomenieme aspoň niektoré z nich.

⁶ V biologických sieťach predstavuje tento efekt dôležitú vlastnosť. Prejavom takejto reprezentácie napr. v somatosenzorickej kôre je skutočnosť, že organizmus má na svojom povrchu fyzicky viac i menej citlivé miesta.

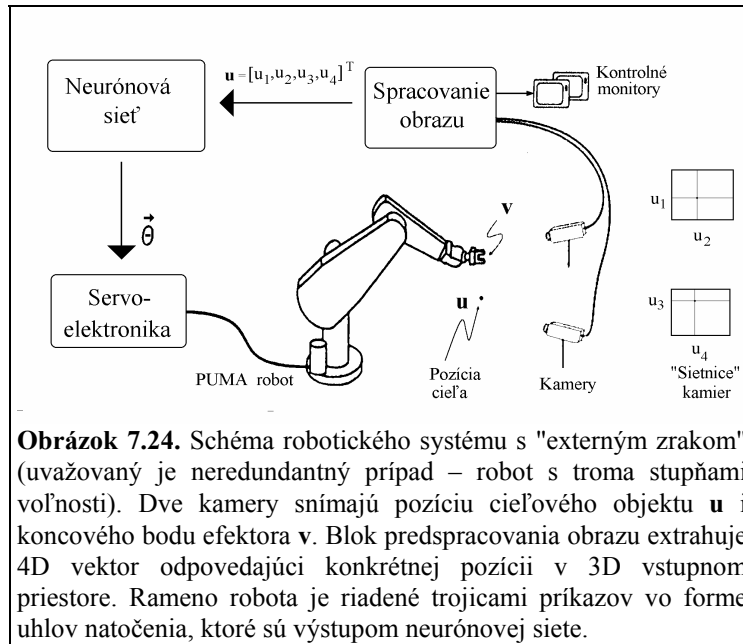
Rozpoznávanie reči [28]. Táto aplikácia SOM spočíva v transformácii akustického rečového signálu na sekvenciu hlások, tzv. fonetický prepis (ako napr. v slovenčine sekvencia d'-j-e-u-č-a). Prostriedkom tejto transformácie je natrénovaná SOM – tzv. *fonémová mapa* (obr. 7.23), ktorá svojou odpoveďou je schopná lokalizovať odozvu na aktuálny vstup \mathbf{x} odpovedajúci nejakej hláske (fonéme). Aplikácia bola urobená pre fínsky a japonský jazyk, a to predovšetkým z dvoch dôvodov: (a) oba jazyky sú inflexnými jazykmi (t.j. príbuzné slová sa v nich tvoria pomocou predpôň a prípon), preto je výhodnejšie voliť malé akustické jednotky (hlásky, slabiky) ako objekty rozpoznávania, lebo ich počet je obmedzený (rozpoznávanie na úrovni slov by prestávalo byť efektívne pri ich vysokom počte, pretože každý tvar slova treba považovať sa samostatnú kategóriu); (b) oba jazyky obsahujú fonémy, ktoré sú ľahko rozlíšiteľné na základe ich stacionárnych spektrálnych vlastností.

Na to, aby bolo možné trénovať SOM, je potrebné rečový signál predspracovať. Vo všeobecnosti existuje viacero spôsobov predspracovania rečového signálu, ktoré sa bežne používajú. Kohonen použil spektrálnu analýzu. Predspracovanie signálu (počnúc krokom (3) realizované na signálovom procesore) spočívalo v nasledovných krokoch: (1) dolnopriepustná filtrácia s $f = 5,3$ kHz, (2) 12-bitová AD konverzia s frekvenciou vzorkovania 13,02 kHz, (3) 256-bodová rýchla Fourierova transformácia (FFT) každých 9,83 ms používajúc Hammingovo okno, (4) logaritmizácia spektra a jeho vyhladenie, (5) umiestnenie 15 zložiek vhodným zlúčením komponent FFT v rozsahu 200-5000 Hz do spektrálneho vektora \mathbf{x} , (6) centrovanie \mathbf{x} okolo strednej hodnoty (odčítaním strednej hodnoty od všetkých zložiek), (7) normovanie \mathbf{x} .

Takýmto spôsobom sa z každého okna konštantnej dĺžky 9,83 ms (ktorú možno v reči považovať za štatisticky stacionárny úsek) získala spektrálna reprezentácia vo forme vektora $\mathbf{x} \in \mathfrak{R}^{15}$. Po natrénovaní SOM na takejto množine vektorov boli jednotlivým neurónom priradené návestia podľa väčšinového pravidla, a váhy SOM boli pred rozpoznávaním doladené pomocou LVQ (podkapitola 7.7). Pri testovaní SOM predstavovali odozvy (reprezentované pozíciou víťazných neurónov) v SOM trajektóriu, ktorej uzly odpovedali jednotlivým po sebe nasledujúcim vstupným vektorom \mathbf{x} . Z hľadiska nadobudnutej reprezentácie je zaujímavé, že súradnice SOM nemajú explicitný význam – sieť si ich vybrala automaticky počas tréningu. Z výstupnej fonetickej sekvencie bolo možné dosiahnuť ortografický prepis s 90%-nou presnosťou, pričom po odstránení koartikulačných efektov pomocou gramatických pravidiel dosiahol implementovaný *fonetický písací stroj* presnosť v rozsahu 92–97% v reálnom čase.



