

7. Samoorganizujúce sa mapy

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 [30]. Ako vyplýva už z názvu, SOM patrí do kategórie modelov, ktoré sa učia *bez učiteľa* (samoorganizovane, angl. *unsupervised learning*), t.j. algoritmus učenia nemá informáciu o požadovaných aktivitách výstupných neurónov v priebehu tréningu (ako napr. v algoritme back-propagation, kapitola 5), 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 *zobraziť* 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).

Fenoménu topologického zobrazenia príznakov (angl. *feature mapping*) má výrazné zastúpenie v biologických neurónových sieťach, konkrétne v mozgoch vyšších cicavcov i človeka [23]. *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 počiatkových 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 podnetov prichádzajúcich z okolitého prostredia, a teda z pohľadu spôsobu učenia prebieha samoorganizovane.

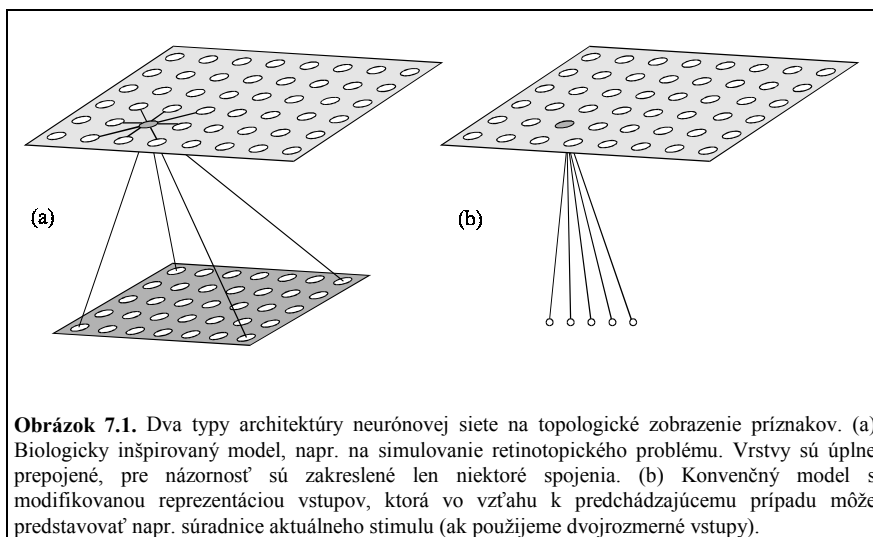
Uvedené fakty boli inšpiračným zdrojom pre snahu simulovať proces samoorganizácie pomocou výpočtového modelu. Navyše, ako sa neskôr ukázalo, model SOM vďaka svojej relatívnej jednoduchosti a ilustratívnosti našiel uplatnenie pri riešení rôznych praktických problémov, kde sa využíva topologické zobrazenie, 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. Pomerne rozsiahly zoznam aplikácií možno nájsť v [30].

7.1.1 Prvé biologicky inšpirované modely

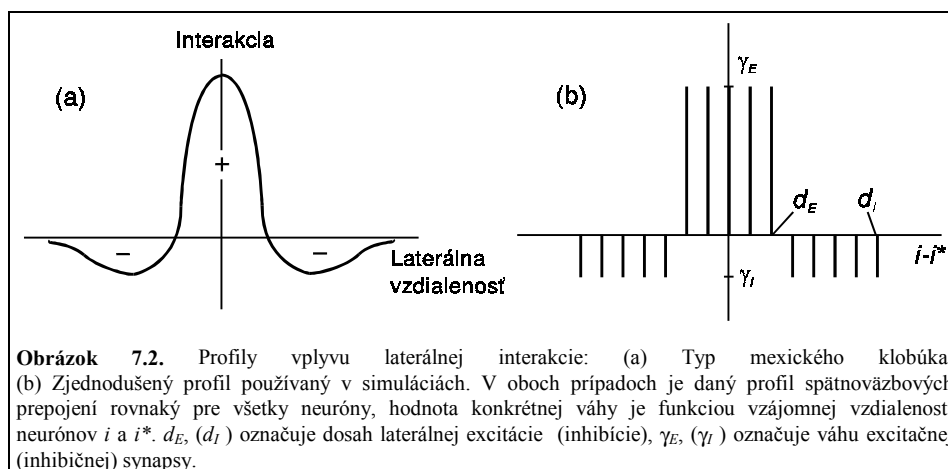
Jednými z prvých, ktorí sa zaoberali problémom topologického mapovania pomocou neurónových sietí, boli Willshaw a von der Malsburg [46]. V snahe porozumieť biologicky zaujímavému problému — mechanizmu projekcie zo sietnice na mozgovú kôru (retinotopický problém) — navrhli model neurónovej siete s architektúrou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

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. 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 pravidle¹ s následným normovaním váh. Výsledkom tréningu na dipólových stimuloch bolo získanie projekcie typu pozícia verzus pozícia, s dodržanou vlastnosťou topologického usporiadania odoziev vo výstupnej vrstve.

¹ Hebbovo pravidlo je založené na princípe zvyšovania hodnoty synaptických váh medzi neurónmi, ktoré sú synchronne aktívne, t.j. $\Delta w_{ij} \propto y_i y_j$ (v spomínanom modeli namiesto vstupného neurónu figuruje receptor). Adaptujú sa všetky spojenia, pričom miera adaptácie teda závisí od korelácie medzi výstupmi oboch neurónov.



Treba spomenúť, že očakávaný efekt topologického usporiadania by sa nebol dostavil, keby

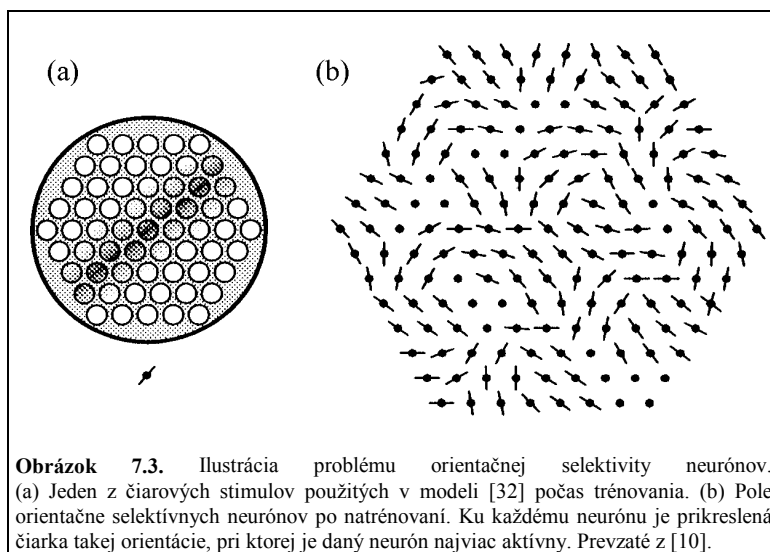


bol len jeden receptor naraz aktívny. Simultánna aktivita minimálne dvoch susedných receptorov predstavuje redundanciu a umožňuje sieti zachytiť korelačné, priestorové vzťahy na vstupe, čo je nutnou podmienkou pre dosiahnutie správnej samoorganizácie.

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 verzus 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 y označuje distribúciu aktivít jednotlivých neurónov vo výstupnej vrstve (reťazi) a zložky vektora net označujú analógie postsynaptických potenciálov. Dynamiku takejto siete možno popísať rovnicou

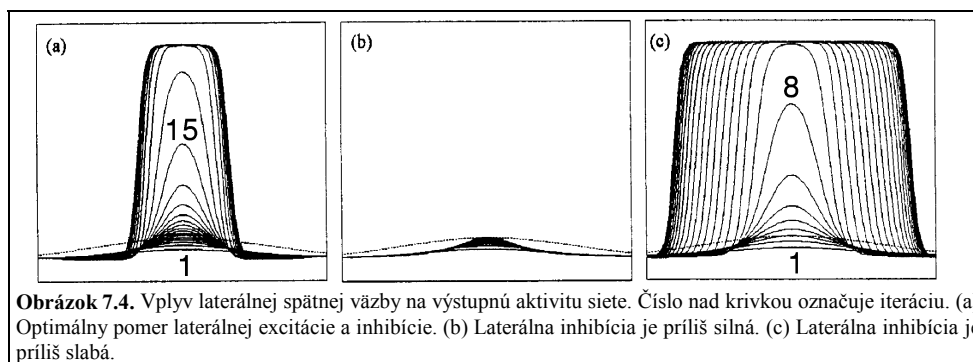
$$y(t+1) = \mathcal{S}[net] = \mathcal{S}[z + L \cdot y(t)] \quad (7.1)$$

Na vstup sa teda privedie lokálny stimul x (aktivita receptorového poľa), vypočíta sa počiatočná vnútorná aktivita výstupných neurónov $z = W \cdot x$, kde W je matica váh medzi receptormi a neurónmi, a laterálna spätná väzba sa nechá niekoľko iterácií pôsobiť. Je daná symetrickou maticou L , koeficienty ktorej možno jednoducho získať z profilu na obr. 7.2b.² Vektorová funkcia \mathcal{S} , pozostávajúca zo sigmoid, zabraňuje nekonečnému nárastu výstupnej aktivity.

² 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.

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 x gaussovského tvaru (znázornená čiarkovane) so zložkami generovanými podľa vzťahu $x(i) = 1 \cdot \exp(-(i-50)^2 / 2.30^2)$, pre jednoduchosť $i=1,2,\dots,100$, t.j. dimenzia vstupu rovná počtu výstupných neurónov. Výstup y každého neurónu bol ohraničený 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 počítačová výstupná aktivita neurónov maximum. Platí, že veľkosť aktivačnej bubliny ("zhľuku" 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 nejestvujú, 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. $i^* = \arg \max(\mathbf{w}_i^T \mathbf{x})$, 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:

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

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

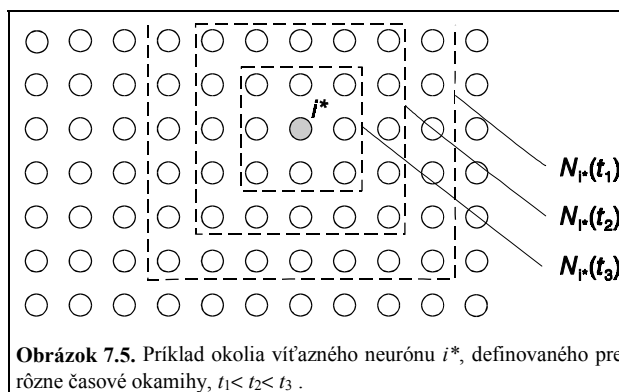
$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t) \cdot h(i^*, i) \cdot [\mathbf{x}(t) - \mathbf{w}_i(t)] \quad (7.4)$$

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 diskrétno 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 texte, kontrakciu váhových vektorov na okrajoch mapy.



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

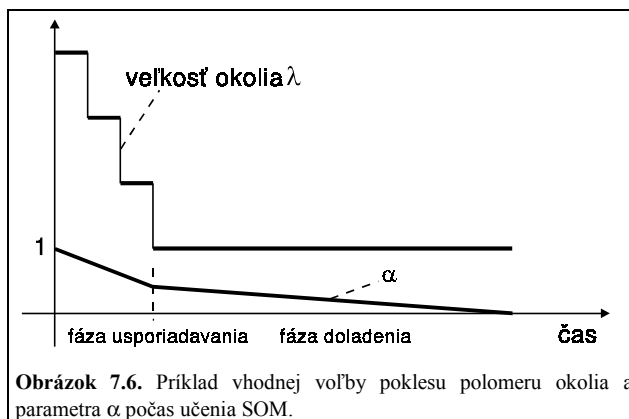
$$h(i^*, i) = \exp\left(-\frac{d_E^2(i^*, i)}{\lambda^2(t)}\right) \quad (7.6)$$

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 trénovacej 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),

1) 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 diskretné 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.

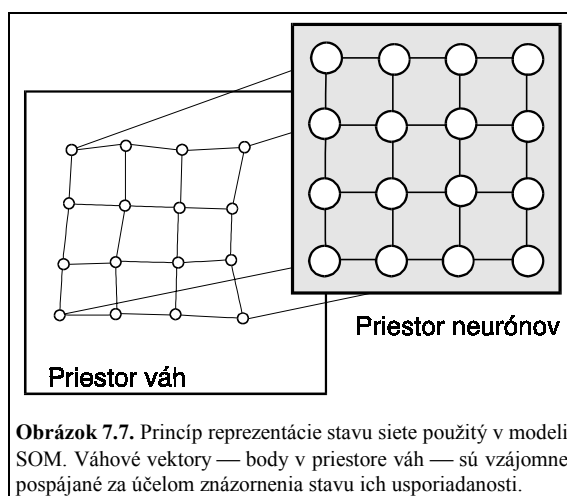
Na presnom počte iterácií takisto nezáleží. Kohonen uvádza empiricky získanú pomôcku, podľa ktorej počet iterácií má byť minimálne 500-násobok počtu neurónov v sieti. Bežne sa počet iterácií pohybuje v rozmedzí rádovo 10000-100000. Dôležité je, aby počas fázy usporiadavania, pokiaľ je ešte parameter 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. $\xi: \mathbf{X} \subset \mathfrak{R}^N \rightarrow \mathfrak{S} = \{i; 1, 2, \dots, n\}$, kde N je dimenzia vstupov, a n počet neurónov v sieti³. Takto máme do činenia s dvoma priestormi —

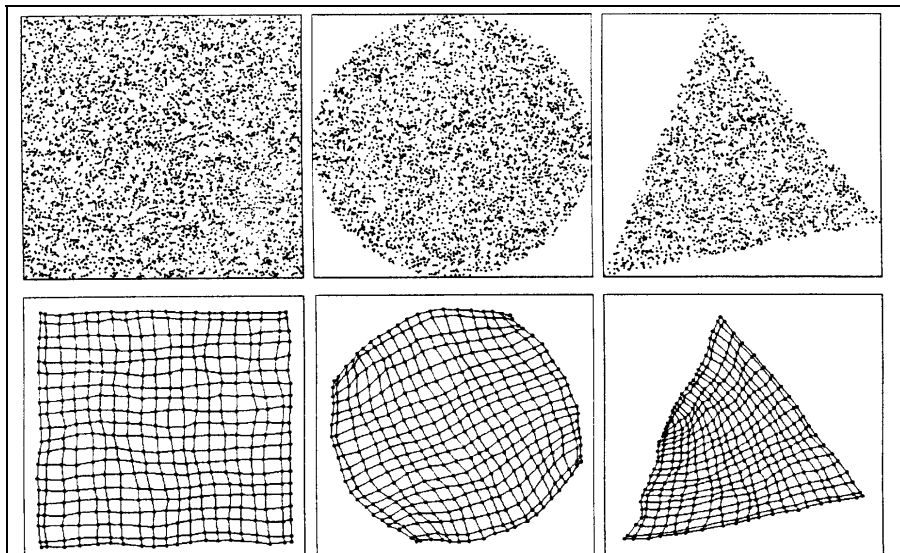
³ Ako vidieť, v modeli SOM je výstupná informácia reprezentovaná inak ako v ostatných modeloch ako napr. viacvrstvový perceptrón, či nejaká lineárna dopredná sieť. Namiesto výstupnej aktivity siete tu nás zaujíma len pozícia víťazného neurónu. V snahe nazrieť na SOM ako na systém realizujúci transformáciu $\varphi: \mathbf{X} \rightarrow \mathbf{Y}$ i tu je

priestorom vstupov 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.

