KAPITOLA 13

VYUŽITIE NEURÓNOVÝCH SIETÍ PRE KOMPRESIU OBRAZU

13.1 NEUROCOMPUTING

Neurocomputing je technologická disciplína, týkajúca sa systémov spracovania informácií - napr. neurónových sietí (neural networks) - ktorá je schopná vyvinúť operačné schopnosti ako adaptívnu odpoveď informačnému prostrediu. Je to nový a rozdielny prístup k spracovaniu informácií. Stal sa prvou alternatívou ku konvenčnému prístupu založenému na programovaných výpočtových prostriedkoch (programmed computing) [128].

Za jeden z najvýznamnejších rozdielov môžeme považovať to, že konvenčné počítačové a informačné systémy pracujú prevažne podľa dopredu daného presného postupu - algoritmu, podľa ktorého sa postupne spracovávajú jednotlivé operácie. Naproti tomu neurónové systémy uskutočňujú veľmi vysoký počet jednotlivých operácií súčasne a predovšetkým pracujú bez dopredu zadaného algoritmu. Ich činnosť je založená na procese učenia, pri ktorom sa neurónová sieť čo najlepšie adaptuje k riešeniu danej úlohy [129].

Žiadny človek nemôže invertovať maticu alebo riešiť systém diferenciálnych rovníc rýchlosťou, ktorá by bola porovnateľná so súčasnými pracovnými stanicami. Avšak žiadny počítačový systém sa nemôže rovnať schopnosti ľudského vizuálneho systému rozoznávať objekty rôznych tvarov a orientácií. Problémy riešené efektívnejšie neurónovými sieťami majú typické dve charakteristiky:

- sú všeobecne zle definované
- zvyčajne vyžadujú enormné množstvo operácií [131].

Neurónové siete sú teda výhodné predovšetkým pre prácu s neurčitými, nepresnými, neúplnými a tiež navzájom čiastočne rozpornými informáciami [129]. Treba si však uvedomiť, že napriek všetkým aspektom svojej výhodnosti je táto informačná technológia iba na počiatku svojho vývoja. Je veľmi pravdepodobné, že budúce informačné systémy budú oba prístupy (konvenčný a neurónový) účelne kombinovať.

Táto kapitola sa prehľadovo zaoberá využitím neurónových sietí pre kompresiu obrazových údajov (so snahou minimálneho využitia matematického aparátu). Ukázané sú modely neurónových sietí, ktoré sú v súčasnosti mimoriadne frekventované v kompresii údajov. Tieto modely zahŕňajú sieť so spätným šírením, Kohonenovu samoorganizujúcu sa mapu, CPN sieť (mapujúce neurónové siete), ďalej Daugmanovu sieť pre výpočet koeficientov neortogonálnej transformácie, neurónovú sieť ako prediktor pri DPCM a neurosieťový prístup k PCA (principal component analysis). Uvedené sú aj ich existujúce modifikácie.

13.2 ZÁKLADNÉ POZNATKY O NEURÓNOVÝCH SIEŤACH

Neurónová sieť je štruktúra pre paralelné distribuované spracovanie informácií v tvare orientovaného grafu s nasledujúcimi subdefiníciami a obmedzeniami [128]:

1. Uzly grafu sa nazývajú výkonné prvky (processing elements).
2. Hrany grafu sa nazývajú spojenia (connections).
3. Každý výkonný prvok môže obsahovať ľubovoľný počet vstupných spojení.
4. Každý výkonný prvok môže mať ľubovoľný počet výstupných spojení, ale signály pre ne musia byť rovnaké. V skutočnosti má každý prvok jediné výstupné spojenie, ktoré sa môže rozvetvovať a vytvoriť viacnásobné výstupné spojenia obsahujúce rovnaký signál.
5. Výkonné prvky môžu mať lokálnu pamäť.
6. Každý výkonný prvok obsahuje prenosovú funkciu, ktorá môže využiť lokálnu pamäť, vstupné signály a ktorá vytvorí výstupný signál výkonného prvku.
7. Vstupné signály prichádzajú do neurónovej siete cez spojenia, začiatok ktorých je vo vonkajšom prostredí. Výstupy neurónovej siete do vonkajšieho prostredia sú spojenia vychádzajúce von zo siete.

Neurónové siete sa môžu učiť zmenou vhodného súboru svojich parametrov. Môžu to byť ich prenosové funkcie, synaptické váhy, prahové funkcie, atď., prípadne zmenou počtu prvkov siete a ich konfigurácie [129]. Na najvšeobecnejšej úrovni môžeme rozoznať tri typy učenia (trénovania) [128]:

- učenie s učiteľom (supervised learning)
- ohodnotené (známkované) učenie (graded alebo reinforcement learning)
- samoorganizácia (self-organization).

Učenie s učiteľom predpokladá stav, keď sieť pracuje ako vstupno-výstupný systém. Počas učenia s učiteľom je do siete privádzaná postupnosť príkladov, ktoré sú správnymi (žiadanými) vstupno-výstupnými pármi. Sieti je teda exaktne povedané, aký výstup má emitovať. Aktuálny výstup siete však môže byť v istom zmysle iba "odhadom správneho výstupu. V mnohých prípadoch vstupno-výstupné páry použité počas učenia s učiteľom sú príkladmi danej funkcie f. Iným prípadom je stochastický vzťah medzi vstupom a výstupom. Ohodnotené (známkované) učenie je podobné učeniu s učiteľom, rozdielom však je, že namiesto privedenia správneho výstupu každého vstupno-výstupného páru sú priebežné výsledky učenia hodnotené (známkované) tak, že proces učenia vedie k žiadanému cieľu. Tento spôsob učenia je menej vhodný pre všeobecné aplikácie, jeho použitie je v oblasti riadenia a optimalizačných úloh.

Pri samoorganizácii sa sieť modifikuje reakciou na vstup sama. Nedostáva žiadne správne výstupy ani ohodnotenie. Táto kategória učenia môže vyzerať neužitočne, ale dajú sa ňou dosiahnuť pri spracovaní informácií prekvapujúce výsledky. Príkladom je samoorganizujúca sa mapa (self-organizing map).

Ako príklad uvádzame päť rozličných najznámejších kategórií zákonov učenia. Sú to: koincidenčné učenie (coincidence learning), účelové učenie (performance learning), súťažné učenie (competitive learning), učenie s filtráciou (filter learning), časopriestorové učenie (spatiotemporal learning). Tieto kategórie učenia spolu so zákonmi učenia patriacimi do týchto kategórií sú uvedené v [128] a [129].

13.3 MAPUJÚCE NEURÓNOVÉ SIETE

Mapujúce neurónové siete [128] riešia približnú implementáciu ohraničeného mapovania alebo funkcie $f: A \in R^n \rightarrow R^m$ z ohraničenej podmnožiny A n-rozmerného euklidovského priestoru do ohraničenej množiny $f[A]$ m-rozmerného euklidovského priestoru za pomoci učenia s príkladmi $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots$, kde $\mathbf{y}_k = f(\mathbf{x}_k)$. Pre naše účely predpokladáme, že tieto príklady mapovania sú

generované výberom vektorov $\mathbf{x}$ náhodne podľa danej hustoty pravdepodobnosti $\rho(\mathbf{x})$ (ρ je nulová mimo A). Tiež predpokladáme, že po učení bude sieť použitá pre vektory $\mathbf{x}$ náhodne vybraté podľa $\rho(\mathbf{x})$.

13.3.1 Meranie presnosti aproximácie funkcií

Predpokladajme, že mapujúca sieť sa adaptuje modifikovaním koeficientov (váh) a nie modifikáciou spojení. Keď je vektor $\mathbf{x}$ privedený do mapujúcej neurónovej siete, výstupný vektor označíme ako $\mathbf{Y}(\mathbf{x}, \mathbf{w})$, kde $\mathbf{w}$ je váhový vektor siete. Pre meranie presnosti siete potrebujeme porovnať aktuálny výstup siete $\mathbf{Y}(\mathbf{x}, \mathbf{w})$ so správnym výstupom $f(\mathbf{x})$ cez veľký počet testovacích vzoriek. Pre testovanie presnosti mapujúcej siete potrebujeme nové náhodne vybraté príklady $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_k, \mathbf{y}_k), ...$ odlišné od príkladov použitých pre učenie. Túto množinu príkladov pre testovanie nazývame testovacia množina (test set). Tieto príklady musia byť odlišné preto, lebo ak by sme použili rovnaké príklady pre učenie aj testovanie, zistili by sme iba ako dobre sa sieť naučila príklady pre učenie. Nás však zaujíma, ako dobre sa sieť naučila aproximovať funkciu pre ľubovoľné hodnoty $\mathbf{x}$. Predpokladáme, že testovacia množina je nekonečne veľká. Ak je daná mapovacia sieť a testovacia množina, sieť môže byť testovaná porovnaním jej výstupu so správnou hodnotou funkcie. Každé jednotlivé vyhodnotenie siete pre jeden príklad sa nazýva testovacia skúška (testing trial). Nech $(\mathbf{x}_k, \mathbf{y}_k)$ je príklad použitý pre k-tu testovaciu skúšku, t.j. $\mathbf{y}_k = f(\mathbf{x}_k)$. Opäť predpokladáme, že $\mathbf{x}_k$ je vybratý náhodne z množiny A podľa danej hustoty pravdepodobnosti ρ. Teraz môžeme definovať

$$F_k(\mathbf{x}_k, \mathbf{w}) = |f(\mathbf{x}_k) - \mathbf{Y}(\mathbf{x}_k, \mathbf{w}_k)|^2, \tag{13.1}$$

F_k je štvorec aproximačnej chyby mapujúcej siete pre k-tu testovaciu skúšku. Predpokladáme, že vektor $\mathbf{w}$ je počas testovania konštantný, váhy už nemajú možnosť adaptácie.