Robotika. Jednou zo základných úloh v tejto oblasti je naučiť robota umiestniť svoje koncové rameno (efektor) do žiadanej polohy. Schéma realizovaného systému na riešenie tohto problému s použitím SOM je na obr. 7.24 [45]. Cieľom je naučiť neurónovú sieť realizovať transformáciu $\mathcal{G}: \mathbf{u} \in U \subseteq \mathfrak{R}^4 \rightarrow \theta \in \mathfrak{R}^3$ bez učiteľa. Prístup spočíva v adaptívnom kvantovaní vstupného priestoru U na N disjunktných oblastí $F_i, i \in \{1, \dots, N\}$ a aproximácii θ v každej oblasti lineárnym zobrazením, ktoré sa postupne "dolaďuje". Počet oblastí N je daný zvoleným počtom neurónov SOM (s 3D architektúrou), z ktorých každému je priradený váhový vektor $\mathbf{w}_i \in \mathfrak{R}^4$ aproximujúci ťažisko oblasti F_i , výstupný vektor θ_i a matica \mathbf{A}_i (typu 3×4), ktoré spolu určujú lineárny Taylorov rozvoj $\theta(\mathbf{u})$ v rámci F_i :



Obrázok 7.24. Schéma robotického systému s "externým zrakom" (uvažovaný je neredundantný prípad – robot s tromi stupňami voľnosti). Dve kamery snímajú pozíciu cieľového objektu \mathbf{u} i koncového bodu efektora \mathbf{v} . Blok spracovania obrazu extrahuje 4D vektor odpovedajúci konkrétnej pozícii v 3D vstupnom priestore. Rameno robota je riadené trojicami príkazov vo forme uhlov natočenia, ktoré sú výstupom neurónovej siete.

$$\theta(\mathbf{u}) = \theta_i + \mathbf{A}_i \cdot (\mathbf{u} - \mathbf{w}_i) \quad (7.28)$$

Na adaptáciu váh neurónov ako aj ich výstupov θ_i a \mathbf{A}_i je použitý rozšírený algoritmus SOM¹¹ (ako aj algoritmus "neural gas" [34])

$$\begin{aligned} \mathbf{w}_i &\leftarrow \mathbf{w}_i + \varepsilon \cdot h(i, i^*) \cdot (\mathbf{u} - \mathbf{w}_i) \\ \theta_i &\leftarrow \theta_i + \varepsilon \cdot h(i, i^*) \cdot \Delta\theta_i \\ \mathbf{A}_i &\leftarrow \mathbf{A}_i + \varepsilon \cdot h(i, i^*) \cdot \Delta\mathbf{A}_i \end{aligned} \quad (7.29)$$

kde ε a $h(i^*, i)$ sú štandardné parametre SOM, veličiny $\Delta\theta_i, \Delta\mathbf{A}_i$ sa vypočítajú spôsobom bližšie opísaným v [45]. Učenie prebieha tak, že v každom tréningovom kroku sa zadá cieľová pozícia \mathbf{u} a rameno robota sa iteratívne približuje k cieľu. V prvom priblížení sa nájde víťaz a požadované uhly sa vypočítajú podľa (7.28). Ďalej sa generovaný výstup získa iteratívnym spôsobom, a to váhovaným spriemernením príspevkov od všetkých neurónov v okolí víťaza podľa rekurentného vzťahu

$$\theta_n^{out} = \theta_{n-1}^{out} + s^{-1} \cdot \sum_i h(i^*, i) \cdot \mathbf{A}_i \cdot (\mathbf{u} - \mathbf{v}_n) \quad (7.30)$$

kde $s = \sum_i h(i^*, i)$ a \mathbf{v}_n je pozícia efektora v n -tej iterácii. Zakaždým sa aplikujú učiace pravidlá (7.29). Po "predložení" asi 3000 cieľových pozícií je robot schopný dosiahnuť žiadanú pozíciu s presnosťou 1,3 mm, pričom, ako uvádzajú autori, tá by sa dala ešte zvýšiť použitím kamier s lepšou rozlišovacou schopnosťou.

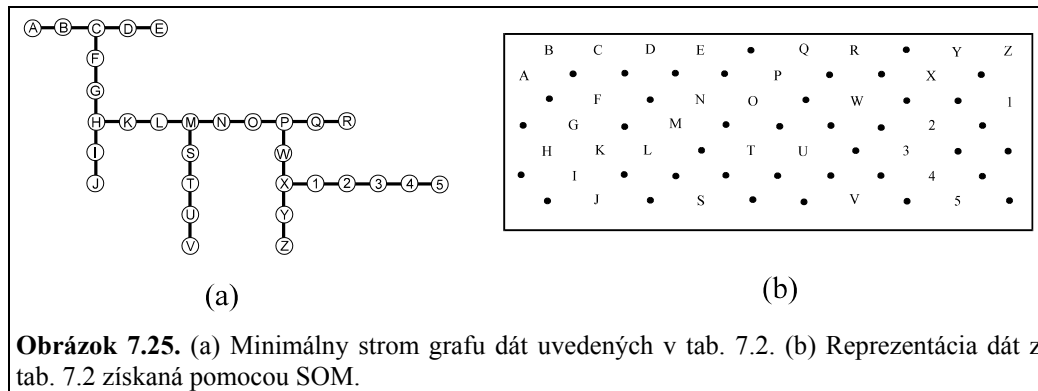
Tabuľka 7.2. Množina prvkov použitá na formovanie hierarchickej reprezentácie. Každý z prvkov (ozn. ako A, B, C,...) predstavuje 5-rozmerný vektor so súradnicami danými v odpovedajúcom stĺpci pod ním.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	
1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0	0	0	0	0	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	3	3	3	3	6	6	6	6	6	6	6	6	6	6	6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	1	2	3	4	2	2	2	2	2	2	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	5

¹¹ Rozšírenie algoritmu SOM spočíva práve v priradení maticového výstupu jednotlivým neurónom a využití vlastnosti zachovania topológie pri ich adaptácii.

