

**CORINA BOTOCA**

**DEZVOLTAREA APLICATIILOR  
RETELELOR NEURONALE CELULARE ÎN  
TEHNICA OPERATORILOR ȘI A  
MEMORIILOR ASOCIATIVE**

21.7.88  
[ ]

BIBLIOTECA CENTRALĂ  
UNIVERSITATEA "POLITEHNICA"  
TIMIȘOARA

# Cuprins

Cuvânt înainte . . . . .	1
Lista de simboluri . . . . .	5

## **PARTEA I NOTIUNI GENERALE. ASPECTE ALE PROIECTĂRII SI IMPLEMENTĂRII REțelelor NEURONALE CELULARE . . . 7**

<b>1. Introducere . . . . .</b>	<b>7</b>
1.1. Stadiul actual al cercetării în domeniul rețelelor neuronale celulare . . . . .	7
1.2. O analiză a dezavantajelor și avantajelor rețelelor neuronale celulare . . . . .	10
1.2.1 Dezavantajele . . . . .	10
1.2.2 Avantajele rețelelor neuronale celulare . . . . .	10
<b>2. Rețele neuronale celulare. Generalități . . . . .</b>	<b>13</b>
2.1. Rețele neuronale celulare generalizate . . . . .	13
2.1.1 Arhitectura rețelelor neuronale celulare generalizate . . . . .	13
2.1.2. Structura unei celule . . . . .	17
2.2. Rețele neuronale celulare analogice . . . . .	18
2.2.1. Dinamica rețelei . . . . .	18
2.2.2. Structura unei celule . . . . .	20
2.3. Rețele neuronale celulare discrete . . . . .	21
2.3.1. Dinamica rețelei . . . . .	21
2.3.2. Structura unei celule . . . . .	22
<b>3. Aspecte ale proiectării rețelelor neuronale celulare și modalitățile de implementare . . . . .</b>	<b>24</b>
3.1 Metode de determinare a matricilor de conexiune . . . . .	24
3.1.1. Algoritmi pentru convergența la puncte fixe. . . . .	26

3.1.2. Metode de tip gradient pentru învățare de traiectorii prescrise pentru rețelele neuronale celulare analogice . . . . .	27
3.1.3 Metode de învățare de traiectorii prescrise în rețelele neuronale celulare discrete . . . . .	29
3.1.4 Algoritmi globali pentru rețelele neuronale celulare discrete . .	30
3.1.5 Algoritmi orientați spre implementarea hard . . . . .	32
3.1.6 Algoritmi genetici . . . . .	32
3.2 Robustetea . . . . .	33
3.3 Domeniul dinamic . . . . .	34
3.4 Stabilitatea . . . . .	35
3.5 Generalizarea . . . . .	36
3.6 Implementarea hard și soft . . . . .	37
3.6.1 Implementarea soft . . . . .	37
3.6.2 Acceleratorul hard . . . . .	38
3.6.3. Workstation pentru rețele neuronale celulare . . . . .	39
3.6.4. RNC cu ponderi fixe . . . . .	39
3.6.5 RNC dual programabile . . . . .	39
3.6.6 Mașina universală RNC . . . . .	40
3.7 Concluzii . . . . .	41

## **PARTEA II REȚELE NEURONALE CELULARE**

### **CA OPERATORI . . . . . 42**

<b>4 Rețelele neuronale celulare ca operatori spațiali . . . . .</b>	<b>42</b>
4.1 Aplicații posibile . . . . .	42
4.2 Analiza și detecția mișcării . . . . .	43
4.2.1 Cadrul general al rețelelor neuronale celulare pentru mișcări continue. . . . .	44
4.2.2 Metoda de determinare a matricilor de conexiune pentru separarea părții statice a imaginilor ce conțin obiecte în mișcare . . . . .	46
4.2.3 Rezultatele simulării. . . . .	48
4.3 Reconstrucția obiectelor tridimensionale prin interpolare și aproximare . . . . .	50
4.3.1. Modelul pentru descrierea unei suprafețe . . . . .	50
4.3.2 Discretizarea modelului . . . . .	51

4.3.3 Găsirea minimului prin utilizarea unei metode de tip gradient. Interpolarea . . . . .	52
4.3.4 Aproximarea . . . . .	53
4.3.5 Rezultatele simulării. . . . .	54
4.4 rotația obiectelor tridimensionale . . . . .	62
4.4.1 Deplasarea unui singur pixel negru pe o cale circulară. . . . .	62
4.4.2 Rotirea a doi pixeli . . . . .	65
4.4.3 Rezultatele simulării. . . . .	67
4.5 Concluzii . . . . .	68

## **PARTEA III MEMORII ASOCIATIVE . . . . . 69**

### **5. Memorii asociative. Noțiuni fundamentale . . . . . 69**

5.1 Generalități . . . . .	69
5.2 Capacitatea de memorare . . . . .	72
5.3 Determinarea bazinului de atracție . . . . .	73
5.4 Metode de proiectare . . . . .	76
5.4.1 Pentru memorarea informației în punctele de echilibru stabil . . . . .	76
5.4.2 Pentru memorarea informației ca oscilații periodice . . . . .	77
5.5 Concluzii . . . . .	78

### **6. Memorii asociative ce utilizează regula lui Hebb . . . . . 79**

6.1 Modelul Vandewaale . . . . .	79
6.1.1 Caracteristicile rețelei și condițiile impuse asupra modelelor de memorat . . . . .	79
6.1.2 Rezultatele simulării. . . . .	81
6.2 Memorie asociativă cu reacție inversă . . . . .	82
6.2.1 Structura rețelei și dinamica . . . . .	82
6.2.2 Caracteristicile rețelei și condițiile impuse asupra modelelor de memorat . . . . .	84
6.2.3 Capacitatea de memorare . . . . .	85
6.2.4 Rezultatele simulării. . . . .	85
6.2.5 Observații . . . . .	95
6.3 Algoritm rapid de învățare a ponderilor . . . . .	96
6.3.1 Prezentarea algoritmului și a bazelor sale teoretice . . . . .	96
6.3.2 Rezultatele simulării. . . . .	99
6.4 Algoritm rapid cu un număr redus de conexiuni . . . . .	100
6.4.1 Considerații teoretice . . . . .	101