Stredná kvadratická odchýlka siete $F(\mathbf{w})$ (mean-square error MSE) je definovaná ako

$$F(\mathbf{w}) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} F_k(\mathbf{x}_k, \mathbf{w}) \tag{13.2}$$

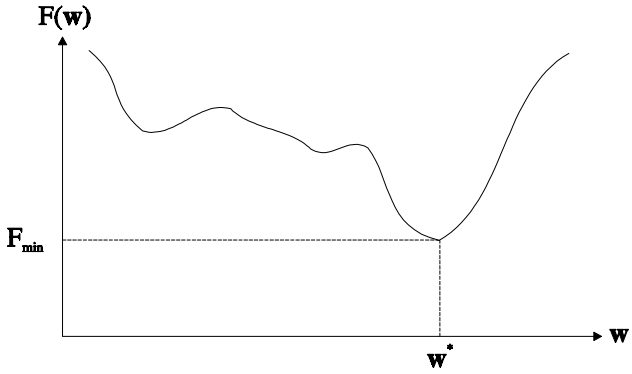
za predpokladu, že táto limita existuje. Alternatívne môžeme definovať $F(\mathbf{w})$ ako

$$F(\mathbf{w}) = \int_A |f(\mathbf{x}) - \mathbf{Y}(\mathbf{x}, \mathbf{w})|^2 \rho(\mathbf{x})\, d\mathbf{x}\,. \tag{13.3}$$

Stredná kvadratická odchýlka je dobre definovaná pre väčšinu neurónových sietí. Vlastnosťou $F(\mathbf{w})$ je, že pri náhodnom výbere testovacích príkladov podľa ρ nezáleží na tom, ktoré príklady použijeme. Limita v (13.2) bude takmer vždy konvergovať k rovnakej hodnote, ak je N veľké. Jedna z možností, ako určiť veľkosť testovacej množiny je skúšať stále väčšie testovacie množiny až pokiaľ stredná kvadratická odchýlka začne konvergovať k pevnej hodnote. Treba však poznamenať, že stredná kvadratická odchýlka ako meradlo správania sa siete nie je vhodné vždy. Ale praktické implementácie vhodnejších meradiel neexistujú. Preto je väčšina neurónových sietí zviazaná so strednou kvadratickou odchýlkou.

Ako je zrejmé, stredná kvadratická odchýlka $F(\mathbf{w})$ je funkciou váhového vektora $\mathbf{w}$ vyhodnocovanej neurónovej siete. $F(\mathbf{w})$ môžeme teda považovať za povrch nachádzajúci sa nad váhovým priestorom, F je výška povrchu pre váhový vektor $\mathbf{w}$. Tento povrch je známy ako chybový povrch (error surface, MSE surface) neurónovej siete. F je nezáporná funkcia, typický chybový povrch je na obr. 13.1. Je zrejmé, že cieľom je nájsť také váhy, ktoré minimalizujú F. Typickým prípadom je, že minimum F nie je nulové, pretože neurónová sieť nie je schopná presne implementovať žiadané mapovanie. Teda minimálna hodnota F je $F_{\min} > 0$. Ak môžeme nájsť váhový vektor $\mathbf{w}^*$, pre ktorý

$F(\mathbf{w}^*) = F_{\min}$, potom neurónová sieť najlepšie aproximuje mapovanie. Štruktúra chybového povrchu je rozhodujúcim faktorom pre siete so spätným šírením. Ale aj pre ostatné neurónové siete nám dáva možnosť pochopiť ich činnosť.

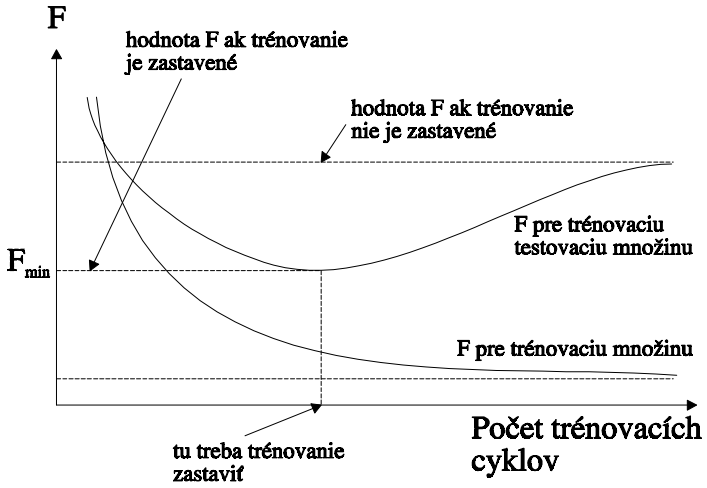


Obr. 13.1 Typický chybový povrch

13.3.2 Trénovanie a pretrénovanie

Jedným zo závažných problémov pri aproximácii funkcií je, že vo väčšine prípadov nemáme k dispozícii nekonečné množstvo trénovacích a testovacích príkladov. Ak by takéto množstvo bolo k dispozícii, sieť musí byť trénovaná čo najväčším počtom príkladov. Jednou z možností, ako si byť istý, že množiny sú dostatočne veľké, je ukázať, že postupné zvyšovanie veľkosti týchto množín už neovplyvňuje správanie sa siete. Inou cestou je testovať činnosť siete pre trénovaciu aj testovaciu množinu a presvedčiť sa, že tak pre trénovaciu ako aj testovaciu množinu sú výsledky rovnaké. Avšak takýto prípad je pomerne zriedkavý, a preto musíme riešiť problém, keď máme k dispozícii iba malé množstvo príkladov (na tomto mieste treba pripomenúť, že ak je množstvo údajov príliš malé, techniky neurónových sietí jednoducho nebudú pracovať). Najlepším postupom je začať vytvorením testovacej množiny. Tá musí obsahovať pokiaľ možno čo najviac príkladov, ktoré sa môžu vyskytnúť. Touto množinou potom testujeme činnosť siete (samozrejme, sieť na túto množinu nebola trénovaná, od siete požadujeme schopnosť generalizácie). Pretože takáto množina je použitá v úplne záverečnej fáze návrhu siete, nedáva sa k dispozícii vývojovým pracovníkom. Tí si musia vytvoriť svoju overovaciu testovaciu množinu (validation test set), trénovaciu množinu a trénovaciu testovaciu množinu (training test set). Trénovacou množinou sa sieť učí, trénovacou testovacou množinou sa skúma správanie siete a po dokončení návrhu je urobený overovací test (validation test) použitím overovacej množiny.

Nečakaným javom pre mapujúce siete založené na charakteristikách (hlavne pre siete so spätným šírením) je pretrénovanie (overtraining). Typický prípad možno vidieť na obrázku 13.2. Chyba siete, ak berieme do úvahy iba trénovaciu množinu, neustále klesá. To je bežný prípad pre takmer všetky mapujúce siete. Počas trénovania musíme učenie v periodických intervaloch zastavovať, váhy nechať konštantné a zmerať správanie sa siete pre trénovaciu testovaciu množinu. Z obr. 13.2 vidíme, že chyba pre trénovaciu testovaciu množinu spočiatku klesá, ale potom začne opäť stúpať. Je teda zrejmé (a vyznačené aj na obrázku), kedy treba trénovanie zastaviť. Pre siete, ktoré netrpia pretrénovaním (napr. CPN sieť, samoorganizujúca sa mapa) sa činnosť siete počas procesu učenia zlepšuje monotónne. Pre tieto siete nie je teda potrebné vytvárať trénovaciu testovaciu množinu. Najlepším postupom je jednoducho použiť pri trénovaní všetky dostupné údaje.



Obr. 13.2 Ilustrácia javu pretrénovania

Problematikou aproximácie funkcií mapujúcimi neurónovými sieťami (nelineárneho vstupno-výstupného mapovania) sa zaoberá napr. Kolmogorovova existenčná teoréma o mapujúcich neurónových sieťach [128], veta o spätnom šírení (backpropagation theorem) [128] a univerzálna aproximačná teoréma (universal approximation theorem) [127].

13.3.3 Prvok ADALINE, viacvrstvové dopredné siete, algoritmus spätného šírenia

Nezávisle od toho, či je neurónová sieť implementovaná do paralelného hardwaru alebo simulovaná na počítači, skladá sa z určitého počtu jednoduchých prvkov, ktoré spolupracujú na riešení problému. Základný stavebný blok mnohých neurónových sietí je prvok ADALINE [131] - adaptívny lineárny prvok (adaptive linear element). Adaline je adaptívny prahový prvok. Skladá sa z adaptívneho lineárneho kombinátora v kaskáde s prahovým zariadením, výstupom ktorého je binárny signál ± 1. Ak neurónová sieť obsahuje iba jeden prvok, na nastavenie váh sa používajú adaptívne algoritmy, napr. LMS algoritmus, pravidlo pre perceptrón (Perceptron rule). Ak prvok ADALINE odpovedá správne s veľkou pravdepodobnosťou na vstupné vzorky, ktoré neboli zahrnuté do trénovacej množiny, hovoríme, že nastala generalizácia. Učenie a generalizácia sú najužitočnejšie atribúty prvkov ADALINE. Pretože techniky spätného šírenia vyžadujú na výstupe prvku hladké nelinearity [131], definícia prvku ADALINE bola zovšeobecnená, funkcia signum v prvku ADALINE je nahradená sigmoidálnou nelinearitou (sigmoid nonlinearity). Sigmoidálny prvok ADALINE je znázornený na obr. 13.3. Pojem sigmoid sa vzťahuje na monotónne rastúcu funkciu v tvare S. Vstupno-výstupná charakteristika sigmoidu je označená $y_k = sgm(s_k)$. Typickou sigmoidálnou funkciou je hyperbolický tangens: $y_k = tgh(s_k)$.

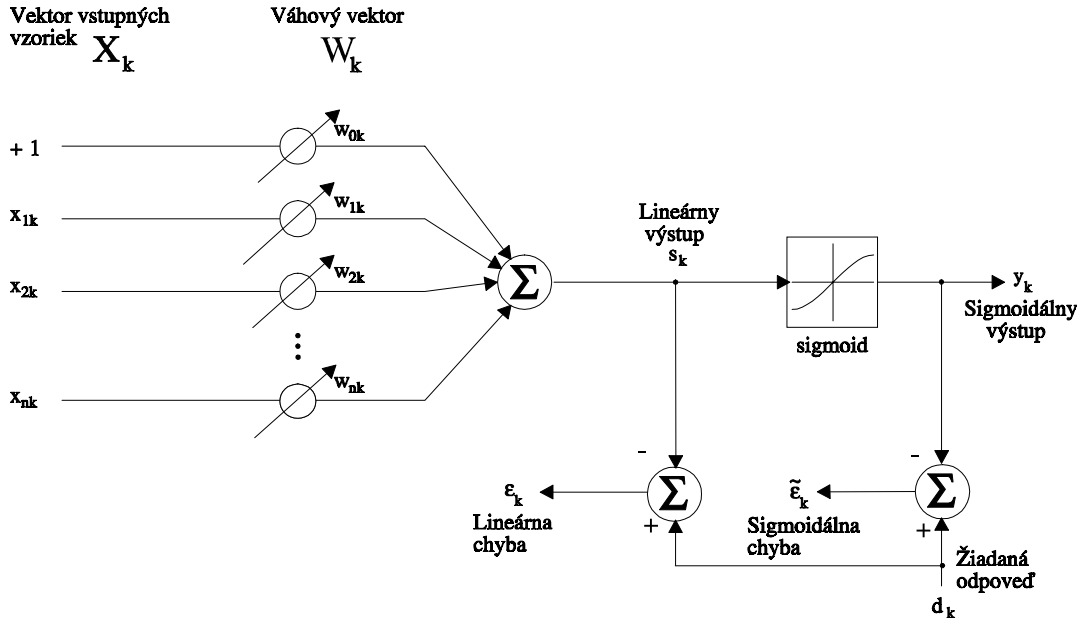
Dopredné siete (feedforward networks):

V súčasnosti majú neurónové siete viac vrstiev a zvyčajne všetky vrstvy sú adaptívne. Siete so spätným šírením, ktoré zaviedol Rumelhart a kol. [132] sú pravdepodobne najznámejším príkladom viacvrstvových sietí. Úplne prepojená trojvrstvová dopredná sieť je na obr. 13.4.