Formovanie hierarchických reprezentácií. SOM je schopná topologicky zobraziť hierarchické vzťahy medzi jednotlivými vstupnými prvkami, pokiaľ sú tieto vhodne popísané pomocou svojich súradníc [25]. Na ilustráciu tejto úlohy možno použiť reprezentáciu podľa tab. 7.2.

Vzťahy medzi jednotlivými prvkami množiny možno zobraziť klasickou metódou, čím získame, minimálny strom (obr. 7.25a). Pri použití SOM na tých istých dátach dostaneme mapu ako napr. na obr. 7.25b. Podobnosť medzi oboma vyobrazeniami je zrejmá: jednotlivé "vetvy" v SOM sú síce všelijako stočené (tak, aby sa vošli do mapy), avšak topologické vzťahy medzi susednými vzormi sú v podstate rovnaké.



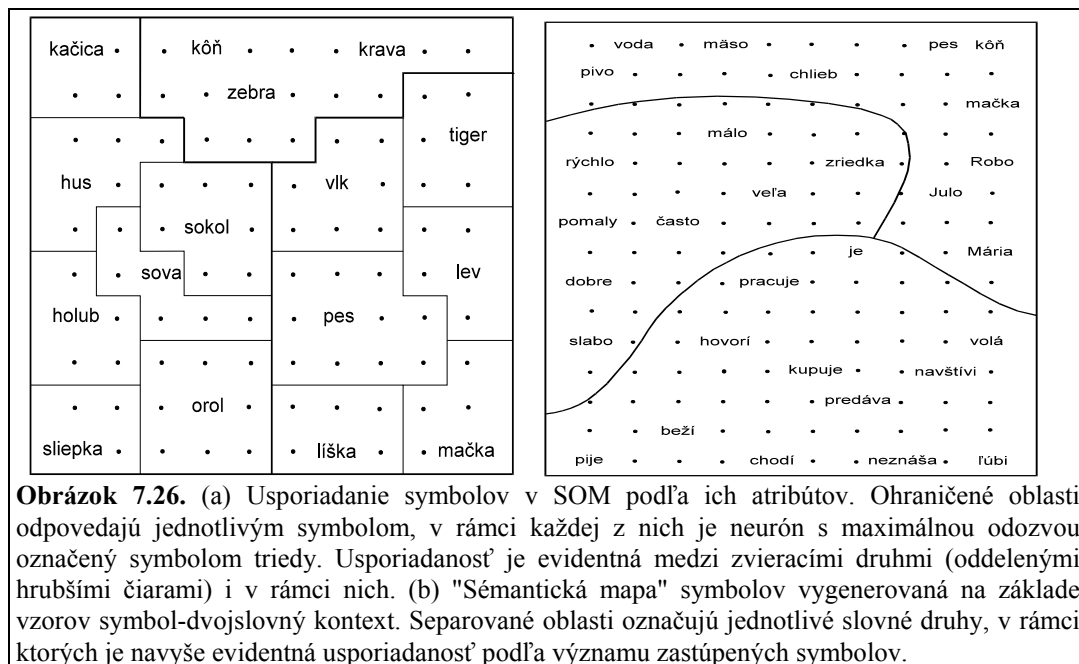
Obrázok 7.25. (a) Minimálny strom grafu dát uvedených v tab. 7.2. (b) Reprezentácia dát z tab. 7.2 získaná pomocou SOM.

Topografické mapy abstraktných dát. V prípade abstraktných, symbolických dát vzniká otázka, ako možno u tých zisťovať a zobrazovať ich vzájomné, sémantické vzťahy [40]. V prípade fyzikálne relevantných dát to nie je problém, pretože ich samotná reprezentácia odzrkadľuje vzájomné vzťahy podobnosti (napr. blízkosť odpovedajúcich súradníc dvoch vektorov v zmysle Euklidovskej metriky). Avšak v prípade symbolov (napr. slov prirodzeného jazyka) to neplatí, pretože medzi kódom symbolu (reprezentovaného napr. písomnou formou) a jeho významom nie je žiadna súvislosť. Keďže vzťahy medzi symbolmi nie sú zistiteľné z ich kódových reprezentácií, je potrebné ich prezentovať v spätosti s *kontextom*, v ktorom sa vyskytujú.