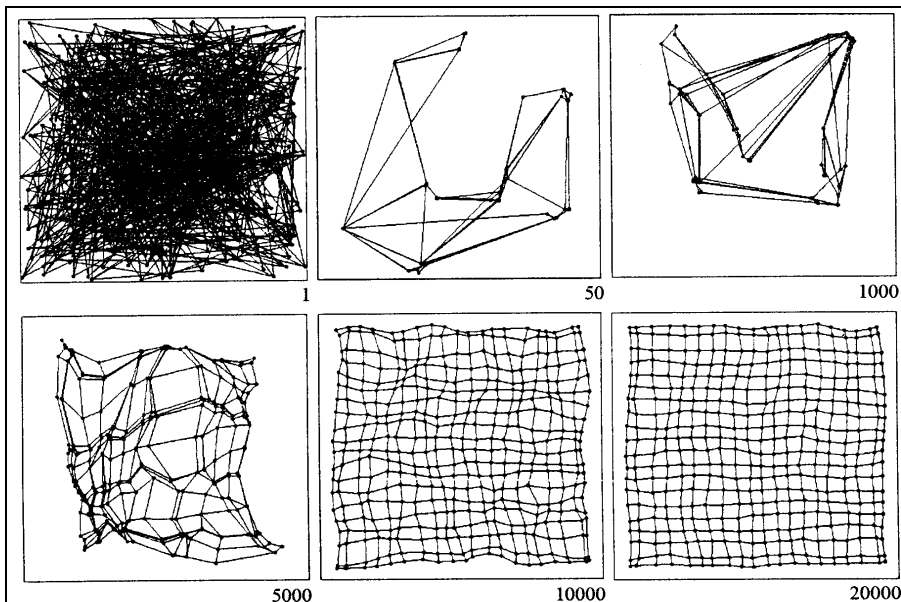


Spájajúce čiary medzi váhovými vektormi znázorňujú skutočnosť, že každé dva spojené váhové vektory prináležia dvojici neurónov, ktoré sú bezprostrednými susedmi v mriežke. Pri pohľade na zobrazené váhy takto dostávame vizuálnu informáciu o ich rozmiestnení i vzájomnom usporiadaní. Podobne vo výstupnom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

možné vypočítava 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 vektora \mathbf{w}_i) je výpočtovo náročnejší, ale biologicky prijateľnejší [17].



Obrázok 7.8. Príklady jednoduchých množín dát s rovnomerným rozdelením v rovine (horná trojica obrázkov) a stavy SOM (dolná trojica obrázkov) zobrazené v priestore váh po natrénovaní (20000 iterácií). Ako vidieť, SOM aproximuje funkciu hustoty vstupných dát.

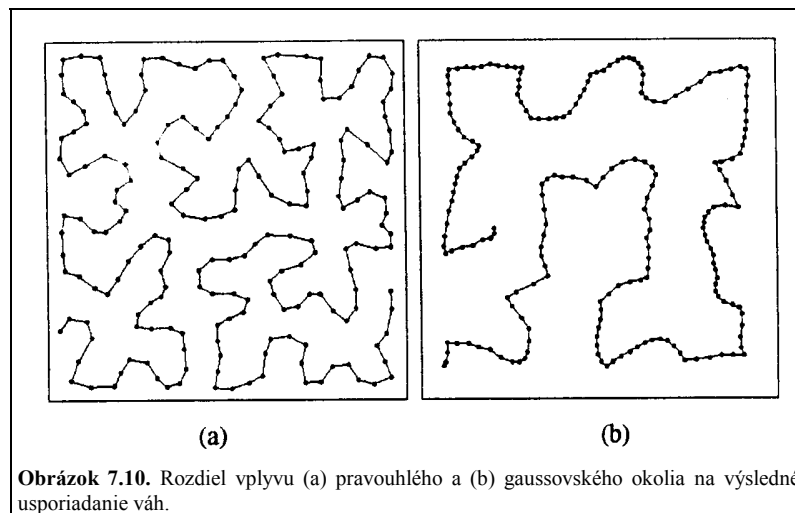


Obrázok 7.9. Usporiadavanie váhových vektorov počas tréningu na dátach s rovnomerným rozdelením. Na prvých troch záberoch vidieť, že SOM sa nachádza vo fáze usporiadavania, zatiaľ čo ostatné tri zábery už odpovedajú fáze dolad'ovania.

Ak budeme pre začiatok uvažovať, že máme dvojrozmerné vstupy $\mathbf{x} = [x_1, x_2]^T$, môžeme priestor váh priamo zobrazit' 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 tréovania jednotlivé vstupné vzory vyberané náhodne z tréovacej množiny. Je vidieť, že váhy majú tendenciu aproximovať funkciu hustoty tréovacej množiny.

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

Ďalším zaujímavým fenoménom, ktorý je možné u SOM ilustrovať, je rozdielny vplyv pravouhlého (rovnicu 7.5) a gaussovského (7.6) okolia, parametrizovaných hodnotou λ .



Obrázok 7.10. Rozdiel vplyvu (a) pravouhlého a (b) gaussovského okolia na výsledné usporiadanie váh.

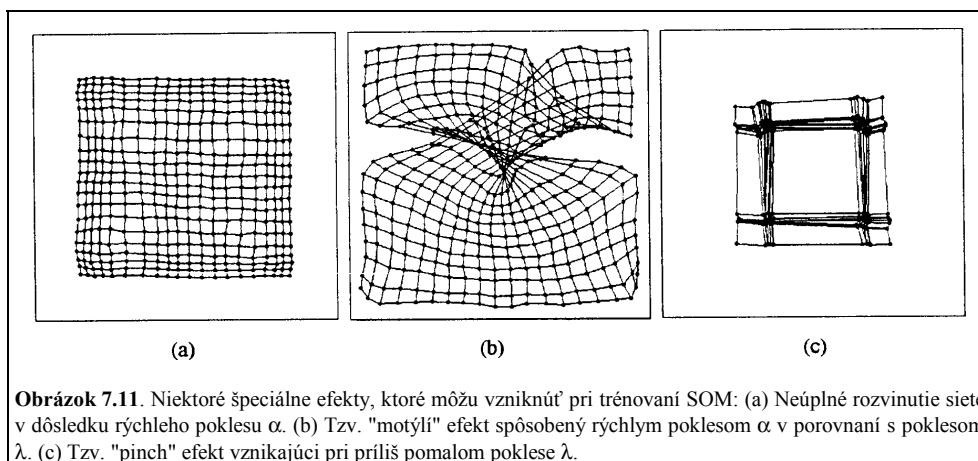
Väčší rozsah gaussovského okolia má za následok vznik "stiahnutejšej" reprezentácie pri rovnakom počte neurónov (obr. 7.10).

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 nežiaducich 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)⁴, ktorý

⁴ Použitý termín nemá nijaký súvis s rovnomenným termínom používaným v teórii chaosu.



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).

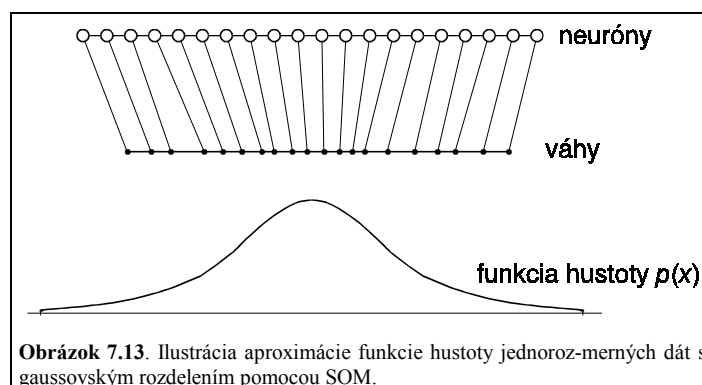
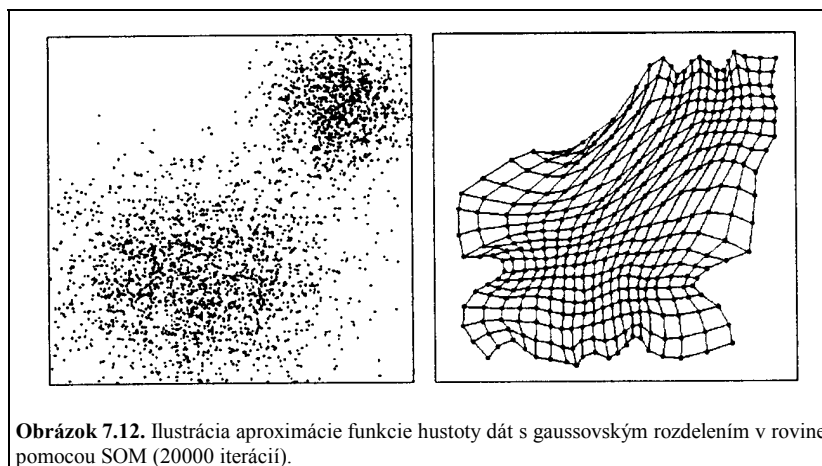
7.3.2 Hraničný efekt

Pri pohľade na výsledné stavy siete 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 nežiaduci, možno ho odstrániť tak, že sa okolie algoritmicke uzavrie do slučky. Potom nastane situácia, že všetky neuróny budú pozične ekvivalentné: v prípade reťaze bude mať *každý* neurón dvoch bezprostredných 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 "trénovaciu 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.

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.⁵

7.4 Teoretická analýza algoritmu SOM

Kohonenov algoritmus v sebe skrýva mnoho otázok, na ktoré je vo všeobecnosti ťažké nájsť odpoveď. Problémy ako: Konverguje algoritmus vždy? Nevystupuje tu problém lokálnych miním? Koľko iterácií treba na konvergenciu? Existujú optimálne hodnoty pre voľbu parametra α a funkcie okolia $h(i, r^*)$? atď., sú analyzovateľné len v špeciálnych, jednoduchších prípadoch, a preto sa väčšina doterajších teoretických prác o SOM zameriavala na jednorozmerný prípad, t.j. jednorozmernú SOM s jednorozmernými vstupmi. V zložitejších prípadoch je situácia sťažená skutočnosťou, že [8] (a) sa nepodarilo definovať, čo je to korektné usporiadaná konfigurácia váh v prípade dimenzie vyššej ako jedna, a (b) Erwin a spol. dokázali, že v prípade spojitých dát nemožno vo všeobecnosti nájsť globálnu kritériálnu (účelovú) funkciu, z ktorej by sa dal odvodiť Kohonenov algoritmus [15].

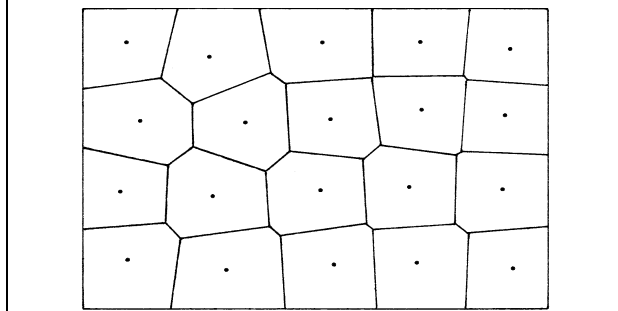
I v jednoduchších prípadoch vystupuje niekoľko matematických problémov, ktoré súvisia s procesom samoorganizácie. Sú to: (a) vlastnosť *vektorovej kvantizácie*, (b) hľadanie *kritériálnej funkcie* minimalizovanej algoritmom SOM, (c) *usporiadanie váh* a (d) *konvergencia váh* SOM. O každom probléme sa stručne zmienime v ďalšom texte.

7.4.1 Vektorová kvantizácia

Ak si na moment odmyslíme špecifickú črtu SOM — usporiadanie neurónov v pravidelnej štruktúre a s tým súvisiacu schopnosť zachovania topológie — môžeme sa na SOM pozrieť ako na vektorový kvantifikátor.