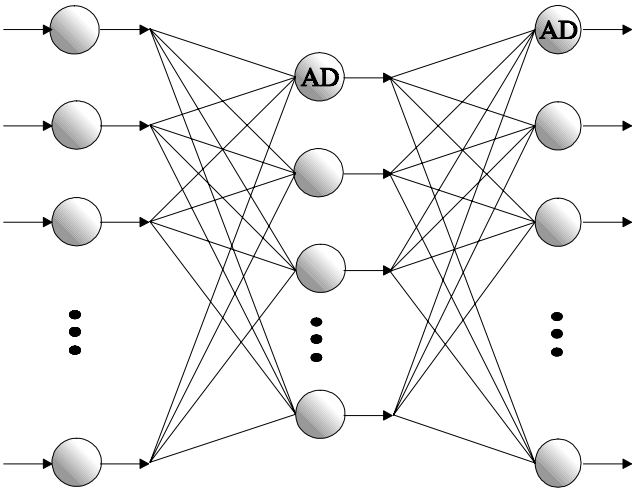
V prípade úplne prepojenej viacvrstvovej siete každý prvok ADALINE dostáva vstupy z každého výstupu predchádzajúcej vrstvy. Počas učenia je výstup každého prvku v sieti porovnaný s korešpondujúcou žiadanou odpoveďou. Chybové signály výstupných prvkov sa dajú určiť bez ťažkostí, hlavným problémom však je získanie chybových signálov prvkov v skrytých vrstvách. Algoritmus spätného šírenia rieši tento problém.

Iteratívne algoritmy pre trénovanie prvku ADALINE, prípadne z nich vytvorené siete rešpektujú jednoduchý princíp - princíp minimálneho narušenia (minimal disturbance principle) [131], ktorý sa dá sformulovať nasledovne:

Adaptuj tak, aby sa zredukovala výstupná chyba pre súčasnú vzorku a aby sa minimálne narušili už naučené vzorky.



Obr. 13.3 Prvok ADALINE so sigmoidálnou nelinearitou



Obr. 13.4 Trojvrstvová dopredná sieť

Existujú dva druhy algoritmov:

- pravidlá korigujúce chybu (error-correcting rules), ktoré menia váhy siete tak, aby sa vo výstupnej odpovedi skorigovala chyba na práve privádzanú vstupnú vzorku,
- gradientové pravidlá (gradient rules) menia váhy siete počas privádzania každej vzorky metódou gradientového zostupu (gradient descent) s cieľom redukovať strednú kvadratickú odchýlku (MSE) spriemerovanú cez všetky trénovacie vzorky.

Klasifikácia algoritmov pre obe triedy je uvedená v [131].

Algoritmus spätného šírenia patrí medzi gradientové pravidlá. Adaptácia siete gradientovými pravidlami (tiež nazývanými metódou najstrmšieho zostupu - method of steepest descent) začína s ľubovoľne inicializovaným váhovým vektorom $\mathbf{W}_0$ (tento vektor obsahuje všetky váhy siete). Meria sa gradient MSE a váhový vektor je zmenený v smere zodpovedajúcom záporu nameraného gradientu. Táto procedúra sa opakuje, MSE sa v priemere redukuje a váhový vektor nadobúda optimálnu hodnotu. Adaptácia váh teda znamená najstrmšie zostupovanie po chybovom povrchu k jeho minimu. Metóda najstrmšieho zostupu je popísaná vzťahom

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu\,(-\nabla_k)\,, \tag{13.4}$$

kde μ je parameter riadiaci stabilitu a rýchlosť konvergencie a ∇_k je hodnota gradientu v bode chybového povrchu korešpondujúcemu s $\mathbf{W} = \mathbf{W}_k$. Používa sa však okamžitý gradient $\hat{\nabla}_k$, lebo je priamo k dispozícii z jednej vzorky. Všeobecný gradient možno určiť len veľmi ťažko. Jeho počítanie by zahrnulo priemerovanie okamžitých gradientov súvisiacich so všetkými vzorkami trénovacej množiny. Toto je zvyčajne nepraktické a takmer vždy neefektívne.

Učenie pomocou spätného šírenia [127], [128], [129], [131], [132], [133] začína prezentovaním vektora vstupných vzoriek $\mathbf{X}$ do siete a jeho šírením v doprednom smere, výsledkom je výstupný vektor $\mathbf{Y}$. Počítajú sa chyby pre každý výstup. Ďalší krok zahŕňa šírenie účinku týchto chýb cez sieť v spätnom smere, pre každý prvok sa hľadá derivácia štvorca chyby δ, z každého δ sa počíta gradient a nakoniec sa upravia váhy každého prvku ADALINE podľa korešpondujúceho gradientu. Potom sa privedie nová vzorka a celý proces sa opakuje. Váhy sa inicializujú na malé náhodné hodnoty. Tento algoritmus nebude pracovať správne, ak sa váhy inicializujú na nulové alebo nevhodne zvolené hodnoty.

Existuje viacero variánt tohoto algoritmu. Veľmi známa je momentová technika (momentum technique) [131].

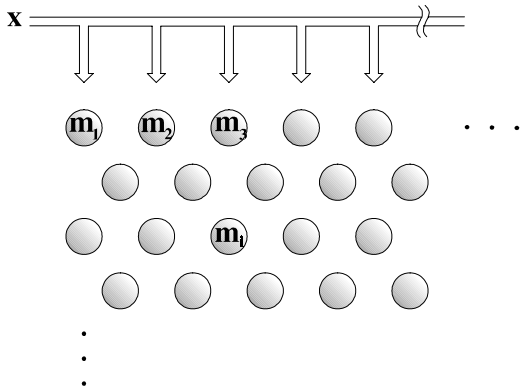
13.3.4 Samoorganizujúca sa mapa

Samoorganizujúca sa mapa (self-organizing map) [134], [128], [135] je mapujúca sieť, ktorá môže byť použitá ako jedna z možných techník návrhu katalógu vektorov (codebook) pri vektorovej kvantizácii.

Uvažujme dvojrozmernú sieť prvkov [134] zobrazených na obr. 13.5. Prvky môžu byť zoradené hexagonálne, pravouhlo a pod. Nech $\mathbf{x} = [x_1, x_2, ..., x_n]^T \in R^n$ je vstupný vektor, ktorý je spojený paralelne so všetkými prvkami i v tejto sieti. Váhový vektor prvku i označme $\mathbf{m}_i = [m_{i1}, m_{i2}, ..., m_{in}]^T \in R^n$.

Najjednoduchším analytickým meradlom pre zhodu $\mathbf{x}$ a $\mathbf{m}_i$ je súčin $\mathbf{x}^T\mathbf{m}_i$. V mnohých prípadoch je však vhodnejším meradlom kritérium založené na euklidovskej vzdialenosti medzi $\mathbf{x}$ a $\mathbf{m}_i$. Minimálna vzdialenosť definuje víťaza $\mathbf{m}_c$.

Je veľmi dôležité, že prvky realizujúce učenie nepracujú od seba nezávisle, ale ako topologicky vztiahnuté podmnožiny (topologically related subsets), ktorým je vnútená podobná forma korekcie. Je to vlastne priestorovo korelované učenie, pri ktorom majú váhové vektory tendenciu dosiahnuť hodnoty usporiadané (zoradené) pozdĺž osí siete.

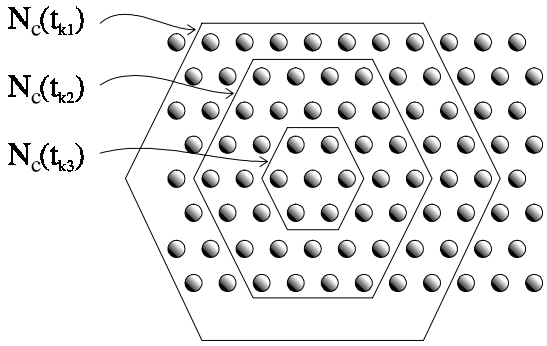


Obr. 13.5 Usporiadanie prvkov samoorganizujúcej sa mapy

Definuje sa preto okolie (neighborhood set) N_c okolo prvku c. Počas každého kroku učenia sú prvky patriace okoliu N_c upravované, zatiaľ čo prvky ležiace mimo okolia N_c ostanú nedotknuté. Okolie N_c je sústredené okolo prvku, ktorý je v najväčšej zhode so vstupom $\mathbf{x}$:

$$\|\mathbf{x} - \mathbf{m}_c\| = \min_i \{\|\mathbf{x} - \mathbf{m}_i\|\} \ . \tag{13.5}$$

Šírka (alebo polomer) okolia N_c je časovou premennou, okolie N_c je veľmi široké na začiatku a s časom sa monotónne zmenšuje. Vysvetlenie tohoto prístupu je nasledujúce: Začiatočné široké okolie dovolí mape približné, "hrubé globálne zoradenie siete, potom zmenšenie okolia zlepší priestorovú rozlíšiteľnosť mapy, pri ktorom však získané globálne zoradenie nie je narušené. To je znázornené na obr. 13.6.



Obr. 13.6 Príklady topologického okolia $N_c(k)$, kde $t_{k1} < t_{k2} < t_{k3}$

Dokonca je možné ukončiť proces učenia pre $N_c(k) = c$ (kde k je poradové číslo cyklu pri procese učenia); to znamená, že bude upravovaný iba prvok s najlepšou zhodou (víťaz). V tomto prípade ide o jednoduché súťažné učenie.

Proces adaptácie (v zápise pre poradové číslo cyklu k) je

$$\mathbf{m}_i(k+1) = \begin{cases} \mathbf{m}_i(k) + \alpha(k)[\mathbf{x}(k) - \mathbf{m}_i(k)] & \text{ak } i \in N_c(k) \\ \mathbf{m}_i(k) & \text{ak } i \notin N_c(k) \ , \end{cases} \tag{13.6}$$

kde $\alpha(k)$ je skalárny adaptačný zisk (scalar adaptation gain) $0 < \alpha(k) < 1$, $\alpha(k)$ by malo s časom klesať.