V prvom príklade bol kontext každého symbolu (meno zvierat'a) reprezentovaný vektorom binárnych *atribútov* (prítomnosť atribútu označená jednotkou, absencia nulou) ako veľkosť zvierat'a (malé, stredné, veľké), vonkajší popis tela (má 2 nohy, 4 nohy, srst', kopytá, hrivu, perie) a čo rado robí (loví, behá, lieta, pláva). Takýto 13-rozmerný vektor atribútov \mathbf{x}_a bol vygenerovaný pre každé zviera, pričom kódy zvierat \mathbf{x}_s boli zámerné vytvorené tak, aby neniesli žiadnu informáciu o vzájomnej podobnosti medzi symbolmi: každý vektor \mathbf{x}_s obsahoval samé nuly, až na jednu hodnotu a , ktorá figurovala na pozícii udávajúcej poradové číslo zvierat'a (1-16). Oba vektory boli zlúčené v jeden 29-rozmerný vektor $\mathbf{x} = [\mathbf{x}_s, \mathbf{x}_a]^T$ charakterizujúci každé zviera, pričom hodnota a bola stanovená na $a = 0,2$, aby vplyv atribútovej časti vektora \mathbf{x} (nositeľ informácie o zvierati) bol väčší ako vplyv symbolovej časti. Vektory \mathbf{x} boli napokon normované kvôli lepšej stabilizácii učenia. Počas tréningu bolo SOM prezentovaných 2000 náhodne vybraných vzorov \mathbf{x} z 16-prvkovej množiny. Proces určovania návestí bol však realizovaný na základe vektorov $\mathbf{x} = [\mathbf{x}_s, \mathbf{0}]^T$, čoho výsledkom je mapa na obr. 7.26a. Kvalitatívne rovnakú reprezentáciu by sme dostali i pri použití $\mathbf{x} = [\mathbf{x}_s, \mathbf{x}_a]^T$, z čoho vyplýva, že hoci reprezentácia vzájomných vzťahov podobnosti bola získaná vďaka prítomnosti atribútových častí počas tréningu, správna odozva SOM v testovacej fáze sa generuje i pri absencii \mathbf{x}_a , t.j. len na základe symbolovej časti.

V druhej ukážke je kontext symbolu reprezentovaný pomocou iných symbolov, ako to možno pozorovať v prirodzenom jazyku. Uvažovaná množina 30 symbolov zahrňovala podstatné mená, slovesá a príslovky. Generované tréningové vzory pozostávali zo zmysluplných trojslovných viet (napr. Robo pomaly beží, lev je mäso, atď.), pričom každý z troch symbolov bol nejako kódovaný ako 7-rozmerný vektor. Opäť, aby sa zvýraznil vplyv kontextu, bol parameter a v symbolovej časti stanovený na $a = 0,2$. Po natrénovaní na 2000 (21-rozmerných) vstupných vzoroch tvaru $\mathbf{x} = [\mathbf{x}_s, \mathbf{x}_a]^T$ boli návestia určené opäť len na základe symbolovej časti a výsledkom je mapa na obr. 7.26b.

Z uvedených príkladov vyplýva, že SOM možno použiť i na generovanie topologicky usporiadaných zobrazení symbolických dát za predpokladu, že tie sú prezentované v kontexte, ktorý nejakým spôsobom popisuje vzťahy podobnosti medzi nimi.



7.5 Príbuzné algoritmy

I napriek úspešnosti použitia SOM v rôznych aplikáciách má algoritmus SOM niektoré nedostatky. Patria k nim najmä tieto fakty:

- forma štruktúry, dimenzia mapy i počet neurónov SOM sú definované *a priori*
- možnosť vzniku "mŕtvych" neurónov
- diskretnosť a "rovnosť" projekcie.

Pri použití SOM sa *a priori* predpokladá, že vstupné dáta ležia (najčastejšie) na dvoj- alebo jednorozmernom podpriestore, čo je nutná ale nepostačujúca podmienka úspešnej použiteľnosti SOM, a podľa toho sa volí dimenzia mapy (v prípade 2D obyčajne štvorcového tvaru). Ak majú dáta naozaj odhadovanú dimenziu a navyše im "vyhovuje" štandardná štruktúra mapy (myslia sa tým bežne používané formy reťaze a mriežky), potom SOM predstavuje efektívny a ilustratívny prostriedok zobrazenia ich topologických vzťahov, čo možno využiť napr. pri následnej klasifikácii v tomto výstupnom priestore. Použitie SOM ako prostriedku na analýzu dát je však obmedzené. Obmedzenosť jej všeobecného použitia spočíva práve v tom, že vyžaduje apriórny odhad inherentnej dimenzie dát – informácie, ktorá má byť práve jedným z výsledkov ich analýzy! Takisto, zvolený počet neurónov nemusí byť optimálny: príliš málo neurónov znižuje presnosť aproximácie, ich privysoký počet spôsobuje nárast výpočtovej zložitosti s možnosťou neadekvátneho zvýšenia presnosti aproximácie.

Fixovanie formy štruktúry (štvorcová, obdĺžniková) môže mať v závislosti od štruktúry dát za následok vznik tzv. "mŕtvych" neurónov (napr. ak dáta pozostávajú zo vzdialených zhlukov), t.j. takých, ktorých váhové vektory zakotvia v oblastiach s nulovou pravdepodobnosťou výskytu vstupov. Inými slovami, ide o neuróny, ktoré nikdy nezvíazia, preto ostávajú nevyužitú.

Diskretnosť a "rovnosť" projekcie znamená, že konkrétny vstupný vektor sa premietne na jediný (víťazný) neurón, ktorého súradnice v mape môžu nadobúdať len rovnomerne vzdialené diskkrétne hodnoty (1,2,...). V niektorých aplikáciách môže byť táto skutočnosť nevýhodou (nedostatočná presnosť), čo je na druhej strane možné eliminovať plošným zvýšením počtu neurónov.

Uvedené fakty boli podnetmi pre návrh príbuzných algoritmov samoorganizácie v snahe odstrániť tieto nedostatky, alebo aspoň niektoré z nich. Stručne sa o niektorých z nich zmienime.