6.4.2	Rezultatele simulării . . . . .	102
6.4.3	Aplicație. Ghidarea automată a unui vehicul . . . . .	106
6.4.4	Observații . . . . .	109
6.5	Concluzii . . . . .	110
<b>7.</b>	<b>Memorii asociative obținute prin metoda relaxării . . . . .</b>	<b>113</b>
7.1	Considerații teoretice . . . . .	113
7.2	Rezultatele simulării . . . . .	114
<b>8.</b>	<b>Memorii asociative obținute prin descompunerea</b>	
	<b>în valori singulare . . . . .</b>	<b>116</b>
8.1	Considerații teoretice . . . . .	116
8.2	Rezultatele simulării . . . . .	121
8.3	Concluzii . . . . .	129
<b>9.</b>	<b>Memorii asociative obținute prin metoda pseudoinversei .</b>	<b>131</b>
9.1	Memorie asociativă discretă . . . . .	132
9.2	Memorie asociativă discretă având capacitatea de a învăța și uita modele . . . . .	133
9.2.1	Memorie asociativă discretă cu capacitatea de învățare .	133
9.2.2	Memorie asociativă discretă cu capacitatea de uitare . .	135
9.3.	Memorie asociativă complet stabilă . . . . .	137
9.4	Memorii asociative de ordin superior ce utilizează metoda pseudoinversei. . . . .	138
9.5	Rezultatele simulării. . . . .	141
9.6	Studiu comparativ al performanțelor memoriilor asociative. . . .	150
<b>10.</b>	<b>Considerații finale . . . . .</b>	<b>153</b>
10.1	Perspective ale evoluției rețelelor neuronale celulare . . . . .	153
10.2	Perspective ale evoluției rețelelor neuronale în contextul inteligenței artificiale . . . . .	154
10.3	Contribuții personale . . . . .	155
<b>Anexa 1</b>	. . . . .	<b>157</b>
<b>Anexa 2</b>	. . . . .	<b>158</b>
<b>Anexa 3</b>	. . . . .	<b>161</b>
<b>Anexa 4</b>	. . . . .	<b>168</b>
<b>Bibliografie</b>	. . . . .	<b>170</b>

# Cuvânt înainte

Sinergia a doi factori : nevoia și imaginația au marcat, în esență, evoluția civilizației umane. De la ideea realizării inteligenței artificiale, care a apărut pe la începutul acestui secol, ca pură ficțiune, până la roboții “inteligenți” de astăzi, creați din nevoia de a ne ușura viața, drumul a fost extrem de anevoios, dar raportat la scara evoluției umane, extrem de rapid.

Semnificația termenului de “inteligentă” este încă o problemă de dezbatere. Inteligența implică abilitatea de a deduce date utile din informația memorată, de a face raționamente, de a construi arhitecturi ale gândirii.

Încă de la începutul cercetărilor s-au conturat două direcții diferite de dezvoltare a inteligenței artificiale:

- paradigma logico-simbolică - dezvoltarea unui soft adecvat, utilizând calculatoarele clasice existente (inteligenta artificială);
- paradigma conexionistă - realizarea unui hard de concepție cu totul nouă, care să simuleze mecanismele gândirii umane (rețelele neuronale);

Cele două direcții de cercetare au evoluat în paralel uneori, alteori una a focalizat interesul în detrimentul celeilalte. S-au conturat ca un rezultat al cercetărilor din domeniul fiziologiei normale și patologice a sistemului nervos uman, a fizicii, matematicii, calculatoarelor, și le-au impulsionat la rândul lor evoluția. Dezvoltarea rețelilor neuronale a fost stimulată în special de rezultate slabe ale programelor de inteligență artificială în procesul de învățare. Învățarea este o caracteristică esențială a inteligenței umane, iar pentru rețelele neuronale învățarea este o caracteristică intrinsecă.

Programele realizate pentru calculatoarele convenționale au eșuat în a furniza cadrul pentru reprezentarea cunoașterii într-o manieră care să permită sinteze automate, demonstrând creativitatea inteligenței umane. Ele nu au realizat nici măcar un contact real cu posibilitățile și slăbiciunile creierului. Se pare că materia cenușie, deși mult mai înceată, are o arhitectură mult mai potrivită pentru procesarea informațiilor naturale. Oamenii sunt mai preciși, mai fluenți, mai rapizi, în special în problemele complexe, îmbunătățindu-și performanța prin experiență.

Acolo unde formalismul convențional a eșuat ca urmare a exploziei combinatoriale și a timpului mare de procesare datorat secvențialității, rețelele neuronale vin să ofere un

cadru nou de procesare a cunoașterii. În mod similar cu organizarea creierului, rețelele neuronale utilizează o procesare masiv paralelă și distribuită.

În ultimii ani, odată cu revigorarea cercetării în domeniul teoriei mulțimilor fuzzy, s-a afirmat tot mai mult tendința fuziunii dintre rețelele neuronale și teoria fuzzy, calculul neuro-fuzzy, care oferă un grad mai sporit de robustețe și un coeficient de inteligență mai ridicat.

Integrarea tehnicilor inteligente poate avea consecințe spectaculoase, sinergia lor evoluând spre o formă de inteligență artificială netradițională.

În acest context rețelele neuronale celulare s-au dezvoltat și au evoluat rapid ca o specializare aparte, permițând abordarea unor probleme complexe spațio-temporale, oferind soluții originale și performante. Posibilitatea implementării integrate pe scară foarte largă (VLSI) și viteza de procesare fără precedent sunt argumente în plus ce le impulsionează afirmarea.

Lucrarea de față urmărește introducerea noțiunilor fundamentale, a terminologiei specifice, prezentarea într-o manieră unitară și sintetică a bazelor teoretice ale proiectării rețelelor neuronale celulare, cu avantajele și dezavantajele fiecărei metode și pune în lumină o paletă largă de posibile aplicații practice.

Se prezintă într-o manieră comparativă memoriile asociative implementate cu rețele neuronale celulare. Rodul cercetării îl reprezintă o serie de soluții originale și performante în câteva probleme de interes practic: recunoașterea formelor, memoriile asociative.