Alternatívny zápis využíva skalárnu funkciu (scalar kernel function) $h_{ci} = h_{ci}(k)$:

$$\mathbf{m}_i(k+1) = \mathbf{m}_i(k) + h_{ci}(k)[\mathbf{x}(k) - \mathbf{m}_i(k)] \,, \tag{13.7}$$

kde $h_{ci}(k) = \alpha(k)$ vnútri okolia $N_c(k)$ a $h_{ci}(k) = 0$ mimo okolia N_c. Definícia môže ale byť omnoho všeobecnejšia. Ak označíme súradnice prvkov c a i vektormi $\mathbf{r}_c$ a $\mathbf{r}_i$, potom vhodný tvar pre h_{ci} môže byť

$$h_{ci} = h_0 \exp\left(-\|\mathbf{r}_i - \mathbf{r}_c\|^2/\beta^2\right) \,, \tag{13.8}$$

kde $h_0 = h_0(k)$ a $\beta = \beta(k)$ sú vhodné klesajúce funkcie času.

Všeobecné pravidlá pre učenie mapy sú uvedené v [134].

Proces adaptácie môže začať náhodným výberom inicializačných váh $\mathbf{m}_i(0)$, jediným obmedzením je, že by mali byť rôzne.

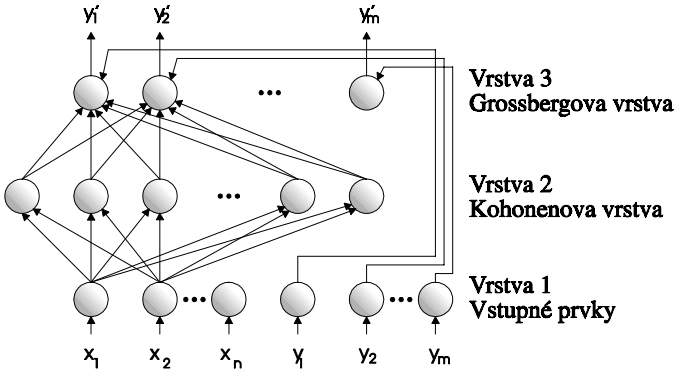
Pretože učenie je stochastický proces, konečná štatistická presnosť mapovania závisí od počtu krokov, ktorý musí byť dostatočne veľký, táto požiadavka sa nijako nedá obísť. Istým empirickým pravidlom je, že pre dobrú štatistickú presnosť by mal byť počet krokov aspoň 500 krát väčší ako počet prvkov siete. Počet prvkov vektora $\mathbf{x}$ nemá vplyv na počet iteračných krokov, povolený je veľký počet prvkov vektora $\mathbf{x}$. Ak je k dispozícii iba malý počet vzoriek, musia byť opakované, aby sa splnil požadovaný počet krokov.

Pre približne prvých 1000 krokov môže byť hodnota $\alpha(k)$ blízko 1, potom musí monotónne klesať. Presné pravidlo nie je dôležité, $\alpha = \alpha(k)$ môže byť lineárnou alebo exponenciálnou funkciou, alebo môže byť nepriamo úmerné k. Príkladom je $\alpha(k) = 0.9\,(1 - k/1000)$. Počas tejto úvodnej fázy prebehne zoraďovanie $\mathbf{m}_i$, ostatné kroky sú potrebné pre jemné vylepšenie mapy. Po procese zoradenia môže $\alpha = \alpha(k)$ nadobúdať malé hodnoty (napr. 0,01) počas dlhej doby učenia.

Mimoriadna pozornosť musí byť venovaná výberu $N_c = N_c(k)$. Ak je okolie na začiatku učenia príliš malé, mapa nebude zoradená globálne. Namiesto toho bude rôzne "rozparcelovaná. Tomuto javu môžeme predísť, ak na začiatku bude okolie $N_c = N_c(0)$ dostatočne široké, a potom sa s časom zmenšuje. Začiatočný polomer N_c môže byť dokonca väčší ako polovica priemeru siete. Počas približne prvých 1000 krokov, keď nastáva správne zoradenie siete a $\alpha = \alpha(k)$je dostatočne veľké, sa môže polomer N_c zmenšovať lineárne, až bude obsahovať iba jeden prvok mapy. Počas procesu jemného vylepšenia môže N_c ešte stále obsahovať najbližšie susedné prvky prvku c.

13.3.5 CPN sieť (counterpropagation network)

Kombináciou Kohonenovej vrstvy (typ učenia pre ňu patrí do súťažného učenia) a Grossbergovho učenia (patrí do učenia s filtráciou) dostaneme nový typ siete [128]. Táto sieť pracuje ako štatisticky optimálna samoprogramovacia vyhľadávacia tabuľka (lookup table). Nazýva sa CPN sieť - sieť s protismerným (ústretovým) šírením (counterpropagation network) [128], [136], [137]. Budeme sa zaoberať dopredným (forward-only) variantom CPN siete znázorneným na obr 13.7.



Obr. 13.7 Dopredná CPN sieť

Táto sieť sa skladá z troch vrstiev: zo vstupnej vrstvy (vrstva 1) obsahujúcej n prvkov, ktoré distribuujú vstupné signály $x_1, x_2, \ldots, x_n$ (a m prvkov privádzajúcich správne hodnoty signálov $y_1, y_2, \ldots, y_m$ do výstupnej vrstvy), z Kohonenovej vrstvy (vrstva 2) s N prvkami, ktorých výstupné signály sú $z_1, z_2, \ldots, z_N$ a z Grossbergovej vrstvy (vrstva 3) reprezentujúcej aproximácie prvkov $y_1, y_2, \ldots, y_m$ vektora $\mathbf{y} = f(\mathbf{x})$. Počas učenia sú tieto správne hodnoty privádzané do vrstvy 3. Počas učenia sa vyberie $\mathbf{x}_k$, určí sa $\mathbf{y}_k = f(\mathbf{x}_k)$ a oba vektory sú privedené do siete. Rovnice pre prenosovú funkciu prvkov vo vrstve 2 sú

$$z_i = \begin{cases} 1 & \text{ak } i \text{ je najmenšie prirodzené číslo, pre ktoré} \\ & D(\mathbf{w}_i^{\text{staré}}, \mathbf{x}) \le D\left(\mathbf{w}_j^{staré}, \mathbf{x}\right) \text{ pre všetky } j \\ 0 & \text{inak ,} \end{cases} \tag{13.9}$$

kde D je euklidovská vzdialenosť.

Po skončení súťažného procesu je ďalším krokom úprava váh Kohonenovým učením: Výkonný prvok, ktorý zvíťazil v súťaži o úpravu váh si upraví svoj váhový vektor podľa rovnice

$$\mathbf{w}_i^{\text{nové}} = (1 - \alpha(t))\, \mathbf{w}_i^{\text{staré}} + \alpha(t)\, \mathbf{x} \,. \tag{13.10}$$

Iné výkonné prvky si váhy neupravujú. Podobne ako pri samoorganizujúcej sa mape je $\alpha = \alpha(t)$ funkciou času. Začiatočná hodnota je vysoká (napr. 0,8) a postupne s časom klesá.

Po skončení činnosti vrstvy 2 začína pracovať vrstva 3. Tá dostáva signály z vrstvy 2 (jeden z nich je 1, ostatné 0). Každý výkonný prvok vrstvy 3 dostáva všetky prvky vektora $\mathbf{z}$. Výkonné prvky vrstvy 3 sa riadia nasledujúcimi rovnicami (Grossbergovo učenie)

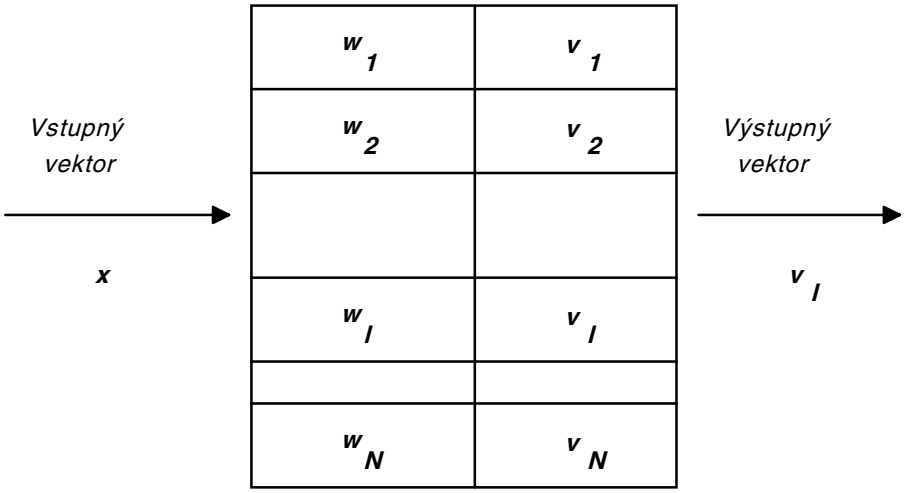
$$y_j' = \sum_{i=1}^{N} u_{ji}^{\text{staré}}\, z_i \,, \tag{13.11}$$

$$u_{ji}^{\text{nové}} = u_{ji}^{\text{staré}} + a\, (-u_{ji}^{\text{staré}} + y_j)\, z_i \,, \tag{13.12}$$

$\mathbf{u}_j = (u_{j1}, u_{j2}, \ldots, u_{jN})$ je váhový vektor korešpondujúci s j-tym výkonným prvkom vrstvy 3 a a je rýchlosť učenia pre Grossbergovo učenie ($0 < a < 1$). Rovnica prenosovej funkcie (13.11) slúži na vybratie váhy korešpondujúcej so vstupom z víťazného prvku vrstvy 2 (pre ktorý $z_i = 1$) a emitovanie hodnoty tejto váhy ako výstup y_j' výkonného prvku. Čiže po dostatočne dlhom učení je výstupom siete

vektor $\mathbf{v}_i = (u_{1i}, u_{2i}, \ldots, u_{mi})$, ak výkonný prvok i vyhral súťažný proces vrstvy 2. Vektor $\mathbf{v}_i$ je veľmi blízky správnemu vektoru **y** zodpovedajúcemu vektoru **x**.

Po skončení učenia sieť funguje ako vyhľadávacia tabuľka uvedená na obr. 13.8. Vstupný vektor **x** je porovnávaný s váhovými vektormi vrstvy 2, aby sa našiel vektor najväčšej zhody $\mathbf{w}_l$ (použitím euklidovskej vzdialenosti D). Sieť emituje výstupný vektor $\mathbf{v}_l$ vrstvy 3 korešpondujúci s vektorom $\mathbf{w}_l$. Presne toto je funkciou vyhľadávacej tabuľky.