VQP (Vector Quantization and Projection) [13]. Výstižný názov napovedá, že vstupné dáta sú najprv vektorovo kvantované (vo vstupnom priestore), a potom projektované do priestoru nižšej dimenzie, a to s cieľom zachovania topológie. Spoločnými znakmi so SOM je to, že dimenziu výstupného priestoru a počet neurónov *treba tiež a priori zvoliť*, avšak odlišnosť spočíva v tom, že VQP *nepredpokladá* žiadnu geometrickú štruktúru neurónov (teda ani funkcia okolia tu nefiguruje). Každému neurónu i je priradený vstupný vektor $\mathbf{x}_i \in \mathfrak{R}^N$ (odpovedajúci váhovému vektoru \mathbf{w}_i v SOM) a výstupný vektor $\mathbf{y}_i \in \mathfrak{R}^P$ (odpovedajúci vektoru \mathbf{r}_i v SOM), pričom $P < N$. Neuróny na výstupe teda nemajú zafixované pozície, navyše v pravidelnej štruktúre ako u SOM, ale určenie ich optimálnych pozícií je práve cieľom algoritmu VQP. Prvým krokom je vektorová kvantizácia vstupného priestoru (t.j. nájdenie prototypov \mathbf{x}_i), na ktorú autori aplikovali vylepšenú modifikáciu algoritmu "neural gas" [34]. Následná projekcia prototypov, ktorej kritériom je zachovanie lokálnej topológie, sa odvádza z minimalizácie chybovej funkcie tvaru

$$E = \frac{1}{2} \sum_i \sum_{i \neq j} (X_{ij} - Y_{ij})^2 \cdot F(Y_{ij}) \quad (7.31)$$

pričom $X_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$ a $Y_{ij} = d(\mathbf{y}_i, \mathbf{y}_j)$, kde $d(\dots)$ predstavuje euklidovskú vzdialenosť. $F(\dots)$ je pozitívna, monotónne klesajúca funkcia, aby podporovala zachovanie lokálnej topológie: čím menšia vzájomná vzdialenosť dvoch bodov na výstupe, tým väčší je ich príspevok v chybovej funkcii. Výsledkom štandardnej gradientovej metódy je vzťah pre $\Delta \mathbf{y}_i = \alpha \sum_{j \neq i} G(X_{ij}, Y_{ij})(\mathbf{y}_i - \mathbf{y}_j)$, z čoho vyplýva výpočtová zložitosť rádu

$O(n^2)$, lebo ako vidieť, adaptujú sa všetky neuróny, pričom posun každého z nich sa vypočítava ako sumácia cez všetky ostatné neuróny. V snahe znížiť výpočtovú zložitosť pri zachovaní presnosti dospeli autori k myšlienke vybrať v každom adaptačnom kroku len jeden neurón (víťaza pre aktuálny vstup) a adaptovať všetky ostatné neuróny podľa pravidla (toho istého, ale bez sumácie)

$$\Delta \mathbf{y}_j = \alpha \frac{X_{ij} - Y_{ij}}{Y_{ij}} [2F(Y_{ij}) - (X_{ij} - Y_{ij}) \cdot F(Y_{ij})] (\mathbf{y}_j - \mathbf{y}_i) \quad \forall j \neq i \quad (7.32)$$

čím sa výpočtová zložitosť znížila na $O(n)$. V porovnaní so SOM je algoritmus VQP rýchlejší, a vďaka "voľnej" geometrii výstupných vektorov je schopný aproximovať dátové štruktúry bez vzniku "mŕtvych" neurónov.

TRN (Topology Representing Network) [35]. Na tento algoritmus sa možno pozeráť ako na aproximáciu dátovej štruktúry pomocou (neorientovaného) grafu. Jeho výsledkom sú pozície uzlov (neurónov), ktorých počet n sa *vopred stanoví* (ako u SOM), a štvorcová matica prepojení $C[n \times n]$ definujúca existenciu hrán medzi nimi (t.j. ak hrana medzi uzlami i, j existuje, potom $C_{ij} > 0$, inak $C_{ij} = 0$), pričom podmienkou je zachovanie topológie (v zmysle definície uvedenej v podkapitole 7.6). Avšak na rozdiel od SOM i VQP, výstupná reprezentácia (pozície uzlov) má dimenziu vstupného priestoru, teda N . Inými slovami, generovaná projekcia (na uvažovaný graf) nie je sprevádzaná explicitnou redukciou dimenzie. TRN je kombináciou algoritmu "neural gas" [34], ktorým sa adaptujú uzly, a Hebbovho učenia so súťažiením,¹² ktoré slúži na vytváranie nových resp. elimináciu existujúcich hrán. Adaptácia uzlov sa realizuje podľa vzťahu (podobného ako u SOM)

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t) \cdot \exp(-k_i / \lambda(t)) [\mathbf{x} - \mathbf{w}_i(t)] \quad (7.33)$$

kde parametre $\alpha(t)$ a $\lambda(t)$ sú funkcie monotónne klesajúce v čase, k_i udáva počet neurónov j , pre ktoré pri danom vstupe \mathbf{x} platí: $\|\mathbf{x} - \mathbf{w}_j\| < \|\mathbf{x} - \mathbf{w}_i\|$. Takto sa v každej iterácii stanoví sekvencia neurónov i_0, i_1, \dots, i_{n-1} (podľa blízkosti k aktuálnemu vstupu) a modifikácia prvkov matice C , ktoré boli pri inicializácii nastavené na 0, sa realizuje nasledovnou trojicou krokov:

- (i) ak $C_{i_0 i_1} = 0$, spoj uzly i_0 a i_1 , t.j. prirad' $C_{i_0 i_1} > 0$
- (ii) resetuj "vek" hrany $i_0 - i_1$, t.j. $t(i_0, i_1) = 0$ a zvýš "vek" všetkých spojení s víťazom, t.j. $t(i_0, j) = t(i_0, j) + 1 \quad \forall j | C_{i_0 j} > 0$
- (iii) zruš všetky spojenia s víťazom, ktorých vek presiahol limit T , t.j.