Pri probléme vektorovej kvantifikácie [21] je úlohou nahradiť danú množinu alebo distribúciu vstupných dát X početne menšou množinou referenčných vektorov, nazývaných *prototypy*. Účel takejto náhrady sa uplatňuje napr. pri prenose údajov v telekomunikáciách či v kompresii dát (obrazových, rečových), keď sa dosiahne výrazné zníženie objemu dát pri minimálnom poklese kvality. Vďaka vektorovej kvantizácii stačí namiesto celej množiny X preniesť (či uchovať) len množinu prototypov spolu s informáciou (index prototypu) o príslušnosti každého vstupného vektora ku tomu-ktorému prototypu (na základe podobnosti v zmysle euklidovskej vzdialenosti). Vektorovou kvantizáciou sa priestor X rozdelí na disjunktné oblasti, ktoré tvoria tzv. *Voronoiho mozaiky* (obr. 7.14).

⁵ 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.



Obrázok 7.14. Voronoiho mozaiky. Priestor je rozdelený na polyedrálne oblasti určujúce "sféry vplyvu" jednotlivých prototypov (ozn. ako body) podľa minima euklidovskej vzdialenosti. Hranice medzi oblasťami sú tvorené hyperrovinami (na obr. úsečkami), ktoré prechádzajú stredmi spojnic susedných prototypov a sú na ne kolmé.

Kritériom vektorovej kvantizácie je nájdenie optimálnych pozícií týchto prototypov $\{\mathbf{w}_i : i = 1, 2, \dots, n\}$ (počet ktorých sa obyčajne vopred stanoví), a to tak, aby sa minimalizovala stredná kvadratická odchýlka medzi vstupom \mathbf{x} a jeho prototypom \mathbf{w}_c , pričom $c = \arg \min \|\mathbf{x} - \mathbf{w}_i\|$. V prípade spojitého rozdelenia dát daného funkciou hustoty $p(\mathbf{x})$ udáva strednú kvadratickú odchýlku (nazývanú tiež chybou rekonštrukcie) vzťah

$$E = \int_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \mathbf{w}_c\|^2 \cdot p(\mathbf{x}) d\mathbf{x} \quad (7.7)$$

V diskretnom prípade, ak vstupná množina pozostáva z M prvkov, má chyba rekonštrukcie tvar

$$E = \frac{1}{M} \sum_j \|\mathbf{x}_j - \mathbf{w}_c\|^2 \quad (7.8)$$

Pri uvažovaní, že prototypy odpovedajú váhovým vektorom, predstavuje algoritmus SOM štandardný vektorový kvantifikátor [36], ktorý minimalizuje funkcionál (7.7), resp. (7.8). Súčasne je však podmienkou $\lambda = 0$, t.j. v každej iterácii sa adaptuje len váhový vektor víťazného neurónu (učenie typu "vít'az berie všetko", angl. *winner-takes-all*).

Za účelom vektorovej kvantizácie boli použité i zložitejšie algoritmy so súťažením, pracujúce na princípe "vít'az berie väčšinu" (angl. *winner-takes-most*) [1], jedným z ktorých je aj Kohonenov algoritmus. Tieto zložitejšie algoritmy sú vďaka kooperácii medzi neurónmi schopné eliminovať nežiaduce efekty vznikajúce pri "winner-takes-all" prístupe, ako napr. existencia "mŕtvych" neurónov (t.j. takých, ktoré sú zo súťaže vylúčené a prestanú sa učiť). Súčasne však väčšia zložitosť týchto algoritmov má za následok, že sú vo všeobecnosti prítvrdým orieškom pre teoretickú analýzu.

7.4.2 Kriteiálne funkcie

Ako v iných algoritmoch, i pri SOM vyplýva snaha nájsť kriteiálnu funkciu minimalizovanú algoritmom adaptácie synaptických váh z toho, že taká funkcia by umožnila zjednodušiť teoretickú analýzu. V prípade Kohonenovho algoritmu bolo dokázané, že vo všeobecnosti pri spojitnej množine X taká globálna kriteiálna funkcia neexistuje [15]. Väčšina prác sa preto zameriava na jednorozmerný prípad.

V špeciálnom prípade s nulovým okolím bolo ukázané, že SOM funguje ako vektorový kvantifikátor, ktorý v spojitom prípade minimalizuje funkcionál (7.7), resp. v diskretnom prípade (7.8). Pri nenulovom okolí je situácia zložitejšia. Kohonen [27] analyzoval spojitý prípad, pre ktorý navrhol funkcionál

$$E(w) = \sum_i \int_{x \in \Omega_i} \sum_k h(i, k) \|x - w_k\|^2 p(x) dx \quad (7.9)$$

Pri odvodzovaní učiaceho pravidla (vzťah 7.4) ukázal, že vypočítaný gradient ∇E pozostáva z dvoch členov: prvý člen predstavuje zmenu váh vnútri Voronoiho buniek Ω_i , druhý vyjadruje posun hraníc medzi nimi.

Erwin a spol. [15] navrhli namiesto jednej kriteiálnej funkcie systém takýchto funkcií — pre každý neurón jednu. Odvodzovanie adaptačného pravidla potom spočívalo v minimalizácii každej z týchto funkcií zvlášť. Podobne ako u Kohonena [27], i tu pozostávala každá kriteiálna funkcia z dvoch členov: prvý predstavoval chybu danú pozíciou váh, druhý vyjadroval zmenu hodnoty sumárnej kriteiálnej funkcie pri posune hraníc medzi Voronoiho bunkami. Súčasne bolo pozorované, že v prvej fáze učenia, keď váhy ešte nie sú usporiadané, má väčšiu hodnotu druhý člen, zatiaľ čo vo fáze doladenia zase dominuje prvý člen.

Kohonen [27] študoval i diskretný prípad (množina vstupov x_1, x_2, \dots, x_N), v ktorom navrhol kriteiálnu funkciu tvaru

$$E(w) = \frac{1}{N} \sum_{j=1}^N \sum_i h(i^*, i) \|x_j - w_i\|^2 \quad (7.10)$$

kde N udáva počet vstupov. Minimalizáciou funkcie (7.10) dospel k štandardnému tvaru učiaceho pravidla pre SOM (vzťah 7.4).

O niečo skôr sa podobným prístupom zaoberali Ritter a spol. [38], ktorí navrhli funkcionál

$$E(w) = \frac{1}{2} \sum_{j^*i} h(i^*, i) \sum_{x_j \in \Omega_{i^*}} p_j (x_j - w_i)^2 \quad (7.11)$$

kde p_j je pravdepodobnosť, že zvolený vstup bude mať hodnotu x_j . Ukázali, že učiace pravidlo pre SOM minimalizuje tento funkcionál.

Snáď najúplnejšie výsledky možno nájsť v práci Růžičku [41], kde bolo dokázané, že pre sieť pozostávajúcu z n neurónov, s N vstupmi s rovnomerným rozdelením možno odvodiť Kohonenov algoritmus minimalizáciou funkcionálu

$$E(w) = \frac{1}{N} \sum_{i=1}^n \sum_{x_j \in \Omega_i} \sum_{m=1}^n h(i, m) \|x_j - w_m\|^2 \quad (7.12)$$

7.4.3 Usporiadavanie váh

Dôkazy usporiadavania váh (samoorganizácie) existujú pre jednorozmerný prípad, pre ktorý je ľahké definovať stav usporiadania. Správne usporiadanie predpokladá, že v reťazi neexistujú inverzie (nesprávne zoradené trojice susedných váh), t.j. váhy spĺňajú jednu z podmienok $w_1 < w_2 < \dots < w_n$ (vzostupné usporiadanie) alebo $w_n < w_{n-1} < \dots < w_1$ (zostupné usporiadanie).

Jednou z možností je definovať nejaký parameter, ktorý kvantifikuje mieru usporiadania. Kohonen [25] definoval pre reťaz n neurónov a prípad jednorozmerných vstupov tzv. index usporiadania

$$D = \sum_{i=2}^n |w_i - w_{i-1}| - |w_n - w_1| \quad (7.13)$$

a dokázal, že pri náhodnom výbere vstupov x z uvažovanej množiny tento index s pravdepodobnosťou jedna konverguje k nule (takmer istá konvergencia). Obdobný prístup je použitý v práci [18].

Stanovenie podmienky pre pokles indexu D v prípade jednorozmernej SOM je značne zjednodušené v práci [7], pričom odvodená podmienka je aplikovateľná aj pre viacrozmerné vstupy. Navyše, ako zdôrazňujú autori, ich prístup vnáša geometrický vzhľad do problému usporiadavania váh a intuitívne vysvetľuje charakter problému, ktorý vzniká pri snahe zovšeobecniť závery pre viacrozmernú SOM.

Pri alternatívnom prístupe sa usporiadavanie váh chápe ako Markovov náhodný proces, pričom konfigurácia váh predstavuje stav procesu. Cieľom je nájsť tzv. absorbčnú množinu (ktorá predstavuje usporiadanú konfiguráciu váh) a dokázať, že existuje sekvencia vstupov (vybraných náhodne, s nenulovou pravdepodobnosťou), ktorá s pravdepodobnosťou jedna zabezpečí vniknutie do tejto množiny v konečnom čase. Dôkazy sú uvedené napr. v [4, 15].

7.4.4 Konvergencia váh

Problém konvergence súvisí s usporiadaním váh, avšak líši sa v tom, že stav po skonvergovaní nemusí byť automaticky usporiadaným stavom (napr. v prípade lokálnych minim kriteriálnej funkcie, ktoré existujú pri nevhodnej funkcii okolia [16]).

Jednorozmerný prípad bol analyzovaný v prácach [24, 25]. Vychádzajúc z predpokladov, že vstupné dáta majú konštantnú funkciu hustoty, že konštantné okolie zahŕňa len najbližších susedov a že počiatočný stav je usporiadaným stavom, Kohonen ukázal, že zo všetkých takýchto počiatočných stavov konvergujú váhy s pravdepodobnosťou jedna do toho istého konečného stavu. Pri tých istých podmienkach dospeli k podobnému záveru i Bouton a spol. [4]. Všeobecnejšia funkcia okolia bola aplikovaná v práci [31], pričom výsledky boli rozšírené i pre dvojrozmerný prípad (mapa s dvojrozmernými vstupmi).

V prípade nerovnomernej distribúcie dát zohráva úlohu lokálny magnifikačný faktor $M(x)$, ktorý kvantifikuje, ako váhy aproximujú funkciu hustoty vstupných dát $P(x)$. Výsledkom analýzy takéhoto všeobecného prípadu [39] je vzťah

$$M(x) \propto P(x)^{2/3} \quad (7.14)$$

ktorý možno interpretovať tak, že SOM má tendenciu podhustovať váhami oblasti s vysokou pravdepodobnosťou výskytu a prehustovať "riedke" oblasti. Treba pripomenúť, že odvodený vzťah sa vzťahuje len na dva špeciálne prípady: jednorozmerný prípad (jednorozmerná SOM, skalárne vstupy) a dvojrozmerný prípad (dvojrozmerná SOM, dvojrozmerné vstupy), v ktorom však funkcia hustoty vstupov sa dá vyjadriť ako súčin dvoch jednorozmerných funkcií hustôt.

V práci [38] bola študovaná konvergencia viacrozmernej SOM a bola zovšeobecnená na prípady, keď sa dimenzia vstupu nezhoduje s dimenziou siete. Ritter a Schulten odvodili nutné a postačujúce podmienky pre parameter učenia α , ktoré zaručujú konvergenciu do ustáleného stavu (t je diskretný čas):

$$\sum_{t=0}^{\infty} \alpha(t) = \infty \quad (7.15)$$

$$\lim_{t \rightarrow \infty} \alpha(t) = 0 \quad (7.16)$$

V práci bola študovaná i stabilita takéhoto stavu v procese zmeny distribúcie vstupných dát. Po natrénovaní mapy na dátach s dominantnou disperziou v dvoch smeroch (pozdĺž ktorých bola mapa rozvinutá), autori sledovali vplyv fluktuácií pri zväčšovaní disperzie v treťom smere, kolmom na dominantné dva. Na tomto príklade ilustrovali tendenciu SOM reorganizovať sa v snahe aproximovať novú funkciu hustoty, ak bola prekročená nejaká kritická hodnota disperzie v treťom smere. Tento fenomén označil Kohonen ako *automatický výber dimenzií príznakov* [25] (pozri podkapitulu 7.6.1).

Doteraz najuniverzálnejší výsledok zrejme poskytuje práca Yina a Allinsona [47], ktorí pri analýze uvažovali všeobecný prípad viacrozmernej SOM s viacrozmernými vstupmi. Funkcia okolia bola taktiež všeobecná — ľubovoľná monotónne klesajúca funkcia centrovaná okolo víťazného neurónu. Autori pre takýto všeobecný prípad formálne dokázali, že hodnoty váh SOM konvergujú do pozícií, ktoré určujú finálny efekt aproximácie funkcie hustoty vstupných dát.

V niektorých prácach sa autori venovali problému, ako optimálne modifikovať veľkosť okolia λ a parametra učenia α tak, aby sa urýchlila konvergencia procesu [9, 14]. Navrhnuté

riešenia porovnávali so štandardným prístupom (t.j. s vopred zvolenými funkciami okolia h i α a ich *ad hoc* aktualizáciou počas učenia), pričom výsledky boli lepšie v nimi navrhnutých riešeniach.