Obr. 13.8 CPN sieť ako vyhľadávacia tabuľka

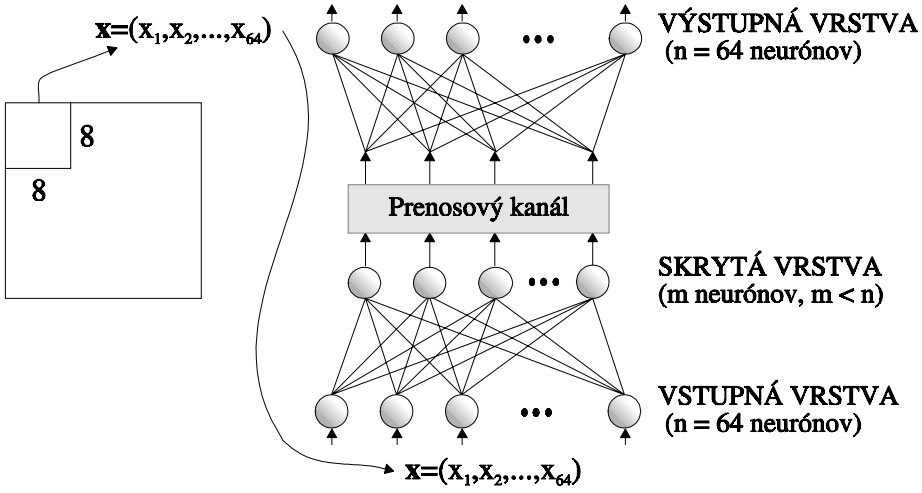
CPN sieť môže pracovať aj v tzv. interpolatívnom móde [128], [136]. V tomto móde je povolené viac ako jednému prvku vrstvy 2 vyhrať súťažný proces. Ak sú výstupy týchto viacerých prvkov nastavené tak, že ich súčet je 1, sieť bude emitovať svoj zvyčajný výstup **y**'. Interpolačný proces môže viesť k značne zvýšenej presnosti aproximácie mapovania bez zvyšovania veľkosti siete.

13.4 VYUŽITIE MAPUJÚCICH NEURÓNOVÝCH SIETÍ PRE KOMPRESIU OBRAZU

13.4.1 Kompresia obrazu využívajúca siete so spätným šírením

Kompresia obrazu pomocou sietí so spätným šírením je známa ako Cottrellova-Munrova-Zipserova technika (Cottrell/Munro/Zipser technique) [139], [128]. Systém pre kompresiu obrazu je uvedený na obr. 13.9.

Obraz je rozdelený na bloky rozmeru 8x8 obrazových prvkov, každý blok teda obsahuje 64 obrazových prvkov. Každý obrazový prvok je reprezentovaný (kódovaný) ako 8-bitové číslo x_i, čo zodpovedá 256 úrovniam stupnice šedej. Rasterizáciou je 64 obrazových prvkov sformovaných do vektora **x**. V spomenutej technike bola použitá trojvrstvová sieť so spätným šírením, ktorá pozostávala zo 64 vstupných prvkov, 64 výstupných prvkov a 16 prvkov v skrytej vrstve (na obr. 13.9 m = 16). Vstupmi do siete sú prvky vektora **x**. Výstupmi siete je 64 hodnôt, ktoré budú, ak je všetko v poriadku, veľmi blízke hodnotám prvkov vektora **x**. 16 prvkov skrytej vrstvy je zodpovedných za vyjadrenie 64 vstupných hodnôt nejakým spôsobom iba 16 hodnotami, z ktorých môže byť 64 pôvodných hodnôt rekonštruovaných vo výstupnej vrstve. Úlohou vstupnej a skrytej vrstvy je teda mapovanie 64-rozmerného priestoru do m-rozmerného priestoru (kde m < 64), pričom sa minimalizuje MSE pre výstupnú vrstvu [145].



Obr. 13.9 Systém pre kompresiu obrazu využívajúci sieť so spätným šírením

Použitie systému pre kompresiu obrazu je teda nasledovné: Vstupná a skrytá vrstva sa nachádza na vysielacej strane. Po privedení bloku 64 obrazových prvkov do siete vytvorí skrytá vrstva súbor 16 čísel, ktoré sú tiež kvantované 8 bitmi. Týchto 16 čísel je prenesených cez prenosový kanál. Výstupná vrstva siete sa nachádza na prijímacej strane, 16 výstupných hodnôt skrytej vrstvy je privedených do výstupnej vrstvy, aby sa zrekonštruovalo 64 hodnôt obrazového bloku. Kompresný pomer je v tomto prípade 4 (namiesto prenesenia 64 bytov stačí preniesť iba 16 bytov). Samozrejme, výstup siete nebude identický so vstupom, ale činnosť systému je dobrá.

Pri učení siete sa používajú obrazy, ktoré štatisticky reprezentujú obrazy, o ktorých sa predpokladá, že budú komprimované.

Cottrell, Munro a Zipser uvádzajú, že subjektívna úroveň kvality obrazov, ktoré sú výsledkom ich techniky, je porovnateľná s inými spôsobmi kompresie obrazu. Iným ich zaujímavým poznatkom je, že činnosť systému sa zhoršuje pre obrazy, ktoré sú štatisticky nekonzistentné s obrazmi použitými pre trénovanie. Ak použili pre trénovanie obrazy ľudí v interiéri a na testovanie obrazy automobilov, pozorovali výrazné zhoršenie kvality testovaných obrazov.

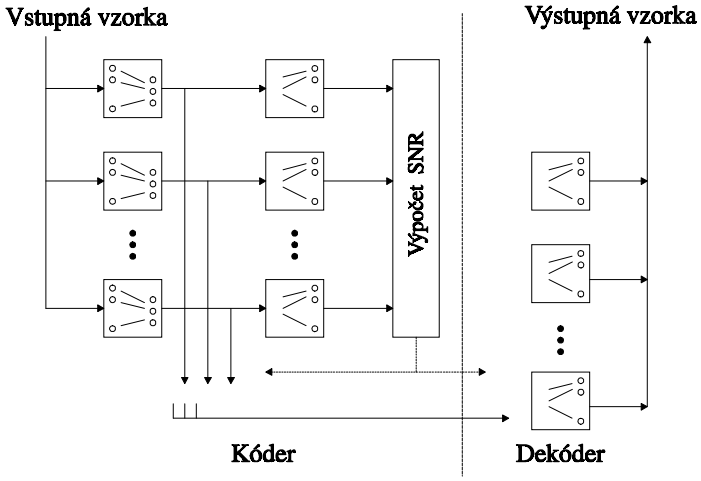
Uvedená konfigurácia pre bloky 8 x 8 a 16 prvkov skrytej vrstvy je typická, samozrejme je možné použiť aj iné konfigurácie.

Existuje niekoľko zhodnotení a modifikácií metódy kompresie obrazu využívajúcej sieť so spätným šírením, napr. [141], [142], [145], [146], [147], [164], [165], [166].

Zaujímavá paralelná štruktúra založená na sieťach so spätným šírením je prezentovaná v [143] (a tiež v [144]). Táto štruktúra je na obr. 13.10. Pozostáva z viacerých sietí, ktoré majú zvyšujúci sa počet prvkov v skrytých vrstvách. Všetky siete sú trénované rovnakým súborom obrazov. Počas testovania je každá vzorka súčasne privedená do všetkých sietí. Vyberie sa tá sieť s najmenším počtom prvkov skrytej vrstvy, pre ktorú je hodnota SNR vyššia ako preddefinovaná prahová hodnota θ. Teda je zaručené, že rekonštruovaný obraz bude mať hodnotu SNR vždy vyššiu, ako θ. Využíva sa skutočnosť, že hladké bloky vybraté z obrazu sa kódujú ľahšie, môžu byť teda spracované sieťou s malým počtom prvkov skrytej vrstvy. Obrazy boli tiež rozdelené do blokov rozmeru 8 x 8, skrytá vrstva obsahovala od 3 do 64 prvkov. Výsledky takejto štruktúry prevyšujú výsledky dosahované jednoduchou štruktúrou uvedenou vyššie. V [144] tí istí autori publikujú kombináciu tejto paralelnej štruktúry a techniky subpásmovej filtrácie.

Uvedieme aj výsledky našich simulácií pre kompresiu obrazu Cottrellovou - Munrovou - Zipserovou technikou.

Pri našich simuláciách bolo dôsledne dodržané pravidlo, aby testovacie obrazy boli odlišné od obrazov použitých ako trénovacia množina (s účelom demonštrovať reálne možnosti neurónových



Obr. 13.10 Paralelná štruktúra sietí so spätným šírením

sietí a nie iba najlepší možný prípad, keď testovací obraz je zároveň aj súčasťou trénovacej množiny). Takýmto spôsobom sú tiež ilustrované dva dôležité atribúty nerónových sietí - učenie a generalizácia. Pre trénovanie sietí so spätným šírením bol použitý súbor 4096 neprekrývajúcich sa 8-bitových vzoriek rozmeru 8 x 8 vybratých zo štyroch obrazov rozmerov 256 x 256. Testovacím obrazom bol iný obraz rozmeru 256 x 256, vybratý z testovacích obrazov. Tento obraz bol tiež rozdelený na neprekrývajúce sa bloky 8 x 8.

Konfiguráciu siete sme označili n-m-n, t. j. sieť s n vstupnými prvkami, m prvkami skrytej vrstvy (v súlade s obr. 13.9) a n výstupnými prvkami. Prvky skrytej a výstupnej vrstvy obsahujú nelinearitu. Výber bloku rozmerov 8 x 8 a tomu zodpovedajúce n = 64 bol urobený z čisto konvenčných dôvodov - je použitý aj v prácach [128], [139], [140], [142], [145], [147] atď. Výstupy skrytej vrstvy sú kvantované 8 bitmi. Pri ukončení trénovania sme brali do úvahy aj jav pretrénovania, ktorým siete so spätným šírením trpia.

Výsledky simulácií sú uvedené pre výstupy skrytej vrstvy kvantované 8 bitmi. Pre sieť 64-12-64 je v tomto prípade kompresia 1,5 bit/bod, pre sieť 64-8-64 je 1 bit/bod, pre sieť 64-6-64 je 0,75 bit/bod a pre sieť 64-2-64 je 0,25 bit/bod. Rekonštruované obrazy LENA 256 x 256 pre tieto štyri rôzne konfigurácie siete so spätným šírením sú uvedené na obr. 13.15.

Samozrejme, výstupné hodnoty skrytej vrstvy môžu byť kvantované aj iným počtom úrovní ako 256. To znamená, že pre kvantovanie výstupov skrytej vrstvy rôznych konfigurácií sietí so spätným šírením je možné použiť rôzny počet bitov.

Pripomíname, že na výstupné hodnoty skrytej vrstvy je možné aplikovať entropické kódovanie, ktoré vedie k zvýšeniu kompresného pomeru.