¹² Hebbovo učiace pravidlo so súťažiením v podstate predstavuje syntézu dvoch princípov: korelačného a kompetičného.

$$\text{nastav } C_{i_0j} = 0, \forall j | (C_{i_0j} > 0) \wedge (t(i_0, j) > T)$$

Pri modifikácii spojení (hrán) $C_{i_0i_t}$ podľa vzťahu $\Delta C_{i_0i_t} \propto y_i \cdot y_j$ je výstup neurónov počítaný ako $y_i = f(\|\mathbf{x} - \mathbf{w}_i\|)$, kde f je kladná monotónne klesajúca funkcia. Uvedený algoritmus umožňuje skonštruovať graf, ktorý je schopný aproximovať i zložité štruktúry s rôznou dimenziou v jednotlivých jeho častiach.

GCS (Growing Cell Structures) [19]. Zatiaľ čo doteraz spomínané modely pracovali s vopred stanoveným, konštantným počtom neurónov, v GCS sa neuróny i pridávajú resp. uberajú. Algoritmus takto umožňuje aproximovať širšie spektrum distribúcií dát pri zachovaní lokálnej topológie, avšak limitujúcim faktorom ostáva to, že výsledný graf je monodimenzionálny, pričom topológiu grafu treba na začiatku zvoliť.¹³ Navyše, výstupná reprezentácia má tú istú dimenziu ako vstupný priestor.

Algoritmus GCS v sebe zahŕňa tri kroky: redistribúcia neurónov (uzlov), pridanie neurónu a odstránenie neurónu. Redistribúcia sa realizuje v každej iterácii, podobne ako u SOM. Rozhodnutie o následnom pridaní neurónu je založené na nasledovnej myšlienke: každý uzol i má priradenú chybovú premennú $err(i)$, ktorá sa v prípade jeho víťazstva inkrementuje o jeho vzdialenosť od aktuálneho vstupu, t.j. $err(i)+ = \|\mathbf{x} - \mathbf{w}_i\|$. Príliš vysoká hodnota $err(i)$ takto signalizuje, že v danej oblasti grafu je nízka hustota neurónov (uzlov). Preto sa nájde "čierna ovca" (bs) medzi neurónmi, t.j. ten s najväčšou hodnotou $err(i)$, a jeho najvzdialenejší bezprostredný (spojený s ním hranou) sused f . Do stredu medzi ne sa vloží nový neurón (nn): $\mathbf{w}_{nn} = (\mathbf{w}_{bs} + \mathbf{w}_f) / 2$, a vytvoria sa jeho spojenia s okolitými neurónmi tak, aby sa zachovala topológia k -simplexov. Následne je ešte potrebné adaptovať chybové premenné.

Odstránenie neurónu je potrebné najmä pri aproximovaní nespojitej distribúcie dát. Vychádza sa z úvahy, že čím dlhšie neurón nezvíťazil, tým väčšia je pravdepodobnosť, že bude odstránený.

DCS (Dynamic Cell Structures) [6]. Algoritmus DCS je zlúčením a rozšírením myšlienok obsiahnutých vo modeloch TRN a GCS. Hlavný rozdiel oproti GCS spočíva v tom, že topológia konštruovaného grafu nemá vopred danú dimenziu, čím sa dáva možnosť vzniku multidimenzionálneho grafu (t.j. s rôznou dimenziou v jeho častiach, ako na obr. 7.27). Nové neuróny sa vkladajú podobným spôsobom ako u GCS: medzi neurón i s najväčšou hodnotou $err(i)$ a jeho bezprostredného suseda j s druhou najväčšou hodnotou $err(j)$; nie však do stredu medzi ne, ale s proporcionálnym posunom podľa pomeru hodnôt ich chybových funkcií. Víťazný neurón sa adaptuje ako u SOM, neuróny s ním susediace podľa vzťahu

$$\Delta \mathbf{w}_j = \alpha(t) \cdot A_{i^*j} \cdot (\mathbf{x} - \mathbf{w}_j) \quad \forall j : A_{i^*j} > 0, j = 1, 2, \dots, n, \quad (7.34)$$

pričom A_{i^*j} sú prvky matice susednosti, ktoré sa modifikujú trochu zložitejším spôsobom ako u TRN:

$$A_{ij}(t+1) = \begin{cases} 1 & \text{ak } y_i \cdot y_j = \max\{y_k \cdot y_l\}, k, l = 1, 2, \dots, n \\ 0 & \text{ak } A_{ij}(t) < \theta \\ \varepsilon \cdot A_{ij}(t) & \text{v ostatných pr'padoch.} \end{cases} \quad (7.35)$$

Vo vzťahu (7.35) figuruje ε ako konštanta zabúdania, ktorá spĺňa podmienku $0 < \varepsilon < 1$, θ možno interpretovať ako prah eliminácie spojení, a $y_i = f(\|\mathbf{x} - \mathbf{w}_i\|)$ predstavuje výstup i -teho neurónu, kde f je kladná monotónne klesajúca funkcia.