7.5 DP verzia Kohonenovho algoritmu

Ako bolo spomenuté v podkapitole 7.2.1, víťazný neurón možno hľadať i na základe inej miery podobnosti, a to skalárneho súčinu

$$d_i = \mathbf{w}_i^T \cdot \mathbf{x} = \|\mathbf{w}_i\| \cdot \|\mathbf{x}\| \cdot \cos \varphi_i \quad (7.17)$$

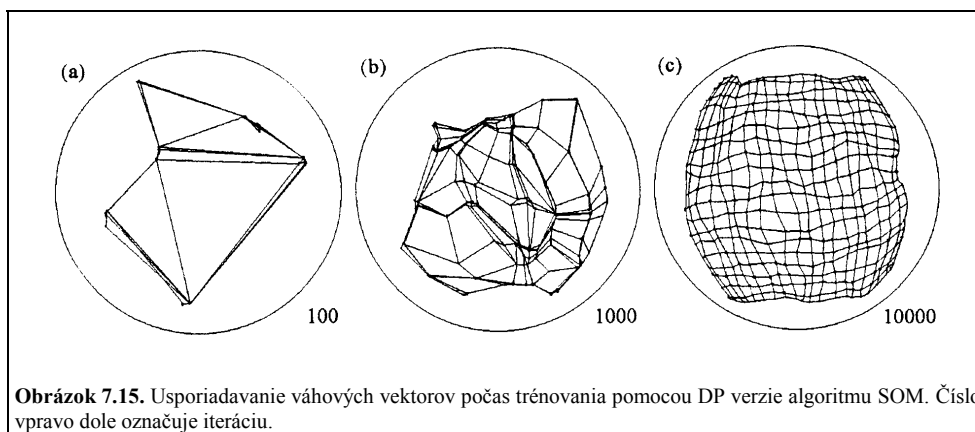
Takto dostávame DP (angl. *Dot Product*) verziu algoritmu SOM (čo je v skutočnosti prvotná verzia algoritmu [26], z ktorej potom prešiel Kohonen k ED verzii). Učiacie pravidlo tu spočíva v čiastočnej rotácii váhového vektora víťazného neurónu a jeho topologických susedov k vstupnému vektoru \mathbf{x} (jedná sa v podstate o učenie Hebbovho typu s následným normovaním váh), a to podľa vzťahu

$$\mathbf{w}_i(t+1) = \frac{\mathbf{w}_i(t) + \alpha(t) h(i^*, t) \mathbf{x}}{\|\mathbf{w}_i(t) + \alpha(t) h(i^*, t) \mathbf{x}\|} \quad (7.18)$$

pričom opäť možno použiť okolie pravouhlé (rovnica 7.5) alebo gaussovské (rovnica 7.6). Parameter $\alpha(t)$ predstavuje rýchlosť učenia, podobne ako v štandardnej verzii algoritmu, počiatočné hodnoty by v tomto prípade však mali byť $\gg 1$, aby sa urýchlil proces konvergencie. Normovanie váh je potrebné na to, aby sa zamedzil ich nekonečný rast počas učenia. Súčasne sa tým dosiahne efekt výberu víťaza na základe jediného parametra, a tým je uhol medzi váhovým vektorom a aktuálnym vstupným vektorom (ako je zrejmé z rovnice 7.17). Tým, že váhy sú normované, ležia na povrchu hypergule s dimenziou $N-1$. V princípe sa ponúkajú dve možnosti:

(a) Priamym aplikovaním DP verzie algoritmu SOM dostaneme aproximáciu vstupného priestoru *invariantnú od normy vstupných vektorov*, t.j. všetky vektory s tou istou orientáciou budú zobrazované na ten istý váhový vektor, ktorý s nimi zvierá najmenší uhol. Takto sa napr. trojrozmerné vstupy premietnu na povrch gule, čo je dvojrozmerná plocha.

(b) Ak chceme aproximovať vstupný priestor podobne ako so štandardnou, ED verzou Kohonenovho algoritmu, je potrebné najprv urobiť transformáciu vstupných dát do priestoru s dimenziou o jednu vyššou. Takýmto spôsobom napr. dvojrozmernú plochu tvorenú vektormi $\mathbf{x} = [x_1, x_2]^T$ premietneme na povrch gule pridaním tretej, konštantnej zložky. Pôvodné dve zložky vektora \mathbf{x} takto predstavujú uhly, tretia odpovedá norme vektora. Prevod zložiek rozšíreného vektora $\mathbf{x} = [x_1, x_2, 1]^T$ zo sférickej súradnicovej sústavy do karteziánskej súradnicovej sústavy $[s_1, s_2, s_3]^T$ je daný trojicou vzťahov



$$\begin{aligned}
 s_1 &= 1 \cdot \cos(x_1) \cdot \cos(x_2) \\
 s_2 &= 1 \cdot \sin(x_1) \cdot \cos(x_2) \\
 s_3 &= 1 \cdot \sin(x_2)
 \end{aligned}
 \tag{7.19}$$

Ak vektory $[s_1, s_2, s_3]^T$ budeme predkladať DP-SOM ako vstupy počas tréovania, sieť bude aproximovať dvojrozmernú plochu v trojrozmernom priestore. Keďže tretia dimenzia je redundantná, mapované súradnice budú odpovedať pôvodným súradniciam vektorov x . Súčasne treba poznamenať, že transformácia (7.19) je jednoznačná, ak platí $-\pi/2 \leq x_1 \leq \pi/2$ a taktiež $-\pi/2 \leq x_2 \leq \pi/2$.

Situácia je zobrazená na nasledovnej sérii obrázkov. Vstupnou množinou boli dáta s rovnomerným rozdelením na intervale $\langle -1, 1 \rangle^2$, transformované pomocou vzťahov (7.19) do trojrozmerného priestoru, čím sa premietli na časť pologule s kladnou prvou súradnicou. Obr. 7.15a-c zobrazujú váhové vektory SOM, pre názornosť premietnuté do roviny s nulovou x -ovou súradnicou. Ako vidieť, SOM v konečnej fáze aproximuje tréovaciu množinu.

7.6 Zachovanie topológie

Zo simulácií SOM na dvojrozmerných dátach s rovnomerným i nerovnomerným rozdelením bolo vidieť, že množina váhových vektorov mala tendenciu adekvátne pokryť dátovú oblasť (vlastnosť vektorovej kvantizácie, podkapitola 7.4.1). Čo je však dôležité, výsledné rozloženie váh nadobudlo v oboch prípadoch formu, ktorú možno intuitívne nazvať ako *usporiadanú*. Inými slovami, sieť sa javila ako plne rozvinutá, bez násilného kríženia uzlov (váhových vektorov). Takýto finálny stav siete súvisí s matematickým aspektom realizovaného zobrazenia — *vlastnosťou zachovania topológie*. Túto vlastnosť možno formulovať nasledovne:

Ak sa vstupný vektor \mathbf{x} zobrazí na neurón i , potom všetky vstupné vektory blízke \mathbf{x} (v zmysle euklidovskej metriky) sa zobrazia na ten istý neurón alebo na jeho topologických susedov.

Vlastnosť zachovania topológie medzi vstupným a výstupným priestorom má svoj význam. Ako bolo spomenuté v úvode kapitoly, v biologických neurónových sieťach je vďaka topografickým mapám dosiahnutá konzistentná a veľmi úsporná reprezentácia geometrie vstupných stimulov. Z matematického pohľadu nám takéto zobrazenie umožňuje zbaviť sa redundantných dimenzií a pracovať v kvalitatívne rovnakom (topologicky ekvivalentnom) výstupnom priestore. Takto sa nielen zníži výpočtová náročnosť riešenej úlohy, ale navyše, pri práci s dátami vo výstupnom priestore môžu byť výsledky konkrétnej úlohy lepšie (napr. pri klasifikácii dát [5]), než by sa dosiahli s dátami uvažovanými v ich pôvodnej reprezentácii, teda vstupnom priestore.

Podľa matematickej definície spojité zobrazenie medzi dvoma topologickými priestormi zachováva topológiu vtedy, ak je prosté (jednojednoznačné) a súčasne existuje k nemu inverzné zobrazenie, ktoré je tiež spojité. Také zobrazenie sa nazýva *homeomorfizmus*. Z tejto definície možno intuitívne dedukovať, že homeomorfné zobrazenie zachováva vzťahy podobnosti (blízkości). Voľnejšou požiadavkou pre zachovanie topológie je, že stačí, ak také zobrazenie zachováva len usporiadanie vzťahov podobnosti (pozri obr. 7.16). Vyplýva to z vety o topologickej ekvivalencii, ktorej dôkaz možno nájsť v práci [20]:

Veta (o topologickej ekvivalencii): *Nech f a g sú metriky a nech (X,f) , (Y,g) sú metrické priestory so spočítateľnými hustými podmnožinami⁶. Nech $H : X \rightarrow Y$ je bijektívne zobrazenie také, že platí:*

$$\forall \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4 \in X: f(\mathbf{x}_1, \mathbf{x}_2) < f(\mathbf{x}_3, \mathbf{x}_4) \Rightarrow g(H(\mathbf{x}_1), H(\mathbf{x}_2)) \leq g(H(\mathbf{x}_3), H(\mathbf{x}_4)) \quad \text{a}$$
$$\forall \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4 \in Y: g(\mathbf{y}_1, \mathbf{y}_2) < g(\mathbf{y}_3, \mathbf{y}_4) \Rightarrow f(H^{-1}(\mathbf{y}_1), H^{-1}(\mathbf{y}_2)) \leq f(H^{-1}(\mathbf{y}_3), H^{-1}(\mathbf{y}_4))$$

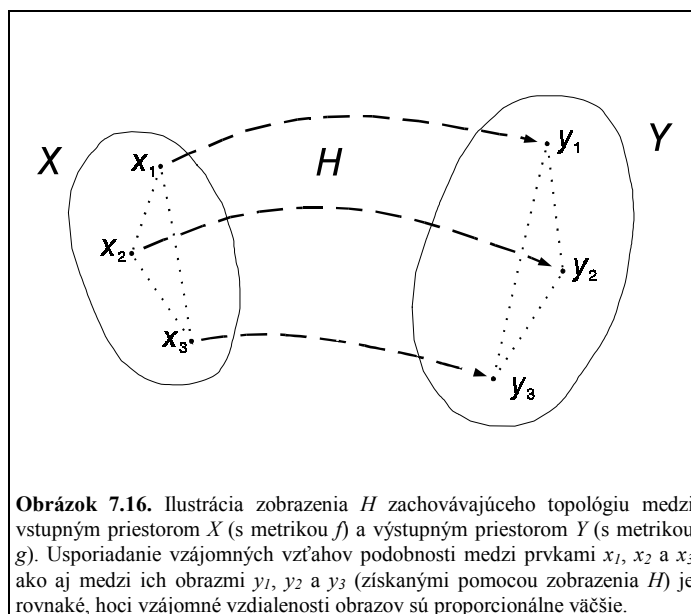
potom H je homeomorfizmus a priestory X a Y sú topologicky ekvivalentné.

⁶ Takýmto metrickým priestorom je napr. priestor $X = \mathfrak{R}^N$, $N < \infty$, s euklidovskou metrikou $f = d_E(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$.

Táto veta hovorí, že pre zachovanie topológie je nutnou podmienkou to, aby usporiadanie podobností (blízkosti) medzi dvojicami bodov vo vstupnom i výstupnom spojitom priestore bolo monotónne. Keďže toto je prísnejšia podmienka, než aká je potrebná na to, aby zobrazenie bolo homeomorfizmom, možno H nazvať *topologickým (topografickým) homeomorfizmom* [20].

V prípade diskretných priestorov (s ktorými máme do činenia pri riešení praktických problémov pomocou SOM) nie je spojitosť (podmienka homeomorfizmu) definovaná tradičným spôsobom pomocou metriky. Môžeme však povedať, že zobrazenie medzi diskretnými priestormi, ktoré spĺňa vlastnosť usporiadania podobností, je diskretnou aproximáciou topologického homeomorfizmu.

Pre potreby definovania vlastnosti zachovania topológie medzi diskretnými priestormi, teda pri použití neurónovej siete, bola zavedená aj iná, praktickejšia definícia tejto vlastnosti. Jej autorom je Martinetz [35], o práci ktorého sa ešte zmienime v podkapitole 7.9. Vychádza sa pri nej z predstavy, že máme množinu váhových vektorov, ktoré ležia na myslenej nelineárnej ploche — *variete* M (angl. *manifold*) danej štruktúrou vstupných dát, pričom uvažovanú SOM môžeme chápať ako *graf* pozostávajúci z vrcholov (neuróny) a spájajúcich hrán (definujúcich vzťahy susednosti neurónov). Najprv je potrebné



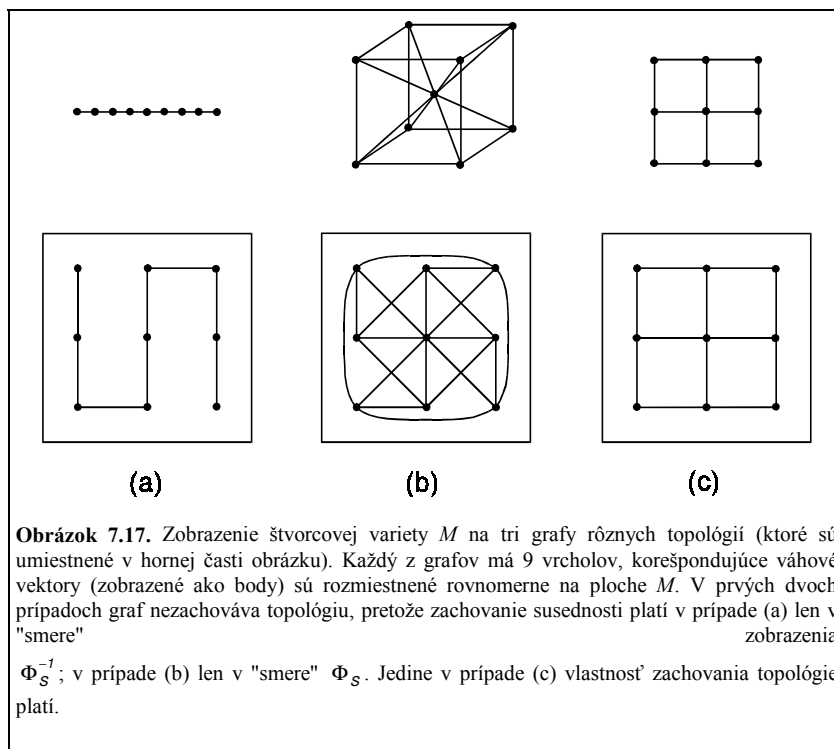
zadefinovať susednosť dvoch vektorov $\mathbf{w}_i, \mathbf{w}_j$ (bodov v euklidovskom priestore). Situácia je triviálna v jednorozmernom prípade (1D varieta M): dva body $\mathbf{w}_i, \mathbf{w}_j$ sú susedné, ak medzi nimi neleží žiadny bod \mathbf{w}_k . Použijúc termín Voronoiho polyédrov, ekvivalentným tvrdením pre N -rozmerný priestor je: dva body $\mathbf{w}_i, \mathbf{w}_j \in \mathfrak{R}$ sú susedné, ak ich Voronoiho polyédre sú susedné, t.j. ak $V_i \cap V_j \neq \emptyset$. Voronoiho polyéder je pre N -rozmerný prípad definovaný ako

$$V_i = \left\{ \mathbf{x} \in \mathfrak{R}^N \mid \|\mathbf{x} - \mathbf{w}_i\| \leq \|\mathbf{x} - \mathbf{w}_j\|, j=1, 2, \dots, n \right\} \quad i=1, 2, \dots, n \quad (7.20)$$