13.4.2 Kompresia obrazu využívajúca samoorganizujúcu sa mapu

Samoorganizujúca sa mapa použitá pre kompresiu obrazu je určená pre vytvorenie katalógu vektorov pre vektorovú kvantizáciu.

Existuje viacero metód pre vytvorenie kódovacej tabuľky [71], [148]. Porovnanie niekoľkých metód vytvorenia kódovacej tabuľky je uvedené v [149]. Metódy sú však výpočtovo veľmi náročné, takže je veľmi ťažké vytvoriť systémy pracujúce v reálnom čase. Sú algoritmické, a preto vhodnejšie pre von Neumannovské počítanie než pre paralelné distribuované spracovanie informácií [150]. Neurónové siete sú nealgoritmické a majú paralelnú distribuovanú štruktúru, a preto sú vhodné pre kompresiu údajov. Samoorganizujúca sa mapa použitá pre vektorovú kvantizáciu sa vyznačuje rýchlym učením. Vytvorenie kódovacej tabuľky trvá kratšie, ako klasické návrhy [150].

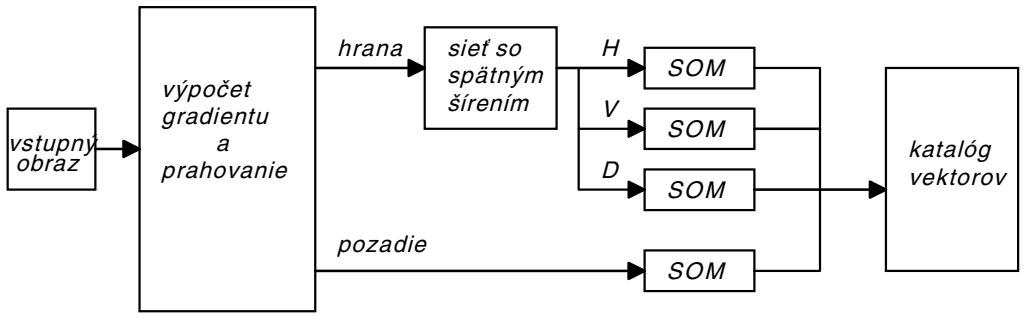
Opäť pre ilustráciu uvádzame výsledky našich simulácií. Počítačové simulácie pre Kohonenovu samoorganizujúcu sa mapu boli založené na princípoch uvedených v kapitole 13.3.4. Použili sme pravouhlé usporiadanie prvkov mapy. Začiatočný polomer okolia N_c bol približne polovica priemeru siete. Tvar okolia bol zvolený podľa vzťahu 13.8. Použité siete boli trénované štyrmi obrazmi identickými s obrazmi použitými pre trénovanie sietí so spätným šírením. V tomto prípade však boli obrazy delené do neprekrývajúcich sa blokov rozmeru 4 x 4. Pripomíname, že 16 váh každého prvku reprezentuje jeden vektor z katalógu vektorov. Pre testovanie bol opäť použitý obraz LENA 256 x 256, tiež rozdelený do neprekrývajúcich sa blokov rozmerov 4 x 4.

Uvádzame štyri výsledky simulácií: mapu so 4096 prvkami - bitová náročnosť R = 0,75 bit/bod, mapu s 1024 prvkami - R=0,625 bit/bod, mapu s 256 prvkami - R = 0,5 bit/bod a mapu so 64 prvkami - R=0,375 bit/bod. Rekonštruované obrazy LENA 256 x 256 pre tieto hodnoty kompresie sú na obr. 13.16.

Pripomíname, že zvýšenie kompresného pomeru sa dá dosiahnuť aplikáciou entropického kódovania na postupnosť indexov vektorov najväčšej zhody z katalógu vektorov, prenášaných namiesto blokov pôvodného obrazu.

Návrh kódovacej tabuľky neurónovou sieťou s modifikovaným Kohonenovym algoritmom je uvedený v [151] s ďalším rozpracovaním v [152]. Popisuje sa metóda kódovania obrazov vizuálnymi obrazcami, ktorá patrí do metód založených na modeloch ľudského vnímania, umožňujúcich dosiahnuť väčšiu kompresiu údajov a subjektívne lepšiu kvalitu rekonštruovaných obrazov.

Zaujímavá metóda kompresie obrazu kombinujúca sieť so spätným šírením (použitú ako klasifikátor) a Kohonenovu samoorganizujúcu sa mapu je uvedená v [153]. Používa samostatné katalógy vektorov pre bloky s hranami a pre pozadie (integrita hrán hrá dôležitú úlohu vo vizuálnom vnímaní). Navrhnutá schéma z [153] je zobrazená na obr. 13.11. Vstupný blok obrazu je najprv klasifikovaný do dvoch tried: hrana a pozadie. Dosahuje sa to výpočtom gradientu pre každý pixel bloku. Veľkosť bloku je 4 x 4 a pre výpočet gradientu pre daný pixel je použitý subblok o veľkosti 3 x 3 centrovaný na danom pixeli. Výsledkom je binárny blok, v ktorom je vysoko aktívnemu pixelu priradená hodnota 1 a nízko aktívnemu pixelu hodnota 0. Ak je počet nenulových pixelov väčší ako prah, blok bude klasifikovaný ako blok s hranou. Aby sa zachovala orientácia hrán, každý blok s hranou je ďalej klasifikovaný do tried s rôznou orientáciou. Definované boli 3 triedy: horizontálna (H), vertikálna (V) a diagonálna (D). Sieť so spätným šírením je tu použitá ako klasifikátor binárnych blokov do rôznych podtried. Jej konfigurácia je 16 prvkov vstupnej vrstvy, jedna skrytá vrstva a 12 prvkov výstupnej vrstvy (sú definované 3 podtriedy pre bloky s horizontálnou hranou, 3 podtriedy pre bloky s vertikálnou hranou a 6 podtried pre bloky s diagonálnou hranou). Po takomto postupe je vstupná vzorka klasifikovaná do 4 podmnožín: pozadie, blok s horizontálnou, vertikálnou a diagonálnou hranou. Každá podmnožina trénovacích údajov je vstupom zodpovedajúcej samoorganizujúcej sa mapy (SOM), výsledkom čoho sú 4 rozličné katalógy vektorov. Počet vektorov každej kódovacej tabuľky je určený vopred. To umožní priradiť dostatočný počet vektorov kódovacej tabuľky pre bloky s hranou a vylepšiť tak vizuálnu kvalitu zachovaním integrity hrán.



Legenda
SOM - samoorganizujúca sa mapa
H,V,D - blok s hranou - horizontálnou, vertikálnou a diagonálnou

Obr. 13.11 Štruktúra pre kompresiu obrazu kombinujúca sieť so spätným šírením a samoor ganizujúcu sa mapu

13.4.3 Kompresia obrazu využívajúca CPN sieť

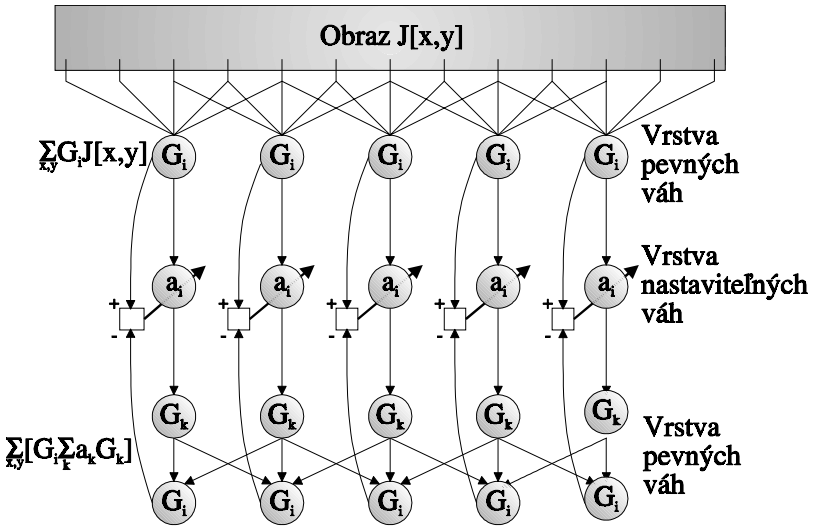
CPN sieť pre kompresiu obrazu je tiež založená na vektorovej kvantizácii. Priestor všetkých vektorov **x** sa použitím CPN siete s 2^N prvkami v Kohonenovej vrstve rozdelí na 2^N rovnako pravdepodobných disjunktných podmnožín (označenie súhlasí s kapitolou 13.3.5). Každá takáto podmnožina sa označí indexom i, ktorý leží medzi 0 a $2^N - 1$. Inými slovami, každému váhovému vektoru Kohonenovej vrstvy sa priradí N-bitové číslo rovné indexu zodpovedajúceho prvku Kohonenovej vrstvy mínus 1. Index i sa prenesie cez prenosový kanál, na prijímacej strane dekóder obsahujúci tabuľku vektorov $\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{2^N-1}$ emituje vektor $\mathbf{w}_i$ ako aproximáciu pôvodného vektora **x**. Teda chyba systému je rovná vzdialenosti vektorov **x** a $\mathbf{w}_i$.

13.5 DVOJROZMERNÁ DISKRÉTNA GABOROVA TRANSFORMÁCIA A JEJ VYUŽITIE PRE KOMPRESIU OBRAZU

Dennis Gabor navrhol nový spôsob analýzy ľubovoľného signálu [154], zaviedol optimálnu množinu bázových funkcií zloženú zo sínusových funkcií času násobených Gaussovskou funkciou času. Takéto Gaussovsky váhované sínusoidy boli nazvané logony (logons). Gabor ukázal, že pre časovo ohraničený signál použitie takýchto bázových funkcií minimalizuje neurčitosť vztiahnutú na súčin efektívneho trvania signálu v čase a jeho efektívnej šírky pásma. Žiadna iná množina bázových funkcií nemá takúto vlastnosť.

John Daugman na základe poznatkov neurofyziológie vyvinul dvojrozmernú verziu logonov [155]. V [155] je tiež popísaná trojvrstvová sieť pre transformáciu dvojrozmerného diskrétneho signálu do takejto zovšeobecnenej dvojrozmernej Gaborovej reprezentácie pre analýzu, segmentáciu a kompresiu obrazu.

Všeobecná architektúra neurónovej siete pre nájdenie koeficientov pre transformáciu signálu (ktorá nemusí byť ortogonálna ani kompletná) je uvedená na obr.13.12.