Concluziile lucrării prefigurează perspectivele evoluției rețelelor neuronale celulare.

Lucrarea este structurată în trei părți: "Noțiuni teoretice generale. Aspecte ale proiectării și implementării rețelelor neuronale celulare", "Aplicații ale rețelelor neuronale celulare ca operatori" și "Memorii asociative".

Capitolul 1 "Introducere" amplacează lucrarea în contextul principalelor direcții de cercetare teoretică și aplicativă realizate până acum în domeniu. Se prezintă dezavantajele rețelelor neuronale celulare, dezavantaje moștenite de la marea categorie a rețelelor neuronale (ale căror descendenți sunt) și avantajele rezultate din caracteristicile lor particulare.

Capitolul 2 "Rețele neuronale celulare. Generalități" introduce tipurile de rețele neuronale celulare, conceptele fundamentale, care vor fi utilizate în capitolele următoare.

Capitolul 3 "Aspecte ale proiectării rețelelor neuronale celulare și modalitățile de implementare" abordează câteva aspecte fundamentale legate de proiectarea RNC și trece în revistă metodele de determinare a interconexiunilor RNC.

Capitolul 4 "Rețelele neuronale celulare ca operatori spațiali" ilustrează prin câteva exemple utilizarea RNC ca operatori de imagini. Se prezintă o metodă de determinare a matricilor de interconexiune în cazul funcționării pe baza unui tabel de adevăr, experimentată prin detecția părților statice ale obiectelor în mișcare. Se abordează

reconstrucția tridimensională prin aproximare interpolare și rotația obiectelor. Sunt prezentate rezultatele experimentelor.

Capitolul 5 “Memorii asociative. Noțiuni fundamentale” expune noțiunile teoretice fundamentale, aspectele legate de proiectarea și evaluarea performanțelor memoriilor asociative implementate prin RNC.

Capitolul 6 “Memorii asociative ce utilizează regula lui Hebb” expune câteva modele publicate în literatura de specialitate și introduce unele noi, originale. Se analizează teoretic arhitecturile propuse, se prezintă experimentele realizate și se comparară performanțele memoriilor abordate.

Capitolul 7 “Memorii asociative obținute prin metoda relaxării” tratează implementarea metodei relaxării în determinarea interconexiunilor unei memorii asociative.

Capitolul 8 “Memorii asociative obținute prin descompunerea în valori singulare” descrie utilizarea algoritmului descompunerii în valori singulare în implementarea unei memorii asociative. Se prezintă rezultatele experimentelor cu o astfel de memorie.

Capitolul 9 “Memorii asociative obținute prin metoda pseudoinversei” tratează memoriile asociative pseudoinverse. Sunt dezvoltate mai multe variante și se prezintă rezultatele experimentelor.

Se prezintă un studiu comparativ al performanțelor memoriilor asociative implementate prin RNC.

Capitolul 10 “Considerații finale” este dedicat concluziilor, perspectivelor RNC și sintetizării contribuțiilor originale.

## **Mulțumiri**

Lucrarea este rezultatul activității de cercetare sub îndrumarea competentă, atentă și permanentă a domnului Prof. Ioan Naforniță. Îi sunt profund recunoscătoare.

Mulțumesc doamnei Prof. Adelaida Mateescu pentru încrederea și suportul moral acordate .

Mulțumesc domnului Prof. L.Goraș pentru solitudinea cu care răspuns ori de câte ori am fost în impas.

Mulțumesc conducerii Facultății de Electronică și Telecomunicații pentru baza materială asigurată, fără de care cercetarea nu ar fi fost posibilă.

În elaborarea tezei un rol major le-au avut stagiile de perfecționare efectuate în cadrul "Analogic and Neural Computing Laboratory, Computer and Automation Institute, Hungarian Academy of Sciences", din Budapesta, sub conducerea Prof. Tamas Roska, o personalitate de talie mondială în domeniul RNC, unde am avut privilegiul de a lucra sub coordonarea directă a Domniei sale, în cadrul unei echipe de cercetare de excepție. Mulțumesc domnului Prof. Tamas Roska pentru amabilitatea de a mă primi în cadrul prestigioasei sale echipe de cercetare, pentru sprijinul acordat, pentru sfaturile competente și nu în ultimul rând pentru încurajarea permanentă.



O mențiune aparte o merită Domnul Dr. Peter Szolgay unul din membrii de bază al colectivului de cercetători ai laboratorului din Budapesta.

Mulțumesc colegilor pentru climatul de colaborare și pentru sprijinul acordat care a făcut posibilă finalizarea lucrării.

Mulțumesc doamnei ing. Carmen De Sabata pentru buna colaborare avută în ultimul an.

Mulțumesc lui Horia Urzică pentru ajutorul acordat la tehnoredactarea tezei.

Mulțumesc soțului meu Mircea și familiei mele pentru înțelegere și sprijin.

# Lista de simboluri

## Abrevieri

- DCC - detector de componente conectate
- 3D - tridimensional
- MA - memorie asociativă
- MAB - memorie asociativă bidirecțională
- MU-RNC - mașina universală cu rețele neuronale celulare
- RL - regiune liniară
- RN - rețea neuronală
- RNC - rețea neuronală celulară
- RNCA - rețea neuronală celulară analogică
- RNCD - rețea neuronală celulară discrete
- RNCG - rețea neuronală celulară generalizată
- RRE - retropropagarea recurentă a erorii
- RRE - retropropagarea erorii în timp
- RS - regiune de saturație
- RSP - regiune de saturație parțială
- VC-dim - dimensiunea Vapnik-Chervonenkis
- VLSI - integrare pe scară largă (very large scale integration)

## Lista notațiilor uzuale<sup>1</sup>

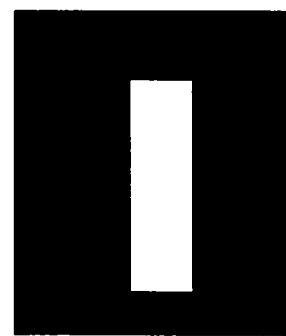
- $A$  - este matricea de conexiune de reacție;
- $A^r$  - este matricea de conexiune de reacție cu întârziere în timp;
- $b$  - gradul de regăsire în sensul distanței Hamming
- $B$  - este matricea de conexiune de control;
- $B^r$  - este matricea de conexiune de control cu întârziere în timp;