Za účelom zovšeobecnenia definície susednosti dvoch bodov vo viacrozmernom priestore, t.j. pre body $\mathbf{w}_i, \mathbf{w}_j \in M \subseteq \mathfrak{R}^N$ bol zadenovaný pojem *maskovaný* Voronoiho polyéder, ktorý predstavuje časť Voronoiho polyédra, ktorá leží na variete M (lebo nás zaujíma susednosť bodov $\mathbf{w}_i, \mathbf{w}_j$ na podpriestore tvorenom touto varietou). Množina všetkých maskovaných Voronoiho polyédrov tvorí celkovú hyperplochu M , t.j. $M = V_1^{(M)} \cup \dots \cup V_n^{(M)}$. Rozšírenie definície susednosti je potom priame: Dva body $\mathbf{w}_i, \mathbf{w}_j \in M \subseteq \mathfrak{R}^N$ sú susedné na M , ak ich maskované Voronoiho polyédre $V_i^{(M)} = V_i \cap M$, $V_j^{(M)} = V_j \cap M$ sú susedné, t.j. ak $V_i^{(M)} \cap V_j^{(M)} \neq \emptyset$. Teraz môžeme pristúpiť k zamýšľanej definícii zobrazenia zachovávajúceho topológiu.

Definícia (zachovania topológie): *Nech G je graf (neurónová sieť) s vrcholmi $i=1, 2, \dots, n$ a hranami (určujúcimi topológiu siete) definovanými pomocou (symetrickej) matice "susednosti" A s prvkami $A_{ij} = \{0, 1\}$, $i, j=1, \dots, n$. Nech $M \subseteq \mathfrak{R}^N$ je daná varieta a nech $S = \{\mathbf{w}_1, \dots, \mathbf{w}_n\}$ je množina bodov $\mathbf{w}_i \in M$, z ktorých každý je priradený k vrcholu i grafu G . Potom graf G so svojimi vrcholmi i odpovedajúcimi bodom $\mathbf{w}_i \in M$ predstavuje **mapu zachovávajúcu topológiu** práve vtedy, keď platí, že zobrazenie $\Phi_S: M \rightarrow G$ ako aj inverzné zobrazenie $\Phi_S^{-1}: G \rightarrow M$ zachovávajú susednosť.*

Ostáva ešte objasniť vlastnosť zachovania susednosti, ktorá, ako vidieť, musí platiť oboma smermi: Zobrazenie $\Phi_S: M \rightarrow G$ zachováva susednosť, ak každej dvojici bodov $\mathbf{w}_i, \mathbf{w}_j$, ktoré sú susedné na M , odpovedajú vrcholy i, j , ktoré sú susedné v grafe G (obr. 7.17a). Analogicky, zobrazenie $\Phi_S^{-1}: G \rightarrow M$ zachováva susednosť, ak každej dvojici susedných vrcholov i, j grafu G odpovedajú body $\mathbf{w}_i, \mathbf{w}_j$, ktoré sú susedné na M (obr. 7.17b).



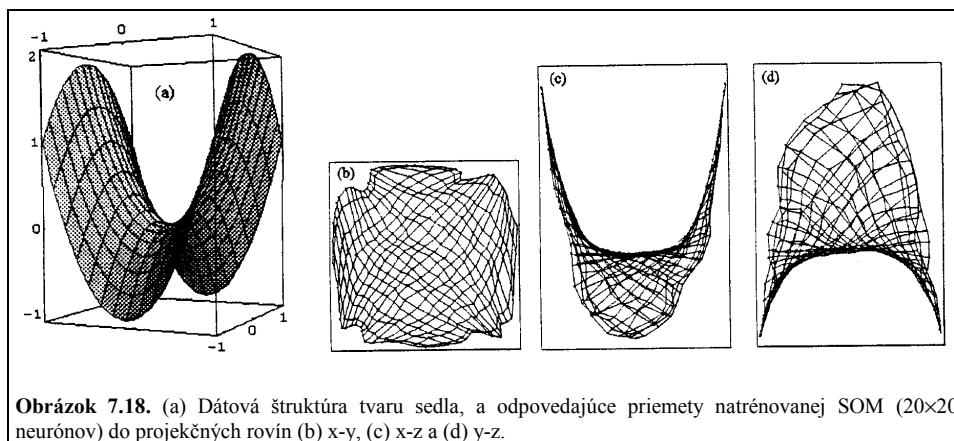
Pri spätnom pohľade na predošlé podkapitoly je vidieť, že s prípadmi (a) a (c) sme sa už v simuláciách stretli (obr. 7.8 a 7.10). Prípad (b) nastáva v praktických problémoch zriedka; už aj kvôli tomu, že dimenzia dát býva málokedy rovná jednej a topológiu siete obvykle volíme kvôli znázorniteľnosti nanajvyš rovnú dvom. Ak sa zamyslíme nad tým, ako zachovanie topológie medzi variétou M a grafom G súviselo s voľbou topológie SOM (grafu), uvedomíme si, že vlastnosť bola dodržaná, len ak sme použili SOM, ktorej topológia má dimenziu *zhodnú* s dimenziou dát (obr. 7.8), teda dva. Reťaz neurónov SOM siete tiež aproximuje funkciu hustoty dát, avšak vlastnosť zachovania topológie nemohla byť dodržaná, a to práve kvôli nerovnosti dimenzií. Pri pohľade na obr. 7.10 vidieť, že pre niektoré oblasti vstupov je vlastnosť splnená, t.j. že v rámci nich vybrané dva blízke vstupy majú topologicky blízkych víťazov, ale súčasne možno vybrať iné dva vstupy, pre ktoré to neplatí. Z pohľadu definície platí susednosť len v opačnom smere, Φ_S^{-1} .

Podmienka zhody dimenzií musí platiť vo všeobecnosti. Ak pracujeme s množinou dát, ktorej prvky sú popísané viacerými súradnicami ($\mathbf{x}_i \subseteq \mathcal{R}^N$), často nastáva situácia, že prvky \mathbf{x}_i nevyplňajú celý N -rozmerný priestor, ale len jeho časť. Inými slovami, skutočná, *inherentná dimenzia* geometrickej štruktúry tvorenej prvkami \mathbf{x}_i býva často menšia ako N . Inherentná dimenzia je daná minimálnym počtom premenných, potrebných na parametrický popis uvažovanej množiny dát. Ak teda zvolíme topológiu SOM⁷ tak, že jej dimenzia sa

⁷ Bežne sa používajú architektúry v tvare mriežky alebo reťaze, možno použiť i trojrozmerný "kryštál". V princípe je algoritmus SOM použiteľný i pre viacrozmerné architektúry, avšak má to viaceré

bude zhodovať s inherentnou dimenziou dát, potom je pravdepodobné, že SOM sa naučí aproximovať dátovú štruktúru, navyše topologicky usporiadane. "Pravdepodobné" preto, že správna aproximácia nastáva s výnimkou prípadov extrémnych nelinearit v dátach.

Situácia je ilustrovaná na obr. 7.18a. Dátová množina bola generovaná podľa vzťahu $z = 2x^2 - y^2 + \text{rnd}(-0,2;0,2)$, pričom $x \in (-1,1)$, $y \in (-1,1)$ a $\text{rnd}(:,.)$ je generátor bieleho šumu na uvedenom intervale.⁸ Vzhľadom k tomu, že rozptyl v smere z-ovej osi je relatívne malý, inherentná dimenzia generovanej množiny nimi tvorenej je len dva (pri parametrizácii zohľadňujúcej malý aditívny šum), hoci dáta ležia v trojrozmernom priestore. Pri použití SOM dostaneme výsledok, ktorého projekcie do rovín tvorených dvojicami súradníc sú na obr. 7.18b-d. Vďaka zhode dimenzií je dvojrozmerná SOM schopná pokryť (aproximovať) nelineárnu štruktúru (plochu) tvorenú dátami.



V praktických problémoch z toho vyplýva nutnosť *a priori* odhadnúť inherentnú dimenziu dát a následne zvoliť sieť s odpovedajúcou architektúrou (topológiou). V prípade nesprávneho odhadu je potom potrebné vyskúšať SOM s inou topológiou. Pri úspešnom použití siete dostávame *nelineárne zobrazenie* zo vstupného priestoru do priestoru neurónov, ktoré má tú dôležitú vlastnosť, že *redukuje dimenziu popisu dát*. Algoritmus SOM teda umožňuje "objaviť" a aproximovať nízkorozmerné nelineárne štruktúry pri zachovaní topológie, a to pri splnení vyššie spomínanej podmienky zhody dimenzií. Pre porovnanie, lineárne štatistické metódy ako napr. metóda hlavných komponent (angl. *principal component analysis*, PCA) sú schopné detekovať len lineárne vzťahy medzi vstupnými súradnicami, v dôsledku čoho sa napr. štruktúra tvaru sedla (obr. 7.18a) javí ako trojrozmerná, hoci jej inherentná dimenzia je rovná dvom.

nevýhody: s dimenziou exponenciálne narastá výpočtový čas i počet potrebných neurónov v sieti; navyše, stráca sa ilustratívnosť výstupnej reprezentácie, ktorá je vizualizovateľná len pre architektúry s dimenziou ≤ 3 .

⁸ Zobrazený útvar síce pre jednoduchosť predstavuje prípad pre $\text{rnd}(0,0)$, použité dáta pre tréningovanie SOM však mali z-ovú zložku "zašumenú" podľa stanoveného intervalu.

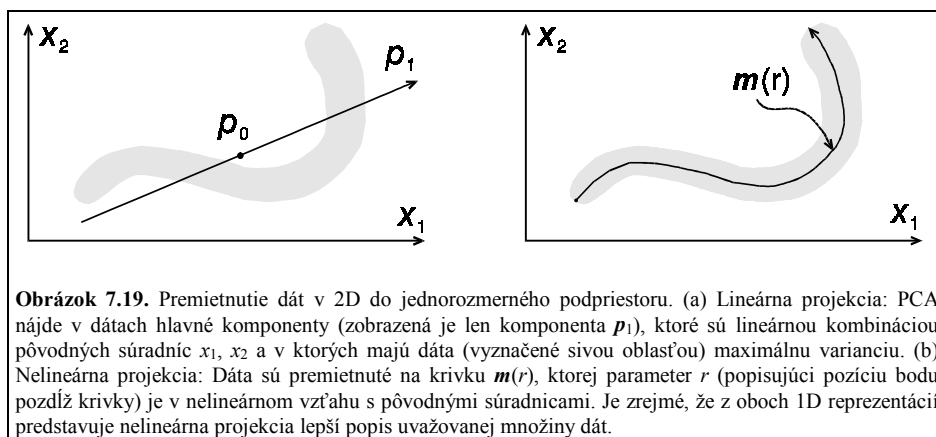
7.6.1 Extrakcia a topologické zobrazenie príznakov

Oba algoritmy — PCA i SOM — patria do kategórie algoritmov, ktoré umožňujú *extrahovať príznaky* (angl. *feature extraction*), t.j. charakteristické črty vstupných dát. Účelom extrakcie príznakov je transformovať dáta do priestoru nižšej dimenzie ($f: \mathfrak{R}^N \rightarrow \mathfrak{R}^D$), a to tak, aby sa ich vzájomné vzťahy (čo možno najviac) zachovali. Má to dve hlavné výhody: (a) jednak sa tým zníži dimenzia popisu dát, čím sa zjednoduší ich ďalšie spracovanie a (b) súčasne sa umožní vizualizácia a analýza dát (ak $D \leq 3$).

Algoritmy na extrakciu príznakov sú založené na klasických prístupoch [42] alebo sa jedná o algoritmy implementované pomocou neurónových sietí [33]. Hoci väčšina "neurónových" algoritmov na extrakciu príznakov je len priamou implementáciou klasických algoritmov (čo je i prípad PCA), neurónový prístup má niektoré výhody. Napríklad, implementácia Sammonovho algoritmu pomocou neurónovej siete už má schopnosť zovšeobecnenia, zatiaľ čo pôvodný algoritmus túto vlastnosť nemá [33].

V prípade PCA sa jedná o lineárnu transformáciu f . Tá "nájde" v dátach *hlavné komponenty* (angl. *principal components*), tvoriace ortogonálny súradnicový systém, v ktorých majú dáta maximálnu variáciu (stred novej súradnicovej sústavy sa pritom presunie do ťažiska \mathbf{p}_0 dátovej množiny, pozri obr. 7.19a). V situácii na obr. 7.19a ide o zobrazenie $f: \mathfrak{R}^2 \rightarrow \mathfrak{R}^1$ (do smeru hlavnej komponenty \mathbf{p}_1). Na jeho implementáciu stačí použiť jeden lineárny neurón s dvoma vstupmi (a jedno z pravidiel učenia Hebbovho typu [22]), pričom jeho výstup bude udávať hodnotu extrahovaného príznaku (súradnica v smere \mathbf{p}_1). Keďže nová súradnica je lineárnou kombináciou pôvodných súradníc, v prípade nelineárnej štruktúry dát sa stratí po premietnutí do lineárneho podpriestoru (kolmý priemet bodov \mathbf{x} na priamku \mathbf{p}_1) značné množstvo informácie.