Obr. 13.12 Trojvrstvová neurónová sieť pre nájdenie optimálnych koeficientov ľubovoľnej transformácie (vo všeobecnosti neortogonálnej a nekompletnej)

Uvažujme diskrétny dvojrozmerný signál $\mathbf{J}[x,y]$, napr. obraz rozmerov 256x256 pixelov [x,y], ktorý chceme analyzovať alebo komprimovať pomocou množiny koeficientov $\{a_i\}$ nejakej množiny dvojrozmerných elementárnych funkcií $\{\mathbf{G}_i[x,y]\}$. Daný obraz $\mathbf{J}[x,y]$ môžeme považovať za vektor v

65536-rozmernom vektorovom priestore. Snažíme sa teda reprezentovať $\mathbf{J}[x,y]$ buď presne alebo v optimálnom zmysle prostredníctvom vybranej množiny vektorov $\mathbf{G}_i[x,y]$. Toto si vyžaduje nájsť také koeficienty $\{a_i\}$, že výsledný vektor $\mathbf{H}[x,y]$

$$\mathbf{H}[x,y] = \sum_{i=1}^{n} a_i \, \mathbf{G}_i[x,y] \tag{13.13}$$

je buď identický s $\mathbf{J}[x,y]$ (kompletný prípad) alebo generuje rozdielový vektor $\mathbf{J}[x,y] - \mathbf{H}[x,y]$ minimálnej veľkosti (optimalizačný prípad). Ak elementárne funkcie $\{\mathbf{G}_i[x,y]\}$ tvoria kompletnú ortogonálnu množinu, potom reprezentácia $\mathbf{H}[x,y]$ je presná (rozdielový vektor je nulový) a riešenie pre $\{a_i\}$ je jednoduché:

$$a_i = \frac{\sum_{x,y} (\mathbf{G}_i[x,y] \, \mathbf{J}[x,y])}{\sum_{x,y} \mathbf{G}_i^2[x,y]} \, . \tag{13.14}$$

Ak elementárne funkcie $\{\mathbf{G}_i[x,y]\}$ netvoria kompletnú ortogonálnu množinu, potom vo všeobecnosti reprezentácia $\mathbf{H}[x,y]$ bude nepresná a žiadaná množina koeficientov $\{a_i\}$ sa musí určiť pomocou optimalizačného kritéria, ako napr. minimalizáciou štvorca veľkosti rozdielového vektora:

$$E = \|\mathbf{J}[x,y] - \mathbf{H}[x,y]\|^2 = \sum_{x,y} (\mathbf{J}[x,y] - \mathbf{H}[x,y])^2 \, . \tag{13.15}$$

Účelová funkcia rozdielového vektora (difference-vector cost function) (13.15) je kvadratická pre každý prvok z množiny $\{a_i\}$, a teda existuje jediné globálne minimum pre E. Neurónová sieť uvedená na obr. 13.12 iteráciami konverguje k žiadanej reprezentácii obrazu (prostredníctvom množiny $\{a_i\}$) implementáciou gradientového zostupu (gradient descent) po rozdielovom povrchu $E(a_i)$, ktorý reprezentuje závislosti kvadratickej účelovej funkcie pre všetky koeficienty z množiny $\{a_i\}$.

Neurónová sieť uvedená na obr. 13.12 začína vrstvou s pevnými váhami, ktoré sú dané ľubovoľnou množinou (vo všeobecnosti neortogonálnou) elementárnych funkcií $\{\mathbf{G}_i[x,y]\}$. Sumáciou jednotlivých pixelov cez tieto váhy je výstupom i-teho neurónu tejto vrstvy skalárny súčin i-tej elementárnej funkcie $\mathbf{G}_i[x,y]$ so vstupným obrazom $\mathbf{J}[x,y]$ v danom regióne. Druhá vrstva obsahuje nastaviteľné váhy pre násobenie každého z týchto výstupov podľa riadiaceho signálu vznikajúceho z interlaminárnych interakcií. Tretia vrstva je identická s prvou vrstvou a reprezentuje tú istú množinu elementárnych funkcií. Nastaviteľné váhy prostrednej vrstvy tvoria transformovanú reprezentáciu obrazu prostredníctvom množiny koeficientov $\{a_i\}$. Adaptívny riadiaci signál upravuje každú váhu o hodnotu Δ_i, ktorá je daná rozdielom dopredného a spätnoväzbového signálu. Dopredný signál reprezentuje úroveň aktivity neurónu prvej vrstvy, spätnoväzbový signál je skalárnym súčinom váhovej funkcie zodpovedajúceho neurónu tretej vrstvy a váhovanou sumou všetkých ostatných susedných neurónov tejto vrstvy, s ktorými je spojený. Teda úprava váhy je

$$\Delta_i = \sum_{x,y} (\mathbf{G}_i[x,y] \, \mathbf{J}[x,y]) - \sum_{x,y} \left[\mathbf{G}_i[x,y] \left(\sum_{k=1}^{n} a_k \, \mathbf{G}_k[x,y] \right) \right] \tag{13.16}$$

a iteratívne pravidlo pre úpravu každého koeficientu je

$$a_i \Rightarrow a_i + \Delta_i \, . \tag{13.17}$$

Táto neurónová sieť nepotrebuje učiteľa, ktorý generuje signál pre úpravu váh porovnávaním aktuálnej reprezentácie so žiadanou vzorkou. Namiesto toho adaptívny riadiaci signál Δ_i vzniká iba z interlaminárnych interakcií. Úprava váh je vždy v smere dolu po účelovom povrchu (cost surface) $E(a_i)$ a úprava je úmerná strmosti účelového povrchu v tomto bode. Rovnovážny stav siete, ktorý sa dosiahne keď všetky $\Delta_i = 0$ je stavom, v ktorom účelová funkcia E reprezentujúca štvorec veľkosti rozdielového vektora $\|\mathbf{J}[x,y] - \mathbf{H}[x,y]\|^2$ dosiahla svoje minimum. V stabilnom stave prostredná vrstva siete obsahuje váhy, ktoré reprezentujú optimálne koeficienty $\{a_i\}$ pre reprezentáciu signálu $\mathbf{J}[x,y]$ prostredníctvom množiny elementárnych funkcií $\{\mathbf{G}_i[x,y]\}$, ktoré, ako už bolo spomenuté, nemusia byť ani ortogonálne, ani kompletné.

Konkrétny výber neortogonálnych elementárnych funkcií použitých v [155] v neurónovej sieti pre vrstvy s pevnými váhami bol urobený na základe skutočných neurofyziologických meraní (neurophysiological measurements of the two-dimensional anisotropic receptive field profiles) opisujúcich jednotlivé neuróny vizuálnej mozgovej kôry cicavcov.

Dvojrozmerné Gaborove elementárne funkcie sú parametrizované pre invariantné Gaussovské okno ktoré je umiestnené na (úplne sa prekrývajúcej) kartézskej mriežke

$$\{x_m, y_n\} = \{m \cdot M, n \cdot N\} \tag{13.18}$$

pre celé čísla (m, n) a zodpovedajúce rozmery prvkov mriežky M, N. Komplexné exponenciály, ktoré modulujú tieto prekrývajúce sa Gaussiany sú zodpovedajúco parametrizované pre kartézsku mriežku dvojrozmerných priestorových frekvencií $\{u_r, v_s\}$ vhodných k M, N priestorovej mriežky

$$\{u_r, v_s\} = \{r/M, s/N\} \tag{13.19}$$

pre celé čísla (r, s) v rozpätí $\{-(M-1/2), (M-1/2)\}$ a $\{-(N-1/2), (N-1/2)\}$.

Pre neurónovú sieť uvedenú na obr. 13.12 sú teda pre funkcie pevných váh prvej a tretej vrstvy použité dvojrozmerné Gaborove elementárne funkcie

$$\mathbf{G}_{mnrs}[x,y] = \exp\left(-\pi\,\alpha^2\left[(x - m\,M)^2 + (y - n\,N)^2\right]\right) \cdot \exp\left(-2\pi i\,[r\,\tfrac{x}{m} + s\,\tfrac{y}{n}]\right), \tag{13.20}$$

ktoré sieti umožnia konvergovať do stabilného stavu, kedy z nastaviteľných váh prostrednej vrstvy môžeme prečítať žiadané koeficienty a_{mnrs}.

Tieto získané koeficienty tvoria kompletnú dvojrozmernú Gaborovu transformáciu vstupného obrazu. Každý koeficent je komplexný, avšak pretože vstupný obraz je reálny, pre koeficienty existuje konjugovaná symetria: Cez oba parametre r a s má reálna časť a_{mnrs} párnu symetriu a imaginárna časť má nepárnu symtriu. Súčin rozsahov štyroch indexov m, n, r, s je konštantný a v kompletnom prípade sa rovná počtu pixelov obrazu. Hodnota Gaussovskej konštanty α v (13.20) určuje požadovanú oblasť podpory (počet pixelov) každej elementárnej funkcie tak, že odrezanie Gaussovského laloku je zanedbateľné.

Rekonštruovaný obraz je z kvantovaných koeficientov utvorený jednoducho sumou všetkých dvojrozmerných Gaborových funkcií váhovaných im prislúchajúcimi koeficientami:

$$\mathbf{H}[x,y] = \sum_{m,n,r,s} a_{mnrs}\ \mathbf{G}_{mnrs}[x,y]\ . \tag{13.21}$$

Systém pre kompresiu obrazu založený na Daugmanovej sieti vyjadrí obraz, ktorý sa má preniesť, ako sumu dvojrozmerných Gaborovych elementárnych funkcií. Váhový vektor **a** sa potom kóduje, prenesie a na prijímacej strane je použitý pre vytvorenie aproximácie $\mathbf{H}[x,y]$ pôvodného obrazu $\mathbf{J}[x,y]$.

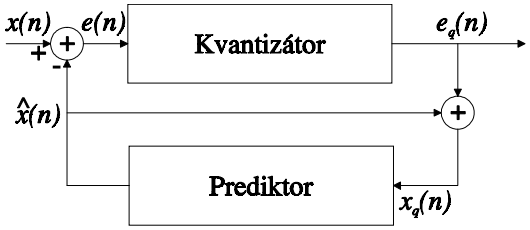
Elimináciou stupňov voľnosti pre dvojrozmerné Gaborove elementárne funkcie tak, že všetky sú si navzájom diláciami, rotáciami a transláciami so spektrálnymi parametrami množiny rozloženej v dvojrozmernej logaritmicko-polárnej mriežke, získame možnosť vyjadriť obrazy na samopodobnej množine primitívnych funkcií s výhodnou redukciou zložitosti. Takáto množina elementárnych funkcií sa podobá waveletovým expanziám. Vytvorenie dvojrozmernej Gaborovej množiny waveletov je uvedené v [155].