<sup>1</sup> Notațiile particulare utilizate în anumite capitole au fost explicate în momentul folosirii lor

- $C$  - capacitatea echivalentă a unei celule
- $C_{ij}$  - celula de coordonate  $i$  și  $j$  într-o rețea neuronală celulară bidimensională
- $E$  - funcția de eroare
- $E_{ij}^q$  - funcția de energie pentru modelul  $q$  asociată celulei  $C_{ij}$
- $h$  - distanța Hamming dintre două modele
- $k, l$  - coordonatele celulelor din vecinătate
- $M$  - dimensiunea pe axa  $x$  a unei rețele neuronale celulare bidimensionale
- $m_0$  - complexitatea eșantioanelor
- $N$  - dimensiunea pe axa  $y$  a unei rețele neuronale celulare bidimensionale
- $N_r$  - vecinătatea de rază  $r$
- $I$  - matricea de polarizare;
- $P$  - numărul modelelor de memorat
- $q$  - un model oarecare dintre cele  $P$  de memorat  $q=1, 2, \dots, P$
- $r$  - raza vecinătății
- $R$  - robustețea
- $R_x$  - rezistența unei celule;
- $t$  - timpul continuu
- $u$  - este intrarea independentă
- $x$  - este starea celulei
- $X_{ij}^q$  - submodelul de stare corespunzător modelului  $q$ , din vecinătatea celulei  $C_{ij}$
- $\dot{x}$  - este derivata stării unei celule
- $y$  - este ieșirea unei celule
- $\bar{y}$  - valoarea medie a ieșirii
- $Y_{ij}^q$  - submodelul ieșirilor corespunzătoare modelului  $q$ , din vecinătatea celulei  $C_{ij}$

.....

# Partea



# Notiuni generale

# Capitolul 1

## INTRODUCERE

### 1.1 Stadiul actual al cercetării în domeniul rețelelor neuronale celulare

Leon Chua și Lin Yang au fundamentat în 1988, la Berkeley, rețelele neuronale celulare (RNC), o nouă categorie de rețele neuronale, cu o evoluție spectaculoasă în ultimul deceniu, ce au generat o direcție de cercetare distinctă .

În articolul fundamental, “Cellular Neural Networks: Theory” [27], ei introduc următoarea definiție:

*“Like a neural network, a cellular neural network is a large scale nonlinear analog circuit which processes signals in real time. Like cellular automata, it is made of massive aggregate of regularly spaced circuit clones, called cells, which communicate with each other directly only through its nearest neighbors. Each cell is made of a linear capacitor, a nonlinear voltage controlled current source and a few resistive linear circuit elements.*

*Cellular neural networks share the best features of both worlds; its continuous time feature allows real time signal processing found wanting in the digital domain and its local interconnection feature makes it tailor made for VLSI implementation”.*

Tamás Roska dă ulterior o descriere mai sintetică [82]:

*“Analog processor arrays placed on a 3D regular grid interacting within a finite neighbourhood: this is the CNN (cellular neural network) paradigm. Using other words: it is a programmable prototype machine performing nonlinear dynamic spatial convolutions in real time”*

Din definițiile prezentate se conturează câteva caracteristici esențiale ale RNC: structura geometrică regulată, localitatea conexiunilor și programabilitatea, ceea ce le conferă un potențial de procesare deosebit, în timp real (vezi §1.2.2).

Datorită conexiunilor locale, RNC oferă un model relativ simplu și în același timp puternic pentru o reprezentare deterministă a fenomenelor spațio-temporale (descrise prin ecuații diferențiale neliniare și cu întârziere în timp). RNC reprezintă cadrul adecvat pentru modelarea fenomenelor din diverse domenii : mecanică (analiza rezistenței materialelor), chimie (modele de coroziune), electromagnetism, geofizică (dinamica poluării), biologie (modelul retinei, modele de morfogeneză și de difuzie).

De la arhitectura de bază până la inventarea mașinii universale RNC, și a supercomputerului, în 1992, rețelele neuronale celulare au cunoscut o dezvoltare fără precedent în știință, concentrând interesul cercetătorilor din lumea întreagă, conducând la formarea unor colective puternice, care colaborează între ele și a căror activitate le este dedicată, în mod special. De remarcat sunt :

- “The Nonlinear Electronics Laboratory” condus de profesorul L. O. Chua la “University of California at Berkeley”;
- “The Analogic and Neural Computing Laboratory” condus de profesorul T. Roska la “Computer and Automation Institute, Hungarian Academy of Sciences” din Budapesta;
- “The Vision Research Laboratory” condus de profesorul F. Werblin la “University of California at Berkeley”;
- “The Neurobiology Laboratory” condus de profesorul J. H. Mori, la “University of Medicine in Budapest”;
- “The Analog Integrated Circuit Design Laboratory “ condus de profesorul A. Rodriguez-Vasquez la “University of Seville”;
- Laboratorul condus de profesorul A. Ushida la “University of Tokushima”;
- Laboratorul condus de profesorul P. P. Civalleri la “The Technical University of Torino”;
- “The Laboratory PASTIC” condus de profesorul J. Zerubia la INRIA, Sophia - Antipolis, Franța;
- Laboratorul condus de profesorul J. A. Nossek, la “The Technical University of Munich”;
- “The Signal Processing Laboratory “ condus de profesorul G. Moschytz la ETH Zurich;
- Laboratorul condus de profesorul D. Wolf, la “Goethe University”, Frankfurt;
- Laboratorul condus de prof. J. Vandewalle, la “Catholic University of Leuven”, Olanda;
- Laboratorul condus de M. Tanaka la “Sophia University”, Tokyo;

Cercetarea în domeniul RNC abordează o problemă cu o paletă extrem de largă, care acoperă aproape toate domeniile activității umane:

- recunoașterea formelor:
  - prelucrări simple de imagini: extragere de contur, filtrare, skeletonizare, sortare după orientare, după dimensiune, mărire și micșorare de imagine, mărirea contrastului, aproximare și interpolare, rotire;
  - extragerea unui anumit model din imagine (detecția gurii pe imaginile faciale în mișcare, detecția mânerelor de ușă din imagini pentru protezarea orbilor);
  - analiza și detecția mișcării;
  - îmbunătățirea rezoluției (în faxuri, scannere, copiatoare);
  - recunoașterea caracterelor (scrisului de mână, cu caractere latine și japoneze);
  - îmbunătățirea imaginilor microscopice;
  - compresia imaginilor și reconstrucția lor bidimensională și tridimensională etc.;
- controlul roboților, al deplasării lor într-un mediu necunoscut;
- baze de date prin memorii asociative;
- în procesele tehnologice, în testarea defectelor (defectele din textura materialelor, din realizarea cablajelor);
- în implementarea unor modele biologice, pentru percepția vizuală (modelul retinei), pentru fenomenul de difuzie între membranele celulelor, pentru morfogeneza creierelor ;
- în medicină:
  - dezvoltarea unui sistem RNC analogic pentru analiza mamografiilor în vederea diagnosticării interactive a cancerului mamar;
  - detecția carcinoamelor bronhogenetice și alte analize tomografice computerizate;
  - îmbunătățirea imaginilor din ecocardiografia ultrasonică;
  - analiza cromozomială, toolkit pentru partiția, identificarea și analiza defectelor posibile;
- rezolvarea de ecuații parțiale diferențiale în timp real. În această direcție s-au obținut rezultate promițătoare în îmbunătățirea imaginilor amprentelor digitale;
- rezolvarea de probleme de optimizare;
- realizarea unor supercomputere dintr-o nouă generație, chipul RNC de mașină universală și platforma pentru testarea lui;

Sute de articole având ca subiect investigațiile teoretice, aplicațiile și proiectarea RNC, cât și modelarea neuromorfică prin RNC au fost elaborate, în special în ultimii

cinci ani. Majoritatea studiilor în domeniu au fost publicate în volumele celor trei workshop-uri sub egida IEEE dedicate rețelelor neuronale și aplicațiilor lor. Acestea au avut loc în 1990 la Budapesta, în 1992 la München și în 1994 la Roma.

“International Journal on Circuit Theory and Applications” și “Transactions on Circuits and Systems” au consacrat numere speciale RNC în septembrie - octombrie 1992, ianuarie - februarie 1996, mai - iunie 1996, respectiv în aprilie 1993 (seria I), martie - aprilie 1993 (seria II), 1994, octombrie 1995 (seria I).

## **1.2 O analiză a dezavantajelor și avantajelor rețelelor neuronale celulare**

### **1.2.1 Dezavantajele**

O analiză critică a rețelelor neuronale celulare dezvoltate până în prezent relevă câteva dintre dezavantajele lor, moștenite de la categoria de RN, din care s-au desprins:

- lipsa unei abordări teoretice unitare;
- generarea unei soluții particulare, adecvate numai în cazul unei aplicații concrete, fiind dificilă adaptarea ei pentru o altă aplicație ;
- lipsa unor informații referitoare la arhitectura necesară într-o aplicație dată, date sub forma unor reguli de construcție clară (de exemplu numărul de straturi, numărul de neuronistrat, rețea cu sau fără reacție). Majoritatea implementărilor au evoluat experimental, prin metoda “trial and error”, strict orientate spre găsirea soluției într-o aplicație concretă.
- structurile cu vecinătăți de ordin superior ridică probleme, deși datorită localității interconexiunilor RNC sunt mult mai ușor implementabile;

În cazul RNC cu matrici de interconexiune variabile în spațiu calculul este distribuit în toată masa rețelei, în concluzie depanarea este dificilă, deși mult mai accesibilă decât în cazul altor RN și greu este posibilă localizarea componentei care duce la funcționare eronată la un moment dat.

### **1.2.2. Avantajele rețelelor neuronale celulare**

Avantajele rețelelor neuronale celulare sunt date de caracteristicile care le particularizează în marea clasă a rețelelor neuronale și sunt:

#### **Localitatea conexiunilor**

Localitatea conexiunilor, de inspirație biologică, reprezintă caracteristica esențială a RNC și a fost practic determinată de necesitatea implementării VLSI. Fiecare procesor



elementar (celulă) interacționează cu celelalte procesoare din rețeaua neuronală, într-o vecinătate finită. Raza vecinătății este mult mai mică decât numărul celulelor din rețea. Matricea de interconexiuni se numește “cloning template”. În majoritatea aplicațiilor aceasta este aceeași pentru toate procesoarele, proprietate cunoscută sub numele de invarianță în spațiu.

Marele avantaj față de alte RN este că depanarea, în cazul RNC cu matrici de interconexiune invariante în spațiu, este ușor de realizat, datorită acțiunii locale a operatorilor.

### **Structura geometrică regulată**

Toate procesoarele sunt identice ca structură electrică și pot fi specificate în mod unic prin câțiva parametrii, cel mult 19 numere reale [86]. Funcționarea unei RNC de orice dimensiune poate fi controlată prin modificarea a 19 parametrii a unui singur tip de circuit. **Independența de dimensiune** este cheia programabilității atât din punct de vedere soft, cât și hard.

### **Programabilitatea**

Din punct de vedere soft, cele 19 numere pot fi interpretate ca un program ce implementează un anumit tip de funcționare a neuronului, un anumit algoritm. Implementarea aceluiași algoritm pe un computer digital convențional ar necesita scrierea unui program cu mii de instrucțiuni. Dacă luăm în considerare faptul că acești parametrii se pot permuta între ei de 19! ori, fiecare situație corespunzând unei RNC distincte, compresia este impresionantă, față de cazul instrucțiunilor unui computer digital Von Neumann.

Din punct de vedere hard sunt necesare doar 19 conexiuni spre exteriorul chip-ului. Acestea permit programarea externă, fără restricții datorate mărimii rețelei. Față de calculatoarele clasice seriale, unde timpul de procesare depinde exponențial de mărimea numărului de procesoare, timpul de stabilizare al unei RNC depinde linear în multe dintre cazuri. De fapt, pentru multe aplicații timpul de procesare este independent de dimensiunea rețelei.