V prípade nelineárnej projekcie na krivku $\mathbf{m}(r)$ (obr. 7.19b) sa táto strata informácie podstatne redukuje. Analogicky k lineárnemu prípadu, \mathbf{m} predstavuje *hlavnú krivku* (angl. *principal curve*) uvažovanej množiny dát [37]. Krivka \mathbf{m} je parametrizovaná len jednou premennou r , ktorá udáva pozíciu bodu na krivke (vyjadrenú napr. oblúkovou vzdialenosťou bodu od koncového bodu krivky). Jednorozmerná SOM je schopná takúto krivku diskretne (t.j. konečným počtom) aproximovať pomocou svojich váhových vektorov, a zobrazenie vstupu \mathbf{x} na výstupnú 1D reprezentáciu sa realizuje nájdením bodu (váhového vektora) na krivke, ktorý je najbližšie k \mathbf{x} , t.j. \mathbf{w}_{i^*} . Jeho pozícia na krivke je daná indexom i^* , čo je diskretná aproximácia parametra r . Keďže \mathbf{m} výstižne popisuje geometriu dát, jej parameter r predstavuje ich charakteristickú črtu — *príznak*, ktorého hodnota je pre aktuálny vstup i^* . V prípade aproximácie dvojrozsmernej plochy tvorenej dátami hovoríme o *hlavnej variete* (angl. *principal manifold*), ktorej parametrom je vektor \mathbf{r} s dvoma nelineárnymi súradnicami. Tie reprezentujú hodnoty dvoch extrahovaných príznakov.



Okrem iného, rozdiel medzi PCA a SOM spočíva v spôsobe reprezentácie príznakov. Pri PCA je na reprezentáciu jedného príznaku použitý len jeden neurón (hodnota príznakov je kvantifikovaná výstupmi neurónov), v prípade SOM sa na reprezentácii podieľajú všetky neuróny (hodnota príznakov je daná pozíciou víťaza). S tým súvisí špecifická vlastnosť SOM — extrahované príznaky sú *topologicky zobrazované* (angl. *feature mapping*), a to pozdĺž koordinát mriežky SOM. Podobne, na rozdiel od PCA, v prípade SOM môže byť interpretácia zobrazovaných príznakov v niektorých prípadoch obtiažna, čo samozrejme závisí od typu dát a súvisí s faktom, že nepoznáme transformačný vzťah f . Extrahované (a zobrazované) príznaky nemusia mať žiaden fyzikálny význam. Sieť si ich "zvolí" sama, výlučne v závislosti od geometrie dát.

7.6.2 Miery zachovania topológie

Na to, aby sme vedeli posúdiť, či nami zvolená SOM sa nachádza v usporiadanom stave, potrebujeme mať nejakú informáciu, ktorá charakterizuje

jej stav (t.j. pozície jej váhových vektorov), a to z hľadiska zachovania topológie. Najjednoduchšie je, ak taký kvantifikátor — *miera zachovania topológie* — je vyjadrený skalárnou hodnotou. Doteraz bolo navrhnutých niekoľko takýchto mier, pričom každá má iné vlastnosti. Spomenieme si niektoré z nich.

Miera usporiadania Θ . V práci [12] bola navrhnutá jednoduchá skalárna miera, ktorá kvantifikuje pravidelnosť usporiadania váhových vektorov SOM a je aplikovateľná pre ľubovoľnú dimenziu váhových vektorov. Bola definovaná ako $\Theta = \sigma_1 / \mu_1$, kde

$$\begin{aligned}\mu_1 &= E_{(i,j) \in A_1} (\|w_i - w_j\|) \\ \sigma_1 &= \sqrt{\text{Var}_{(i,j) \in A_1} (\|w_i - w_j\|)}\end{aligned}\tag{7.21}$$

pričom E je stredná hodnota, Var je variácia a A_1 predstavuje množinu dvojíc indexov, ktoré odpovedajú susedným neurónom v štruktúre SOM, t.j. $A_1 = \{(i, j) \mid \|r_i - r_j\| = 1\}$.

Presnejšie povedané, Θ odzrkadľuje rovnomernosť rozostupov medzi susednými dvojicami váhových vektorov (kvantifikovanú štandardnou odchýlkou σ_1), teda pri rovnomernom rozostupe váh $\Theta \rightarrow 0$. Tým, že Θ sa vzťahuje na okamžitý stav siete, možno ju aplikovať opätovne po každej iterácii (resp. každej n -tej iterácii), čím dostávame informáciu o stave usporiadania váhových vektorov v priebehu učenia. Miera Θ má však dva nedostatky: (1) Pri váhových vektoroch vyšších dimenzií podlieha dôsledku vety o centrálnej limite.⁹ Ako vyplýva zo vzťahu pre disperziu normy množiny náhodných vektorov, jej veľkosť sa pre dostatočne veľké n blíži ku konštantnej hodnote b , z čoho vyplýva, že vektory sú viacmenej normované. Tým dostávame pri aplikáciách s vysokorozmernými dátami hodnotu $\Theta \approx 0$ i v počiatočnom štádiu učenia, t.j. keď sú váhy ešte neusporiadane rozložené. Koniec koncov, i v prípade nízkorozmerných váh môže Θ nadobúdať "nesympaticky" nízku hodnotu, čo je však v tomto prípade spôsobené väčšou hodnotou menovateľa μ (je zrejmé, že ak sú váhy náhodne rozložené, je priemerná vzdialenosť susedných dvojíc väčšia). (2) Druhý nedostatok Θ spočíva v tom, že je ťažké z jej hodnoty usúdiť, či architektúra siete vyhovuje dimenzii dát vstupnej množiny (pozri tab. 7.1).

Topografický súčin P [2]. Podobne ako v predchádzajúcom prípade, na výpočet P nie je potrebná znalosť o štatistike vstupných dát, počíta sa výlučne zo vzájomných vzdialeností váhových vektorov vo vstupnom priestore a zo vzdialeností neurónov vo výstupnom priestore. Vzťah pre výpočet P je nasledovný:

⁹ Vetu o centrálnej limite možno stručne formulovať nasledovne: Majme množinu náhodných vektorov $x = [x_1, x_2, \dots, x_n]^T$, ktorých zložky sú nezávislé. Potom pre strednú hodnotu normy vektorov x a ich disperziu platia vzťahy $\mu_{\|x\|} = E(\|x\|) = \sqrt{an - b} + O(1/n)$ a $\sigma_{\|x\|}^2 = \text{Var}(\|x\|) = b + O(1/\sqrt{n})$. Parametre a a b závisia výlučne od centrálnych momentov rádu 1, 2, 3 a 4, a to podľa vzťahov $a = \mu^2 + \sigma^2$, $b = (4\mu^2\sigma^2 - \sigma^4 + 4\mu\mu_3 + \mu_4) / (4a)$, kde $\mu_k = E(\|x\| - \mu)^k$ je centrálny moment k -tého rádu.

$$P = \frac{1}{N \cdot (N-1)} \sum_{j=1}^N \sum_{k=1}^{N-1} \log \left[\prod_{m=1}^k \frac{d^{in}(\mathbf{w}_j, \mathbf{w}_{n_m^{out}(j)})}{d^{in}(\mathbf{w}_j, \mathbf{w}_{n_m^{in}(j)})} \cdot \frac{d^{out}(j, n_m^{out}(j))}{d^{out}(j, n_m^{in}(j))} \right]^{\frac{1}{2k}} \quad (7.22)$$

kde N označuje počet neurónov, $n_m^{in}(j)$ (resp. $n_m^{out}(j)$) predstavuje m -tého najbližšieho suseda j -teho neurónu v priestore váh (resp. neurónov), d^{in} sa vzťahuje na výpočet vzdialenosti v priestore váh a d^{out} na výpočet fyzických vzdialeností neurónov v štruktúre SOM.¹⁰ Ako vidieť, výpočet P je zložitejší, avšak poskytuje viac informácie v porovnaní s mierou Θ . Na základe hodnoty P možno zistiť, či zvolená architektúra siete "pasuje" na vstupné dáta (v takom prípade $P \approx 0$); navyše dáva informáciu o tom, či v prípade potreby treba znížiť alebo zvýšiť dimenziu architektúry siete, aby sme dostali lepšiu hodnotu P . Z hodnôt uvedených v tab. 7.1 vidieť, že najlepší výsledok pre danú vstupnú množinu bol získaný so sieťou 16×16 , ktorá svojou štruktúrou a pomerom dĺžok strán najlepšie odpovedá vstupným dátam. V iných prípadoch, podhodnotenie optimálnej dimenzie (prípade reťaze) znamená $P < 0$, nadhodnotenie dimenzie (kryštál) dáva $P > 0$. V prípadoch 64×4 a 32×8 má sieť síce správnu dimenziu, avšak vzhľadom k pomeru jej strán sa sieť javí ako kvázi-jednorozmerná, preto $P < 0$.

Miera C. V práci [20] bola navrhnutá skalárna miera, ktorá, použijúc označenie pre neurónovú sieť, má tvar:

$$C = \sum_{i=1}^N \sum_{j < i} F(i, j) \cdot G(\mathbf{w}_i, \mathbf{w}_j) \quad (7.23)$$

kde N je počet neurónov, F je funkcia podobnosti vo vstupnom priestore, G vo výstupnom priestore. Pri uvažovaní funkcií F a G ako euklidovských vzdialeností platí, že čím "lepšie" zobrazenie, tým väčšia hodnota C . Uvedená miera bola odvodená z podmienky zachovania usporiadania podobnosti medzi dvoma priestormi. Autor dokázal, že ak medzi dvoma priestormi existuje homeomorfizmus, tak zobrazenie, pri ktorom je zachované usporiadanie podobnosti, dáva maximálnu hodnotu C . Z hodnôt uvedených v tabuľke 7.1 možno usúdiť, že v rámci architektúr tej istej dimenzie hodnota C odpovedá tej najlepšej, t.j. 16×16 neurónov. Problematické je však na základe C určiť, či zvolená architektúra SOM vyhovuje dimenzii vstupných dát.

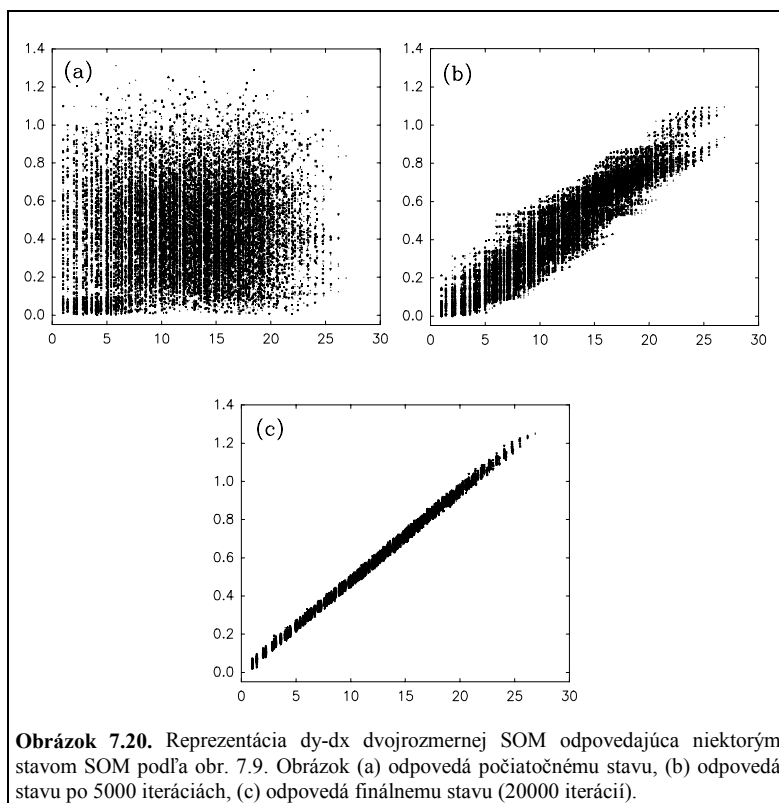
Reprezentácia dy-dx. Okrem skalárnych mier užitočnou pomôckou pri práci so SOM môže byť i grafická reprezentácia stavu. Príkladom takejto je reprezentácia $dy-dx$ [11], ktorá poskytuje grafickú informáciu o pravidelnosti usporiadania siete (je v podstate zovšeobecnením miery Θ). Spočíva v zobrazení vzájomných vzdialeností váhových

¹⁰ Vzájomné vzdialenosti neurónov v SOM nadobúdajú len diskkrétne hodnoty, a to $1, \sqrt{2}, 2, \sqrt{5}, \dots$ atď. Vyplýva to z pravidelnosti usporiadania neurónov v uvažovanej štruktúre, pričom vzdialenosť medzi dvoma susednými neurónmi je rovná jednej.

vektorov $dy(i, j) = \|\mathbf{w}_i - \mathbf{w}_j\|$ ako funkcie vzájomných vzdialeností im odpovedajúcich neurónov v štruktúre SOM, t.j. $dx(i, j) = \|\mathbf{r}_i - \mathbf{r}_j\|$. Výsledkom zobrazenia je množina bodov, z ktorých každý odpovedá nejakej dvojici (i, j) a je zobrazený v rovine $dy-dx$. Ako vidieť na obr. 7.20, reprezentácia $dy-dx$ graficky popisuje mieru korelácie medzi dy a dx , ktorá závisí od stavu usporiadania váhových vektorov. Dá sa však očakávať, že pri aproximácii nelineárnych dátových štruktúr nebude stav usporiadania siete podľa tejto reprezentácie taký evidentný, čo vyplýva z komplikovanejšej korelácie medzi dy a dx .