Iná možnosť výpočtu koeficientov dvojrozmernej Gaborovej transformácie je uvedená v [156], kde je problém riešený technikou SOR (succesive over-relaxation) a je zavedená aj kombinácia Gaborovej a diskrétnej kosínusovej transformácie (DCT) nazvaná Gaborova-DCT transformácia (Gabor-DCT transform).

13.6 PREDIKTÍVNE KÓDOVANIE VYUŽÍVAJÚCE NEURÓNOVÉ SIETE

Na obr.13.13 je bloková schéma diferenčnej pulzne-kódovej modulácie (DPCM) [138].

Prediktor využíva predošlé vzorky $x(n-1)$, $x(n-2)$, ..., $x(n-p)$, alebo v prípade obrazu susedné obrazové prvky pre výpočet odhadu $\hat{x}(n)$ aktuálnej vzorky. Pre prenos alebo uchovanie je použitá diferencia medzi aktuálnou hodnotou a odhadom $e(n) = x(n) - \hat{x}(n)$. S rastom presnosti prediktora variancia diferencií klesá, výsledkom čoho je vyšší predikčný zisk, a teda aj vyšší kompresný pomer.

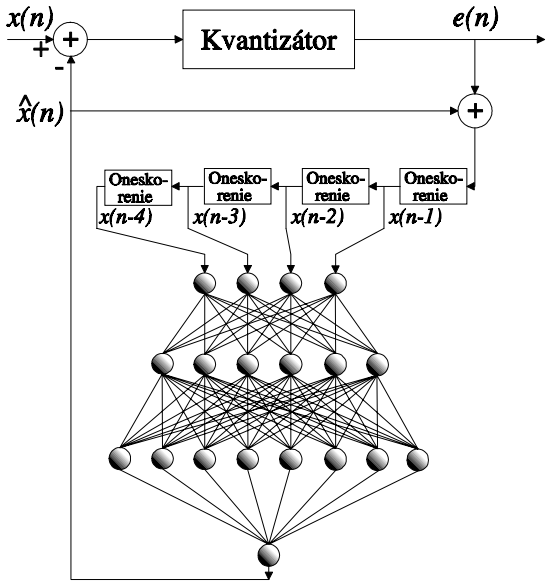


Obr. 13.13 Bloková schéma DPCM

Problémom je návrh prediktora. Možným prístupom je využitie štatistického modelu údajov pre odvodenie funkcie optimálne vzťahujúcej susediace obrazové prvky s aktuálnym prvkom. Jeden z takýchto modelov, úspešne použitý pre obrazy, je autoregresný model, zahŕňajúci lineárne predikčné kódovanie LPC (linear predictive coding) . [138]

Návrh optimálneho prediktora založený na lineárnej váhovanej sume susediacich obrazových prvkov používajúci štatistiku obrazu je teda relatívne jednoduchý. Ak je však vhodnejší nelineárny model, použitie lineárneho prediktora vedie iba k suboptimálnemu riešeniu. Návrh nelineárneho prediktora je ale omnoho zložitejší v porovnaní s lineárnym prípadom.

Neurónové siete môžu poskytnúť niektoré užitočné prístupy k optimálnemu návrhu nelineárnych prediktorov.



Obr. 13.14 DPCM využívajúca sieť so spätným šírením

Ako nelineárny prediktor môže byť použitá sieť so spätným šírením. Vstupom je predchádzajúcich p údajov a výstupom je predikovaná hodnota. Sieť môže obsahovať zvolený počet skrytých vrstiev s rozličným počtom neurónov. Na obr. 13.14 je príklad takejto konfigurácie.

Pretože sieť so spätným šírením je nelineárny systém, variancia predikčných chýb neurónovej siete môže byť nižšia ako pre lineárny prediktor - výsledkom je vyšší predikčný zisk pre DPCM systém. Konkrétny príklad využitia uvedenej schémy je uvedený v [159]. Prediktory vyšších rádov sú uvedené v [160], [161].

13.7 NEUROSIEŤOVÝ PRÍSTUP KU KLT (PCA)

Pre úplnosť veľmi stručne spomenieme samoorganizujúce sa systémy založené na Hebbovskom učení [127], [138] (stručnosťou však v žiadnom prípade nie je myslená nedôležitosť týchto systémov).

Účelom takéhoto algoritmu samoorganizácie je zistiť významné vzory (patterns) alebo príznaky - charakteristické vlastnosti (features) vo vstupných údajoch. Tento problém je označovaný ako výber alebo extrakcia príznakov (feature selection, feature extraction) [138]. Extrakcia príznakov sa vzťahuje na proces, v ktorom je priestor dát transformovaný do priestoru príznakov (feature space), ktorý má teoreticky rovnakú dimenziu ako pôvodný priestor dát. Avšak transformácia je zvolená tak, aby množina dát mohla byť reprezentovaná redukovaným počtom "účinných príznakov a stále obsahovala čo najviac vnútorného podstatného informačného obsahu údajov; inými slovami - množina dát je podrobená redukcii dimenzie.

Spomenutá problematika je vlastne "neurosieťovým jazykom vyjadrený problém, ktorého riešenie čitateľ tejto práce už z predchádzajúcich kapitol pozná ako Karhunenova-Loevova transformácia. V teórii neurónových sietí je možné túto problematiku takmer výlučne nájsť pod označením PCA (principal components analysis).

Pri implementácii KLT s využitím klasického štatistického prístupu však vzniká mnoho ťažkostí (odhad kovariancie obrazu vyžaduje veľa pamäte, riešenie pre vlastné vektory a vlastné čísla a výpočet doprednej a spätnej transformácie sú výpočtovo náročné). To je dôvodom použitia transformácií s pevne danou bázou, napr. DCT, ktorá je použitá aj v norme JPEG.

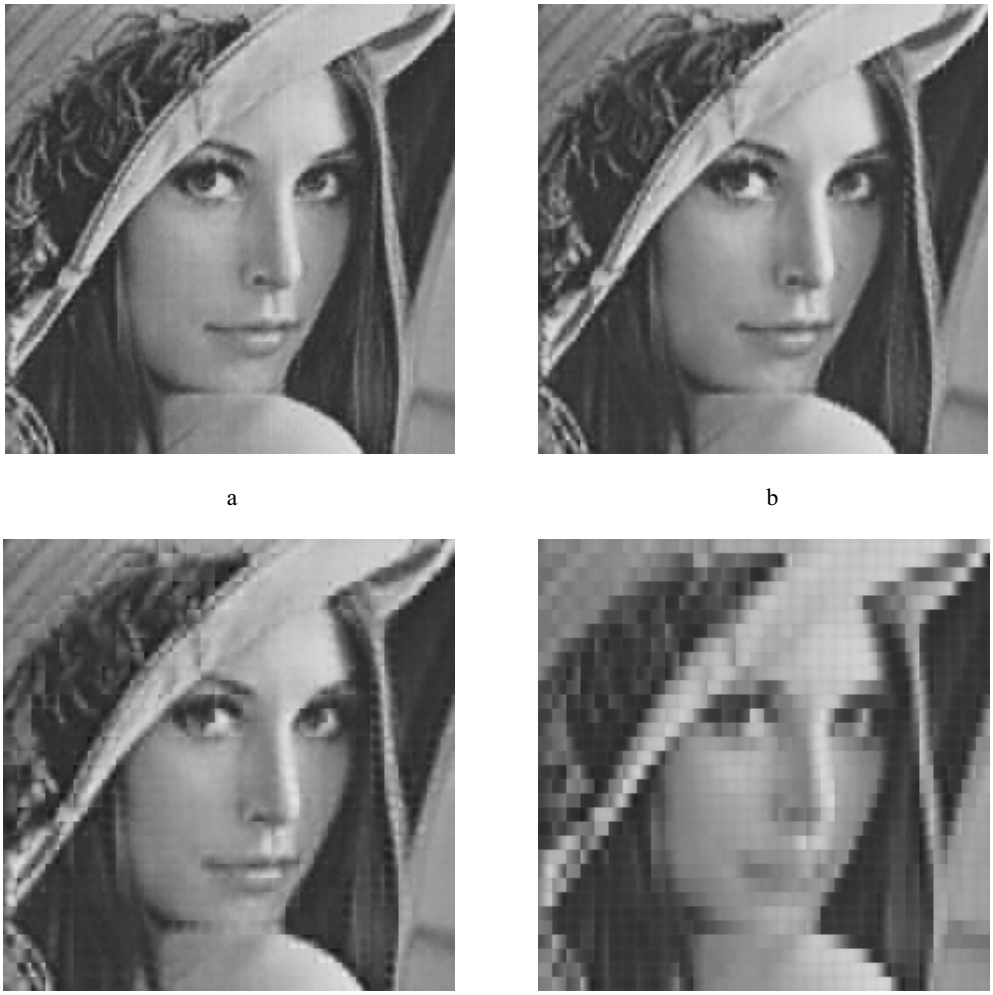
Jedným z riešení problémov súvisiacich s výpočtom bázových vektorov pomocou dekompozície odhadu kovariancie je použitie iteratívnych techník založených na modeloch neurónových sietí. Takýto prístup môže byť výpočtovo účinnejší.

Práce [127] a [138] sa zaoberajú algoritmami využívajúcimi neurónové siete, ktoré môžu vykonať KLT (PCA) pre daný vektor aj s aplikáciami pre kompresiu obrazu:

- lineárny neurón s adaptačným pravidlom Hebbovského typu
- zovšeobecnený Hebbov algoritmus - GHA (generalized Hebbian algorithm)
- APEX (adaptive principal component extraction)

V závere tejto kapitoly pripomíname, že samozrejme sme nemohli obsiahnuť úplne všetky postupy používané v kompresii údajov pomocou neurónových sietí.

Snažili sme sa prezentovať rôzne prístupy ku kompresii obrazu neurónovými sieťami. Tieto zahŕňajú sieť so spätným šírením, pre ktorú by sme mohli prístup ku kompresii obrazu nazvať čisto neurónovým, Kohonenovu samoorganizujúcu sa mapu a CPN sieť pre vektorovú kvantizáciu obrazu neurónovú sieť pre výpočet koeficientov dvojrozmernej diskrétnej Gaborovej transformácie pre transformačné kódovanie obrazových údajov, neurónovú sieť ako prediktor pri DPCM a neurosieťový prístup ku KLT (PCA).



a b

c d

Obr. 13.15 Rekonštruované obrazy, sieť so spätným šírením, bloky 8 x 8: výrez obrazu Lena pre sieť: 64-12-64 (a), 64-8-64 (b), 64-6-64 (c), 64-2-64 (d)

a b

Obr. 13.16 a, b

c

d

Obr. 13.16 Rekonštruované obrazy, samoorganizujúca sa mapa, bloky 4 x4: výrez obrazu Lena pre mapu: so 4096 prvkami (a), s 1024 prvkami (b), s 256 prvkami (c), so 64 prvkami (d)