### **Posibilitatea implementării VLSI**

Prin localitatea conexiunilor și structura geometrică regulată, RNC sunt un concept natural pentru implementările VLSI. Acesta reprezintă avantajul major al RNC față de celelalte arhitecturi neuronale. Datorită proprietății de invarianță în spațiu este posibilă construirea de rețele mari prin potrivirea condițiilor la limită, dintre chip-uri. Aceasta înseamnă că mărimea unei RNC nu este restricționată de numărul neuronilor ce pot fi implementați într-un chip.

Performanța realizată, de 0.3 Terra operații analogice pe secundă, pe o suprafață de  $1 \text{ cm}^2$ , reprezintă o **viteză de procesare fără precedent**, care nu face decât să confirme supozițiile teoretice. Implementările optice sunt în fază experimentală, capabile de procesări cu viteze superioare.

## Potențialul de procesare

Programabilitatea interconexiunilor de control, reacție și de polarizare le conferă RNC un potențial de procesare deosebit, permițându-le o comportare variată și complexă, inclusiv ca oscilator sau simulator de haos, în aplicații ce necesită o deosebită capacitate de procesare (ca de exemplu probleme de optimizare globală, de sortare, de numărare, de filtrare mediană). RNC pot simula unde autogeneratoare (autowaves) pot produce “ Turing pattern-uri<sup>1</sup>” (într-un mod similar reacției de difuzie studiate de Turing), sau pot genera alte fenomene dinamice spectaculoase, cum ar fi undele concentrice, spiralele .

Evident comportarea rețelei depinde de condițiile inițiale, de condițiile la limită, de structura geometrică și de dimensiunile domeniilor spațiale [40].

În colaborare cu neurologii a fost elaborat modelul retinei umane. Comportarea deosebit de complexă a RNC a permis și generarea modelului difuziei chimice dintre două membrane celulare, a unor modele morfogenetice. Astfel prin RNC se elaborează modele cu plauzibilitate biologică, ce permit o mai bună înțelegere atât a anatomiei creierului, cât și a procesării cunoașterii umane. În același timp modelele RNC oferă un cadru experimental al degradării funcțiilor biologice la deteriorările structurale.

Cea mai recentă invenție în domeniu, mașina universală RNC [81],[83] combină abordarea analogică cu cea digitală . Ea este singura rețea neuronală cu o dinamică similară funcțiilor de “tip analogic” implementate în emisfera dreaptă a creierului și a funcțiilor de “tip discret” implementate în emisfera stângă a creierului uman. Cu alte cuvinte, mașina universală RNC este o primă încercare grosieră a implementării asimetriei a creierului uman.

S-a demonstrat că mașina universală RNC este o mașină Turing analogică [87], ea fiind capabilă să rezolve orice problemă ce poate fi soluționată cu un calculator clasic.

Dezvoltarea unor algoritmi de învățare adaptivi și utilizarea tehnologiilor optice, electromagnetice și cuantice în implementarea pe scară industrială deschid noi perspective RNC.

1 Pattern - starea staționară de curent continuu obținută prin comportarea dinamică diferențiată a unor elemente identice, cuplate într-o distribuție spațială regulată. Altfel spus: celulele unei RNC generează pattern-uri dacă au ieșiri inegale în starea de curent continuu.

Fenomene similare care implică o rupere a simetriei sunt studiate în biologie (modelele morfogenetice), biochimie (modelul de difuzie între membranele celulare), fizică, ecologie, electronică.

# Capitolul 2

## REȚELE NEURONALE CELULARE. GĂNERALITĂȚI

### 2.1 Rețele neuronale celulare generalizate

#### 2.1.1 Arhitectura rețelelor neuronale celulare generalizate

În 1993 L.Chua și Guzelis au reunit în conceptul de rețea celulară neuronală generalizată (RNCG) toate variantele cunoscute de RNC [44].

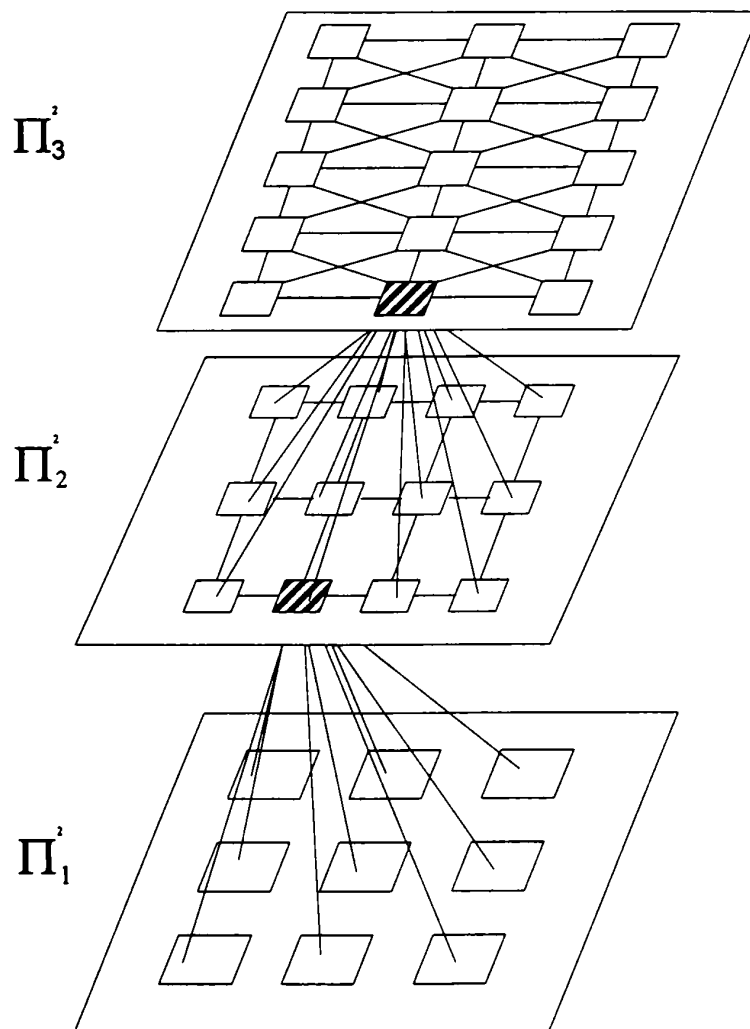
##### **Definiția 1**

*O rețea celulară neuronală generalizată este un sistem neliniar, analogic, dinamic, format dintr-un număr mare de subcircuite cu o aceeași structură, plasate pe o grilă tridimensională. Subcircuitele sunt sisteme dinamice, de ordin arbitrar, conectate local, într-o vecinătate finită.*