Tabuľka 7.1. Priemerné hodnoty spomínaných mier zachovania topológie, pre rôzne architektúry SOM po natrénovaní. Vstupnou množinou boli vo všetkých prípadoch dvojrozmerné vektory s rovnomerným rozdelením na intervale $\langle 0,1 \rangle^2$, vybrané náhodne. Optimálnou architektúrou spomedzi uvádzaných je sieť typu 16×16 neurónov, ktorá najlepšie vyhovuje geometrii vstupných dát.

Architektúra SOM	Θ	P	C
256	0,350±4,4%	-0,1050±9,5%	1694182±1,7%
64×4	0,468±3,4%	-0,0660±3,0%	529876±1,6%
32×8	0,356±7,0%	-0,0301±3,3%	552382±1,3%
16×16	0,232±5,0%	0,0005±4,0%	649241±0,4%
10×10×2	0,604±0,7%	0,0076±3,9%	73663±0,3%
6×6×6	0,501±12,%	0,0382±0,2%	94220±0,4%

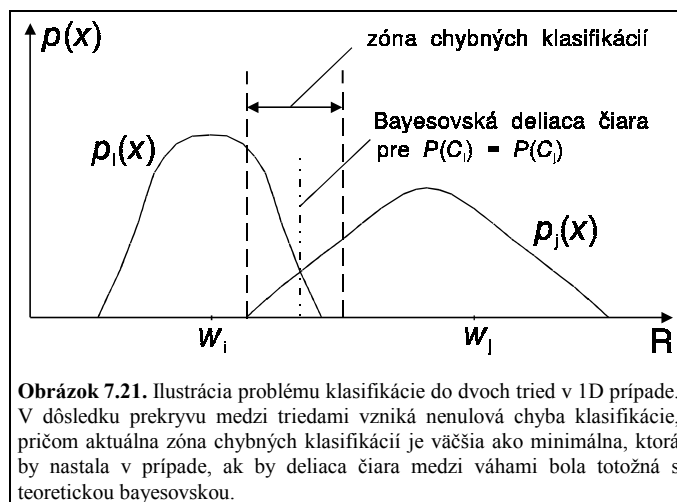


7.7 Hybridné učenie s učiteľom — algoritmy LVQ

Doteraz sme o SOM uvažovali ako o samoorganizovanom modeli, ktorý je schopný premietnuť vstupné vzory do priestoru nízkej dimenzie, a to pri zachovaní ich vzájomných topologických vzťahov. Teraz si ukážeme, ako možno takúto SOM následne použiť na štatistické rozpoznávanie vzorov. Za týmto účelom boli vyvinuté *algoritmy LVQ (Learning Vector Quantization)* [29,25], ktoré sa aplikujú po natrénovaní SOM. Ich úlohou je doladiť váhové vektory tak, aby sa minimalizoval počet chybných klasifikácií, ktoré vznikajú v dôsledku prekryvu medzi triedami.

Pri úlohe štatistického rozpoznávania vzorov pracujeme s množinou X vstupných vzorov x , z ktorých každý patrí do jednej z uvažovaných tried. Úlohou klasifikátora je po predložení vstupu x rozhodnúť, do ktorej triedy patrí. V prípade SOM sa klasifikácia vstupu x určí na základe návestia jemu najbližšieho váhového vektora, ktoré podobne ako u všetkých váhových vektorov označuje príslušnosť k tej-ktorej triede. Na rozdiel od problému vektorovej kvantizácie, s ktorou sme sa stretli v podkapitole 7.4.1, tu nie je dôležité, ktorý konkrétny neurón zvíťazí; podstatné je, aby to bol jeden z neurónov reprezentujúcich tú "správnu" triedu. Z toho možno usúdiť, že problematické situácie môžu

vznikať práve v oblasti hraníc medzi susediacimi triedami. Preto kľúčovým krokom je stanovenie *optimálnych rozhodovacích plôch* (dimenzie $N-1$ pri uvažovaní $X \subset \mathcal{R}^N$), ktoré rozdelia vstupný priestor X na zóny prislúchajúce jednotlivým triedam.



Problém je v jednorozmernom prípade (vstupy i váhy skalárne) znázornený na obr. 7.21, na ktorom je každá z dvoch tried i a j (s prekrývajúcimi sa funkciami hustôt $p_i(x)$ a $p_j(x)$) reprezentovaná jednou váhou. V dôsledku neoptimálnych pozícií váh je deliaca čiara medzi nimi (pravá čiarkovaná čiara) posunutá od optimálnej bayesovskej, ktorá spĺňa podmienku (pozri napr. [44], kapitola 4)

$$p(x|C_i) \cdot P(C_i) = p(x|C_j) \cdot P(C_j), \quad (7.24)$$

kde $p(x|C_i)$ je funkcia hustoty vstupných vzorov x , ktoré patria do triedy C_i a $P(C_i)$ je apriórna pravdepodobnosť, že pri náhodnom výbere vstupu x bude ten patriť do triedy C_i . Pri znalosti pravdepodobností figurujúcich v rovnosti (7.24) možno uvažovať o *diskriminačných funkciách* tvaru $\delta_i(x) = g(p(x|C_i)P(C_i))$, kde g je ľubovoľná monotónne rastúca funkcia. Klasifikácia sa potom vykoná na základe nájdenia $\max(\delta_i(x))$; na hraniciach medzi triedami C_i a C_j platí $\delta_i(x) = \delta_j(x)$.

Snahou LVQ je aproximovať teoretické bayesovské hranice bez znalosti štatistiky vstupných dát. Podobne ako v algoritme SOM, i tu sa adaptácia uskutočňuje iteratívnym posunom váhových vektorov — prototypov, avšak využívajúc tréningové vstupy s návěstiami, teda s *učiteľom*.

Prv než možno pristúpiť k algoritmu LVQ, je potrebné váhové vektory najprv *inicializovať* a potom *priradiť návěstie* každému z nich. Jednou z možností inicializácie je samotný algoritmus VQ (podkapitola 7.4.1), ktorý má tendenciu minimalizovať strednú kvadratickú chybu (vzťah 7.7, resp. 7.8). V prípade, že sa dá predpokladať vysoká štruktúrovanosť dát (ležia na nízkorozmernom podpriestore), je efektívne použiť na inicializáciu SOM, ktorá navyše umožňuje odhaliť a zobrazit' vzájomné topologické vzťahy medzi vstupnými vzormi.

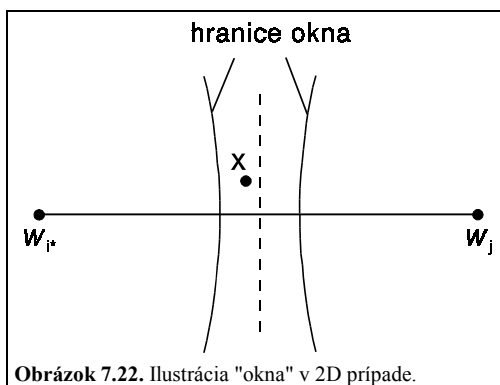
Následné priradenie návští možno realizovať tak, že pri testovaní na reprezentatívnej množine vstupov so známou klasifikáciou sa vyhodnotí početnosť "vítazstiev" každého prototypu a návstvie sa určí na základe väčšinového pravidla. Takýmto spôsobom získame pre každú triedu (obvyčajne) viacero prototypov, ktoré tvoria konvexné množiny.

LVQ1. Prvá verzia algoritmu LVQ na doladenie prototypov využíva koncept "odmeny a trestu" (ktorý figuruje aj pri perceptróne), a to v aplikácii len na víťazný neurón (nájdenny podľa minima euklidovskej vzdialenosti):

$$\begin{aligned} \mathbf{w}_{i^*}(t+1) &= \mathbf{w}_{i^*}(t) + \alpha(t) \cdot [\mathbf{x}(t) - \mathbf{w}_{i^*}(t)], & \text{ak } cls(\mathbf{x}) = cls(\mathbf{w}_{i^*}) \\ \mathbf{w}_{i^*}(t+1) &= \mathbf{w}_{i^*}(t) - \alpha(t) \cdot [\mathbf{x}(t) - \mathbf{w}_{i^*}(t)], & \text{ak } cls(\mathbf{x}) \neq cls(\mathbf{w}_{i^*}) \end{aligned} \quad (7.25)$$

pričom $\alpha(t) \in (0,1)$ je monotónne klesajúca funkcia, avšak s výrazne menšou počiatočnou hodnotou, rádovo 0,01 (v porovnaní s algoritmom SOM). $cls(\mathbf{x})$ označuje triedu, do ktorej \mathbf{x} patrí. Efekt, ktorý možno pozorovať pri LVQ1, je tendencia prototypov vzdäľovať sa od bayesovských hraníc. LVQ1 je však stabilný pri "slušnom" tvare funkcií hustôt jednotlivých tried a vhodnej inicializácii prototypov.

LVQ2. Na rozdiel od LVQ1 sa v LVQ2 adaptujú v každej iterácii dva prototypy — víťazného neurónu i^* a druhého najbližšieho neurónu j k vstupu \mathbf{x} . Snahou LVQ2 je posunúť deliacu čiaru medzi nimi smerom k pomyslenej bayesovskej hranici. Pre tento účel sa zavádza pojem "okna" W okolo deliacej čiary, ako to vidieť na obr. 7.22.



Obrázok 7.22. Ilustrácia "okna" v 2D prípade.

Korekcie sú definované ako

$$\begin{aligned} \mathbf{w}_{i^*}(t+1) &= \mathbf{w}_{i^*}(t) - \alpha(t) \cdot [\mathbf{x}(t) - \mathbf{w}_{i^*}(t)] \\ \mathbf{w}_j(t+1) &= \mathbf{w}_j(t) + \alpha(t) \cdot [\mathbf{x}(t) - \mathbf{w}_j(t)] \end{aligned} \quad \text{pre } \mathbf{x} \in W \quad cls(\mathbf{x}) = cls(\mathbf{w}_j) \neq cls(\mathbf{w}_{i^*}) \quad (7.26)$$

pričom platí, že \mathbf{x} leží v okne W , ak $\min(d_{i^*} / d_j, d_j / d_{i^*}) > s$, kde $d_j = \|\mathbf{x}(t) - \mathbf{w}_j\|$. Hranice okna tvoria Apolloniove hyperplochy (dimenzie $N-1$), pričom ak w je relatívna šírka okna W v jeho najužšom mieste, tak $s = (1-w)/(1+w)$. Optimálna veľkosť okna závisí od počtu tréningových vzorov: čím viac vzorov, tým užšie okno možno zvoliť. Aby sa však

dosiahla dobrá štatistická presnosť, počet vzorov ktoré "padnú" do okna musí byť dostatočne veľký. Kohonen uvádza kompromisnú hodnotu $w = 0,2$.

Ako sa však ukázalo, po väčšom počte iterácií (rádovo 10000) spôsoboval LVQ2 zhoršovanie výsledkov, a to kvôli jeho tendencii monotónne znižovať vzdialenosti $\|\mathbf{w}_{i^*} - \mathbf{w}_j\|$. Na kompenzáciu tohto nežiaduceho efektu navrhol Kohonen modifikáciu LVQ2.1, využívajúc rovnaké vzťahy (7.26), avšak s podmienkou, že \mathbf{w}_{i^*} a \mathbf{w}_j sú dva najbližšie prototypy k \mathbf{x} (teda i^* nemusí byť víťaz) a súčasne platí $\mathbf{x} \in W$ a $cls(\mathbf{x}) = cls(\mathbf{w}_j) \neq cls(\mathbf{w}_{i^*})$. Takáto modifikácia má za následok dvojnásobný počet korekcií prototypov (v porovnaní s LVQ2), pričom je zabezpečený i nárast $\|\mathbf{w}_{i^*} - \mathbf{w}_j\|$, čím sa zvyšuje štatistická presnosť v učení.

LVQ3. Nakoľko i pri LVQ2.1 bolo pozorované čiastočné zhoršovanie výsledkov po dlhšom tréovaní v dôsledku posunov hraničných prototypov do suboptimálnych pozícií, prišiel autor s myšlienkou zahrnúť do algoritmu efekt "kohézie" prototypov v rámci tried. Týka sa modifikácie prototypov v prípade *jednoznačne klasifikovateľných* vstupov, t.j. tých, ku ktorým sú dva najbližšie prototypy z tej istej triedy. Učenie má tvar

$$\begin{aligned} \mathbf{w}_{i^*}(t+1) &= \mathbf{w}_{i^*}(t) - \alpha(t) \cdot [\mathbf{x}(t) - \mathbf{w}_{i^*}(t)] && \text{pre } \mathbf{x} \in W \text{ a } cls(\mathbf{x}) = cls(\mathbf{w}_j) \neq cls(\mathbf{w}_{i^*}) \\ \mathbf{w}_j(t+1) &= \mathbf{w}_j(t) + \alpha(t) \cdot [\mathbf{x}(t) - \mathbf{w}_j(t)] \end{aligned} \quad (7.27a)$$

$$\begin{aligned} \mathbf{w}_k(t+1) &= \mathbf{w}_k(t) + \varepsilon \cdot \alpha(t) \cdot [\mathbf{x}(t) - \mathbf{w}_k(t)], \quad k \in \{i, j\} \\ &&& \text{pre } cls(\mathbf{x}) = cls(\mathbf{w}_j) = cls(\mathbf{w}_{i^*}) \end{aligned} \quad (7.27b)$$

kde parameter ε sa volí v závislosti od veľkosti okna: pri jeho malej šírke musí byť ε tiež malé. Na základe simulácií dospel autor k záveru, že vhodným rozsahom hodnôt je interval (0,1-0,5). Simulácie taktiež potvrdili stabilizačný účinok LVQ3 i pri rozsiahlejších dobách tréovania (rádovo 100000). Voľba optimálnej verzie algoritmu závisí od problému. Avšak štatisticky významné rozdiely vo výslednej chybe klasifikácie sa v simuláciách nepotvrdili. V čom sa verzie viac líšia, je ich stabilita. Z tohto hľadiska je vhodnejšie voliť LVQ1 alebo LVQ3, ktoré majú stabilizačný účinok i v dlhších tréovacích cykloch.