GSOM (Growing SOM) [3]. Tento algoritmus taktiež umožňuje pridávať neuróny (podobne ako GCS a DCS), avšak s obmedzujúcou podmienkou, že štruktúra usporiadania neurónov ostáva pravidelnou – v tvare (viacrozmernej) mriežky. Z toho vyplýva nutnosť pridávať nielen jednotlivé neuróny, ale podľa potreby i celé "pásky" či "vrstvy" neurónov, ak sa algoritmus "rozhodne" zvýšiť dimenziu grafu (napr. k existujúcej reťazi neurónov možno pridať buď jeden neurón, alebo celý pás neurónov, čím sa z reťaze stane mriežka). Motiváciou pre takúto dynamiku nárastu siete je fakt, že vďaka uchovanej pravidelnosti je práca s takouto sieťou podstatne jednoduchšia ako so všeobecným grafom, a navyše GSOM redukuje dimenziu popisu dát.

¹³ Graf s k -dimenziálnou topológiou pozostáva z k -simplexov, čo je jeho základný "stavebný blok", ktorý vznikne vzájomným pospájaním $k+1$ uzlov.

Tabuľka 7.3. Charakteristiky algoritmov samoorganizácie neurónových sietí na formovanie príznakových máp zachovávajúcich topológiu. Pod "redukciou dimenzie" sa myslí zníženie dimenzie reprezentácie výstupnej informácie, "n=konšt." označuje konštantný, vopred stanovený počet neurónov v sieti. "Monodimenzionalita grafu" znamená, že graf má v celej svojej štruktúre rovnakú dimenziu k , t.j. že pozostáva len zo simplexov k -tého rádu. "Fixovanosť topológie" sa vzťahuje na apriórne určenie štruktúry grafu.

Algoritmus	Redukcia dim.	n = konšt.	Monodim. graf	Fix. topológia
SOM	áno	áno	áno	áno
VQP	áno	áno	áno	nie
TRN	nie	áno	nie	nie
GCS	nie	nie	áno	nie
DCS	nie	nie	nie	nie
GSOM	áno	nie	áno	nie

Porovnaním spomínaných algoritmov samoorganizácie z hľadiska ich charakteristických črt dospejeme k nasledovnej tabuľke (tab. 7.3).

Z pohľadu vyššie uvedených charakteristík sa najuniverzálnejším algoritmom spomedzi spomínaných javí algoritmus DCS. Umožňuje pretransformovať vstupné dáta do podoby neorientovaného grafu, ktorý svojou štruktúrou verne odpovedá dátam v každej časti vstupného priestoru. Takáto univerzálnosť výstupnej štruktúry je na jednej strane výhodou, avšak súčasne sa s tým podstatne sťažuje práca s takouto výstupnou reprezentáciou. Je to práve kvôli jej nepravidelnosti, preto je nutné spracovávať a uchovávať komplikované vzťahy medzi jednotlivými uzlami grafu. Naopak, SOM je po tejto praktickej stránke veľmi jednoduchá. To si možno uvedomiť už pri jej zobrazovaní ako pravidelnej mriežky, vo fáze programovania algoritmu učenia, či v následnej implementácii. GSOM predstavuje kompromis – zachováva si jednoduchosť výstupnej reprezentácie SOM (samotný algoritmus je však pochopiteľne zložitejší), a súčasne je pružnejšia pri aproximácii štruktúry dát.

Literatúra

- [1] S.C. Ahalt et al. Competitive learning algorithms for vector quantization. *Neural Networks*, 3(3):277-290, 1990.
- [2] H.-U. Bauer and K.R. Pawelzik. Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE Transactions on Neural Networks*, 3(4):570-579, 1992.
- [3] H.-U. Bauer and T. Villmann. Growing a hypercubical output space in a self-organizing feature map. *Technical report TR-95-030*, ICSI Berkeley, California, 1995.
- [4] C. Bouton et al. Self-organization and convergence of the Kohonen algorithm. In: *Probabilités Numériques* (Eds. N. Bouleau and D. Talay), INRIA, Paris, France, 163-180, 1991.
- [5] W.D. Brandt, H. Behme and H.W. Strube. Bildung von Merkmalen zur Spracherkennung mittels phonotopischer Karten. In: *Fortschritte der Akustik-DAGA 91*, Bad Honnef, Germany, 1057-1060, 1991.
- [6] J. Bruske and G. Sommer. Dynamic cell structures learns perfectly topology preserving map. *Neural Computation*, 7:845-865, 1995.
- [7] M. Budinich and J.G. Taylor. On the ordering conditions for self-organizing maps. In: Proc. of ICANN'94 (Eds. M. Marinaro and P.G. Morasso), Springer-Verlag, London, UK, I. 347-349, 1994.
- [8] M. Cottrell, J.C. Fort and G. Pages. Two or three things that we know about the Kohonen algorithm. Technical report No.31, Université Paris 1, France, 1994.
- [9] D.A. Critchley. Stable states, transitions and convergence in Kohonen self-organizing maps. In: *Proc. of ICANN'92*, (Eds. I. Alexander and J. Taylor), North-Holland, Brighton, UK, 281-284, 1992.
- [10] P. Demartines. *Analyse de données par réseaux de neurones auto-organisés*. Doktorská dizertačná práca, L'Institut National Polytechnique de Grenoble, France, 1994.
- [11] P. Demartines. Organization measures and representations of Kohonen maps. In: *First IFIP Working Group 10.6 Workshop* (Ed. J. Héroult), 1992.
- [12] P. Demartines and F. Blayo. Kohonen self-organizing maps: Is the normalization necessary? *Complex Systems*, 6:105-123, 1992.
- [13] P. Demartines and J. Héroult. Representation of nonlinear data structures through fast VQP neural network. In: *Neuronimes*, 411-424, 1993.
- [14] E. Erwin, K. Obermayer and K. Schulten. Convergence properties of self-organizing maps. In: *Artificial Neural Networks*, (Eds. T. Kohonen et al), Elsevier, Amsterdam, Netherlands, 409-414, 1991.