Altfel spus, o RNCG este un procesor paralel programabil ce efectuează convoluții dinamice spațiale în timp real. RNCG se pot considera o alternativă tridimensională analogică pentru automatul celular bidimensional inventat de von Neumann [81].

O RNCG are, în cazul cel mai complex  $m$  straturi. Fiecare strat, la rândul său poate avea o arhitectură unidimensională, cu  $N_1$  celule, notată  $\Pi^1$  [9], bidimensională, de  $N_1 \times N_2$  celule, notată  $\Pi^2$ , sau o arhitectură  $n$  dimensională, de  $N_1 \times N_2 \times \dots \times N_n$  celule, notată  $\Pi^n$  ( $N_i$  fiind un număr întreg).

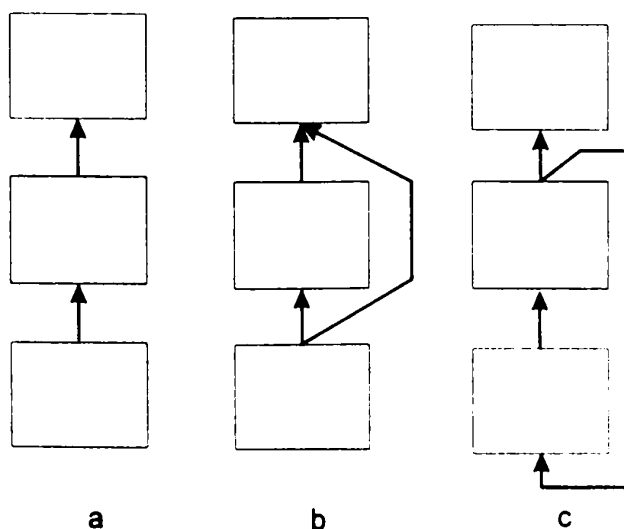
Numim rețea  $n$  dimensională cu  $m$  straturi o RNCG formată din  $m$  straturi cu  $n$  dimensiuni. Fig.2.1 reprezintă structura unei rețele bidimensionale cu trei straturi. Ca în figură, numărul celulelor și tipul conexiunilor dintre ele pot diferi de la un strat la altul. Fiecare unitate poate fi conectată atât cu vecinii (în conformitate cu definiția 2) de pe stratul ei cât și cu cei de pe alte straturi. În cazul cel mai general fiecare celulă a fiecărui strat poate fi conectată la celulele mai multor straturi.



**Fig. 2.1 Rețea neuronală celulară generalizată bidimensională cu trei straturi**

Ca de exemplu, în Fig.2.1 fiecare celulă a fiecărui strat are conexiuni într-o vecinătate de  $r=1$ . Stratul doi este complet conectat cu celelalte două straturi, adică fiecare celulă a stratului doi este conectată cu toate celulele stratului unu, respectiv trei.

Fig. 2.2 ilustrează câteva tipuri de conexiuni posibile într-o rețea cu trei straturi.



**Fig. 2.2 Tipuri de RNCG a) feed-forward; b) cascadă; c) recurentă**

În Fig.2.2a ieșirile unităților dintr-un strat  $k$ ,  $\Pi_k$ , sunt cuplate la intrările unităților din stratul următor  $\Pi_{k+1}$ , caracterizând o RNCG de tip feed-forward. În Fig.2.2b, într-o RNCG de tip cascadă, ieșirile unităților stratului  $k$  sunt conectate nu numai la intrările stratului  $k+1$ , ci și la cele ale straturilor consecutive,  $k+2$ ,  $k+3$ , șamd. RNCG recurentă din Fig.2.2c. are conexiuni bidirecționale și/sau o buclă de reacție.

Unitățile de procesare au ca indici coordonatele spațiale. O celulă din stratul  $k$  al unei RNCG  $n$  dimensionale cu  $m$  straturi se notează  $C_{i_1, i_2, \dots, i_n, k}$  unde  $i_j \in \{1, 2, \dots, N_j\}$  pentru toți  $j \in \{1, 2, \dots, n\}$  și  $k \in \{1, 2, \dots, m\}$ . În particular pentru  $n=1$  o celulă se notează  $C_{i, k}$  respectiv pentru  $n=2$ ,  $C_{i, j, k}$ .

### Definiția 2

*Vecinătatea*  $N_{i, l}$  din stratul  $l$ , al unei celule  $C_i$  din stratul  $k$ , al unei RNCG  $n$  dimensionale, cu  $m$  straturi se definește prin relația:

$$N_{i, l} = \{C_v / d_{k, l}(i; v) \leq r_{k, l}, i \neq v = (v_1, v_2, \dots, v_n, l) \\ v_j \in \{1, 2, \dots, N_j\} \text{ ptr. } \forall j \in \{1, 2, \dots, n\} \} \quad (2.1)$$

unde:

- $d_{k, l}(i, v)$  este distanța metrică (definiția 1 din Anexa 1) dintre vectorii de numere întregi
- $(i_1, i_2, \dots, i_n, k)^T$  și  $(v_1, v_2, \dots, v_n, l)^T$ ;
- $r_{k, l}$  este raza vecinătății, un număr pozitiv întreg;