7.8 Niektoré aplikácie SOM

Aplikácií SOM je veľa, rozsiahly zoznam referencií na ne možno nájsť v Kohonenovej knihe o SOM [30]. Tu spomenieme aspoň niektoré z nich.

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 natréovaná SOM — tzv. *fonémová mapa* (obr. 7.23), ktorá svojou odpoveďou je schopná lokalizovať odozvu na

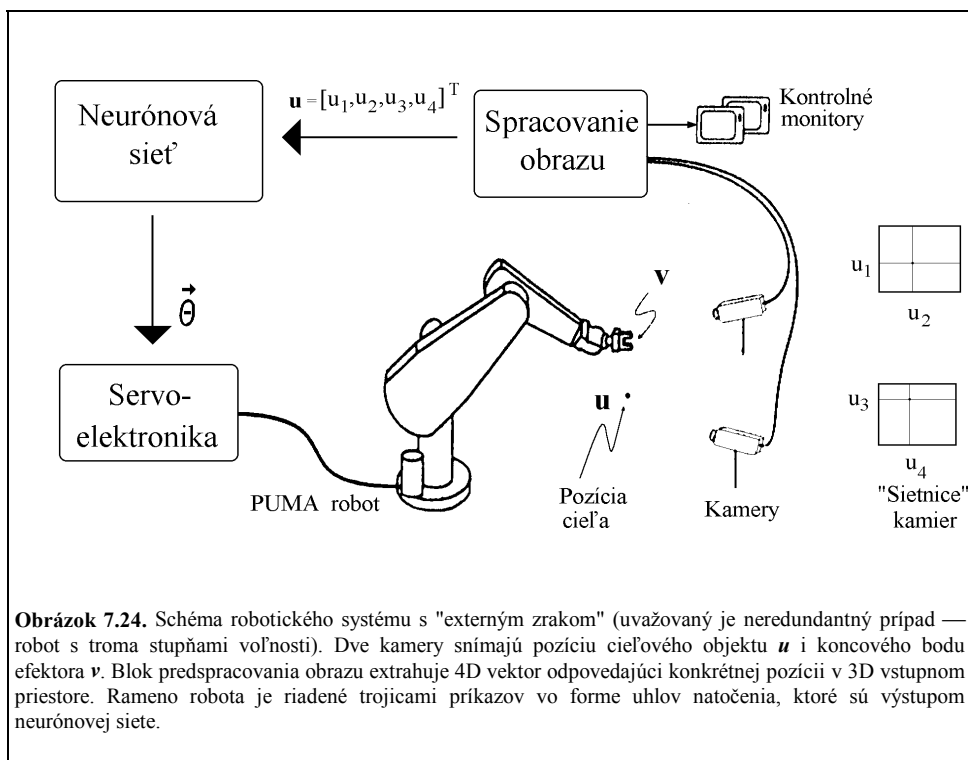
aktuálny vstup x odpovedajúci nejakej hláske (fonéme). Aplikácia bola urobená pre fínsky a japonský jazyk, a to predovšetkým z dvoch dôvodov: (a) oba jazyky sú inflexnými jazykmi (t.j. príbuzné slová sa v nich tvoria pomocou predpôn a prípon), preto je výhodnejšie voliť malé akustické jednotky (hlásky, 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 použijúc Hammingovo okno, (4) logaritmicizácia spektra a jeho vyhladenie, (5) umiestnenie 15 zložiek vhodným zlúčením komponent FFT v rozsahu 200-5000 Hz do spektrálneho vektora x , (6) centrovanie x okolo strednej hodnoty (odčítaním strednej hodnoty od všetkých zložiek), (7) normovanie 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 $x \in \mathcal{R}^{15}$. 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 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.

a	a	a	ah	h	ä	ä	∅	∅	e	e	e	
	o	a	a	h	r	ä	l	∅	y	y	j	i
o	o	a	h	r	r	r	g	g	y	j	i	
	o	o	m	a	r	m	n	m	n	j	i	i
l	o	u	h	v	v	n	n	h	hj	j	j	
	l	u	v	v	p	d	d	t	r	h	hi	j
•	•	•	u	v	tk	k	p	p	p	r	k	s
•	•	•	v	k	pt	t	p	t	p	h	s	s

Obrázok 7.23. Fonémová mapa (s hexagonálnou štruktúrou). Dvojité návestia označujú neuróny, ktoré reagujú na dve fonémy. Rozlíšenie niektorých foném nie je spoľahlivé, nutná je doplnková analýza. Prevzaté z [28].



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 transformáciu $\vartheta: \mathbf{u} \in U \subseteq \mathfrak{R}^4 \rightarrow \theta \in \mathfrak{R}^3$ 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 :

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

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])

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

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

kde ε , ε' , $h(i^*, i)$ 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 spriemerovaním príspevkov od všetkých neurónov v okolí víťaza podľa rekurentného vzťahu

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

kde $s = \sum_i h(i^*, i)$ a \mathbf{v}_n je pozícia efektora v n -tej iterácii. Opakovane 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.

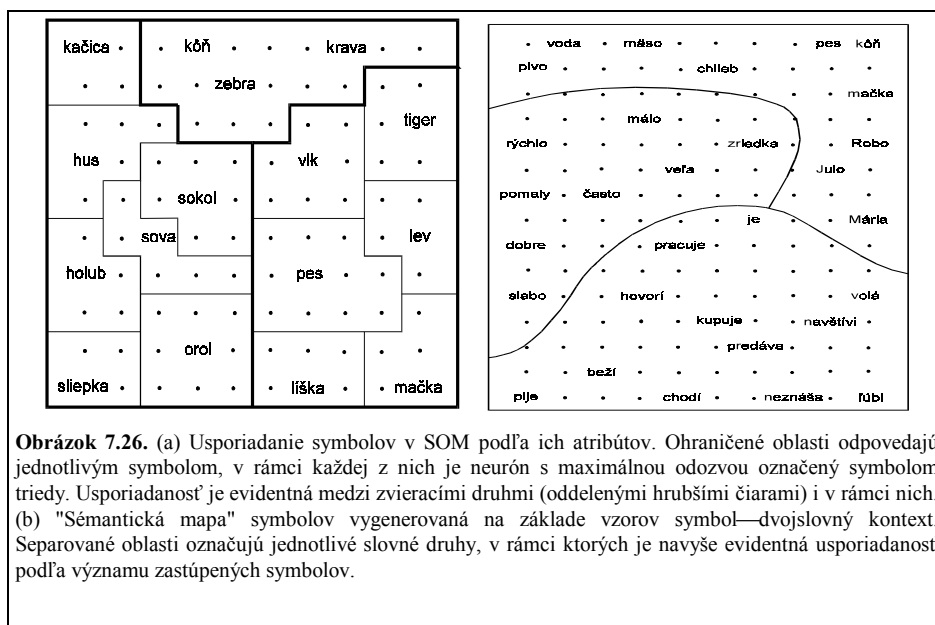
Formovanie hierarchických reprezentácií. SOM je schopná topologicky zobrazit' hierarchické vzťahy medzi jednotlivými vstupnými prvkami, pokiaľ sú tieto vhodne popísané pomocou svojich súradníc [25]. Na ilustráciu tejto úlohy možno použiť reprezentáciu podľa tab. 7.2.

Tabuľka 7.2. Množina prvkov použitá na formovanie hierarchickej reprezentácie. Každý z prvkov (ozn. 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	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
0	0	0	0	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	9	10	11	12	
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	0	0	0	0

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

Vzťahy medzi jednotlivými prvkami množiny možno zobrazit' 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 zrejماً: jednotlivé "vetvy" v SOM sú síce všelijako stočené (tak, aby sa "zmestili" do mapy), avšak topologické vzťahy medzi susednými vzormi sú v podstate rovnaké.



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 x_a bol vygenerovaný pre každé zviera, pričom kódy zvierat x_s boli zámerne vytvorené tak, aby neniesli žiadnu informáciu o vzájomnej podobnosti medzi symbolmi: každý vektor 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 $x=[x_s, x_a]^T$ charakterizujúci každé zviera, pričom hodnota a bola stanovená na $a = 0,2$, aby vplyv atribútovej časti vektora x (nosiť informácie o zvierati) bol väčší ako vplyv symbolovej časti. Vektory x boli napokon normované kvôli lepšej stabilizácii učenia. Počas tréningu bolo SOM prezentovaných 2000 náhodne vybraných vzorov x z 16-prvkovej množiny. Proces určovania návští bol však realizovaný na základe vektorov $x=[x_s, \mathbf{0}]^T$, čoho výsledkom je mapa na obr. 7.26a. Kvalitatívne rovnakú reprezentáciu by sme dostali i pri použití $x=[x_s, x_a]^T$, z čoho vyplýva, že hoci reprezentácia vzájomných vzťahov podobnosti bola získaná vďaka prítomnosti atribútovej časti počas tréningu, správna

odozva SOM v testovacej fáze sa generuje i pri absencii 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 zahŕňovala 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 nejakým spôsobom 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 $x=[x_s, x_a]^T$ 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.

7.9 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 "rovnomernosť" 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 "rovnomernosť" 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 samo-organizácie 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

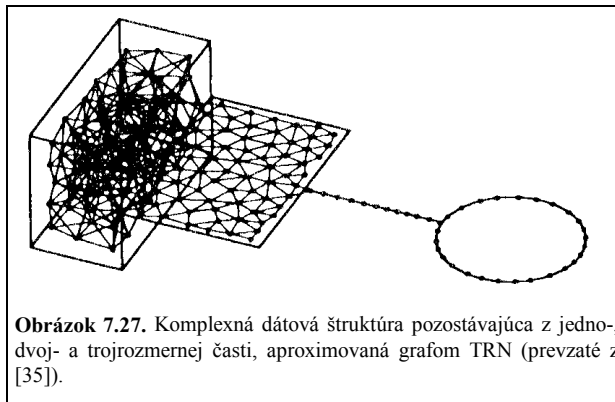
$$E = \frac{1}{2} \sum_i \sum_{j \neq i} (X_{ij} - Y_{ij})^2 \cdot F(Y_{ij}) \quad (7.31)$$

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. $F(\cdot)$ 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 $\Delta \mathbf{y}_i = \alpha \sum_{j \neq i} G(X_{ij}, Y_{ij})(\mathbf{y}_j - \mathbf{y}_i)$, 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)

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

kde F' označuje deriváciu. Tým 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 pozerat' 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 spojení $C[n \times n]$ 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)

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

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 $C_{i_0 i_t} = 0$, spoj uzly i_0 a i_t , t.j. prirad' $C_{i_0 i_t} > 0$
- (ii) resetuj "vek" hrany $i_0 - i_t$, t.j. $t(i_0, i_t) \leftarrow 0$ a zväčš "vek" všetkých spojení s víťazom, t.j. $t(i_0, j) \leftarrow t(i_0, j) + 1 \quad \forall j \mid C_{i_0 j} > 0$
- (iii) zruš všetky spojenia s víťazom, ktorých vek presiahol limit T , t.j. nastav $C_{i_0 j} \leftarrow 0, \quad \forall j \mid (C_{i_0 j} > 0) \wedge (t(i_0, j) > T)$.

Pri modifikácii spojení (hrán) $C_{i_0 i_t}$ podľa vzťahu $\Delta C_{i_0 i_t} \propto y_{i_0} \cdot y_{i_t}$ sa výstup neurónov počíta 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, ako napr. na obr. 7.27.

GCS (Growing Cell Structures) [19]. Zatiaľ čo doteraz spomínané modely pracovali s vopred stanoveným, konštantným počtom neurónov, v GCS sa neuróny i pridávajú resp.

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

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. $err(i) \leftarrow err(i) + \|\mathbf{x} - \mathbf{w}_i\|$. 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 sused f (t.j. taký, ktorý má s ním spojenie). Do stredu medzi ne sa vloží nový neurón (nn): $\mathbf{w}_{nn} = (\mathbf{w}_{bs} + \mathbf{w}_f) / 2$, a vytvorí 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

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

pričom A_{i*j} sú prvky matice susednosti (pozri definíciu, podkapitola 7.6), ktoré sa modifikujú trochu zložitejším spôsobom ako u TRN:

$$A_{ij}(t+1) = \begin{cases} 1 & \text{ak } y_i \cdot y_j = \max\{y_k \cdot y_l\}, \quad k, l = 1, 2, \dots, n \\ 0 & \text{ak } A_{ij}(t) < \theta \\ \varepsilon \cdot A_{ij}(t) & \text{v ostatných prípadoch.} \end{cases} \quad (7.35)$$

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 zobrazení 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

Vo vzťahu (7.35) $y_i = f(\|\mathbf{x} - \mathbf{w}_i\|)$ označuje výstup i -tého neurónu, kde f je kladná monotónne klesajúca funkcia, ε predstavuje konštantu zabúdania, ktorá spĺňa podmienku $0 < \varepsilon < 1$, a θ označuje prah eliminácie spojenia.

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.

Porovnaním spomínaných algoritmov samoorganizácie z hľadiska ich charakteristických črt dospejeme k prehľadovej tabuľke 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, 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, A.K. Krishnamurthy, P. Chen, and D.E. Melton. 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, M. Cottrell, J.C. Fort, and G. Pagés. 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. Pagés. 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, 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. *Self-Organizing Maps*. Springer, 1995.
- [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. Pagés. 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.