- [15] E. Erwin, K. Obermayer and K. Schulten. Self-organizing maps: Ordering, converge properties and energy functions. *Biological Cybernetics*, 67(1):47-55, 1992.
- [16] E. Erwin, K. Obermayer and K. Schulten. Self-organizing maps: Stationary states, metastability and convergence rate. *Biological Cybernetics*, 67(1):35-45, 1992.
- [17] I. Farkaš. *On Vector-coded Feature Mapping Using Self-organizing Neural Maps*. Doktorská dizertačná práca, Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave, 1995.
- [18] E.A. Ferrán. An ordering theorem that allows for ordering changes. In: *Artificial Neural Networks 2*, (Eds. I. Alexander and J. Taylor), North-Holland, Amsterdam, Netherlands, I:165-168, 1992.
- [19] B. Fritzke. Growing cell structures — a self organizing network for unsupervised and supervised training. *Neural Networks*, 7(3):1441-1459, 1994.
- [20] G.J. Goodhill, S. Finch and T.J. Sejnowski. Quantifying neighbourhood preservation in topographic mappings. *Technical report INC-9505*, Institute for Neural Computation, La Jolla, CA, 1995.
- [21] R.M. Gray. Vector quantization. *IEEE ASSP Magazine*, 1:4-29, April 1984.
- [22] J. Hertz, A. Krogh and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, 1991.
- [23] E.I. Knudsen, S. du Lac and S.D. Esterly. Computational maps in the brain. *Annual Review of Neuroscience*, 10:41-65, 1987.
- [24] T. Kohonen. Analysis of a simple self-organizing proces. *Biological Cybernetics*, 44(2):135-140, 1982.
- [25] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, 1988.
- [26] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):56-69, 1982.
- [27] T. Kohonen. Self-organizing maps: Optimization approaches. In: *Artificial Neural Networks* (Eds. T.Kohonen et al.), North-Holland, Amsterdam, Netherlands, II:981-990, 1991.
- [28] T. Kohonen. Speech recognition based on topology-preserving neural maps. In: *Neural Computing Architectures*, (Ed. I. Alexander), North Oxford Academic Publishers, 26-40, 1989.
- [29] T. Kohonen. Statistical pattern recognition revisited. In: *Advanced Neural Computers* (Ed. R. Eckmiller), Elsevier Science Publ. B.V., North-Holland, 137-144, 1990.
- [30] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78:1464-1480, 1990.
- [31] Z.-P. Lo, Y. Tu and B. Bavarian. Analysis of the convergence properties of topology preserving neural networks. *IEEE Transactions on Neural Networks*, 4(2):207-220, 1993.
- [32] C. von der Malsburg. Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14:85-100, 1973.
- [33] J. Mao and A.K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6(2):296-317, 1995.
- [34] T. Martinetz and K. Schulten. "Neural gas" network learns topologies. In: *Artificial Neural Networks*, vol.I, (Eds. T.Kohonen et al.), North-Holland, 397-402, 1991.
- [35] T. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, 7(3):505-522, 1994.
- [36] G. Pages. Voronoi tessellation, space quantization algorithms and numerical integration. *Proc. of ESANN'93*, D facto conf. services, Brussels, Belgium, 221-228, 1993.
- [37] H. Ritter, T. Martinetz and K. Schulten. *Neural Computation and Self-organizing Maps: An Introduction*, Addison-Wesley, 1992.
- [38] H. Ritter and K. Schulten. Kohonen self-organizing maps: Exploring their computational capabilities. *Proc. IEEE ICNN*, San Diego, 109-116, 1988.
- [39] H. Ritter and K. Schulten. On the stationary state of Kohonen's self-organizing sensory mapping. *Biological Cybernetics*, 54(2):99-106, 1986.
- [40] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61(4):241-254, 1989.
- [41] P. Růžička. On convergence of learning results for topological maps. *Neural Network World*, 4:413-424, 1993.
- [42] W. Siedlecki, K. Siedlecka and J. Sklansky. Mapping techniques for exploratory pattern analysis. In: *Pattern Recognition and Artificial Intelligence*, (Eds. E.S. Gelsema and L.N. Kamnal), North-Holland, Amsterdam, Netherlands, 277-299, 1988.
- [43] A. Takeuchi and S. Amari. Formation of topographic maps and columnar microstructures in nerve fields. *Biological Cybernetics*, 35:63-72, 1979.
- [44] J.T. Tou and R.C. Gonzalez. *Pattern Recognition Principles*. Addison-Wesley, 1974.
- [45] J.A. Walter and K. J. Schulten. Implementation of self-organizing neural networks for visuo-motor control of an industrial robot. *IEEE Transactions on Neural Networks*, 4(1):86-95, 1993.
- [46] D.J. Willshaw and C. von der Malsburg. How patterned neural connections can be set up by self-organization. *Proc. of the Royal Society of London B*, 194:431-445, 1976.
- [47] H. Yin and N.M. Allinson. On the distribution and convergence of feature space in self-organizing maps. *Neural Computation*, 7:1178-1187, 1995.