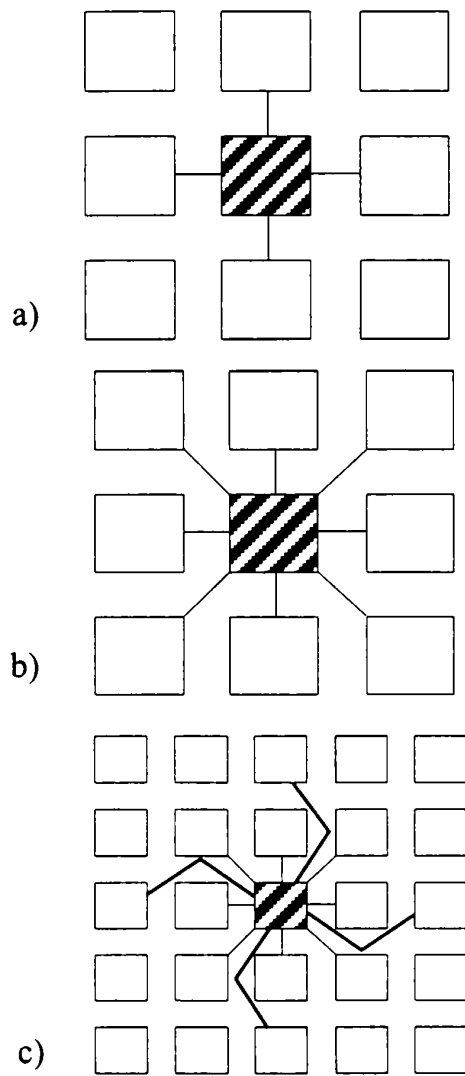
Se pot alege distanțe metrice diferite, pentru diferite valori  $k$  și  $l$ . Setul celor mai apropiate vecinătăți pentru o celulă  $C_i$ , este dat de reuniunea tuturor vecinătăților  $N_{i, l}$ , din toate straturile.

Se observă că în cazurile extreme, vecinătatea cea mai apropiată a unei celule poate fi mulțimea vidă (nici o conexiune cu celulele învecinate) sau poate conține toate celulele RNCG, ca în cazul rețelei Hopfield. În cazul general, se pot alege  $r$  și  $d$  diferite pentru aceeași celulă dintr-un strat, și/sau perechi de straturi. Acest caz generează o topologie neuniformă, ceea ce nu este de dorit pentru o arhitectură neuronală. Vom considera în cele ce urmează rețele cu aceleași valori  $r_{k, k}$  și  $d_{k, k}$  pentru celulele aparținând unui strat  $k$ , și respectiv aceleași valori  $r_{k, l}$  și  $d_{k, l}$  pentru celulele din două straturi diferite  $k, l$ .

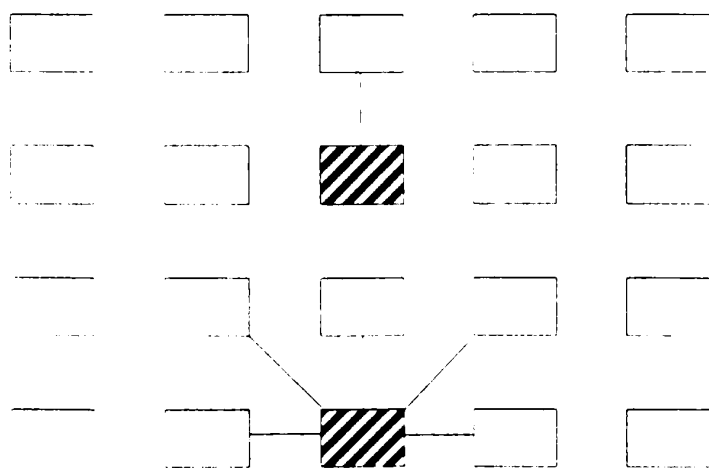
Fig.2.3 prezintă câteva tipuri de conexiuni intrastrat pentru o rețea bidimensională.

În Fig.2.3a conexiunile sunt date de  $r_{k, k}=1$  și  $d_{k, k}=|i_1 - v_1| + |i_2 - v_2|$ , în Fig.2.3b de  $r_{k, k}=1$  și  $d_{k, k}=\max(|i_1 - v_1|, |i_2 - v_2|)$  și în Fig.2.3c de  $r_{k, k}=2$  și  $d_{k, k}=|i_1 - v_1| + |i_2 - v_2|$ ;

Fig. 2.4 prezintă câteva posibilități de interconectare a unei celule între straturi. Vecinătățile sunt date pentru cazul din Fig.2.4a de  $r_{k, k+1} = 1$  și  $d_{k, k+1} = |i_1 - v_1| + 1$ , respectiv în Fig.2.4b de  $r_{k, k+1} = 1$  și  $d_{k, k+1} = \max(|i_1 - v_1|, 1)$ .



**Fig. 2.3 Modalități de conectare a unei celule intrastrat**



**Fig. 2.4 Modalități de conectare a unei celule între straturi pentru o RNCG unidimensională cu două straturi**

## 2.1.2 Structura unei celule

Fiecare celulă este un circuit neliniar, dinamic, de un ordin oarecare, astfel încât topologia circuitului și/sau valorile elementelor pot diferi de la o celulă la alta.

Fig.2.5 reprezintă schema bloc a unei celule. Unitatea de intrare este un circuit rezistiv liniar care însumează ponderat intrările și ieșirile celulelor învecinate .

A doua unitate este un circuit liniar dinamic de ordin  $n_i$ , cu o singură intrare  $e_i$  și o singură ieșire  $\xi_i$ . Unitatea de ieșire trece intrarea  $\xi_i$ , printr-o neliniaritate  $f_i(\cdot)$ , generând semnalul  $y_i$ . Sistemul de ecuații de mai jos descrie funcționarea celulei:

$$\dot{x}_i(t) = a_i x_i(t) + b_i e_i(t)$$

$$\xi_i(t) = c_i^T x_i(t) + h_i e_i(t)$$

$$y_i(t) = f_i(\xi_i(t))$$

(2.2)

$$e_i(t) = \sum_{v \in N_i} A_{iv} \cdot y_v(t - \tau_{iv}) + \sum_{v \in N_i} B_{iv} u_v(t - \sigma_{iv}) + I_i$$

unde:

- $a_i \in \mathbb{R}^{n_i \times n_i}$ ,  $b_i, c_i \in \mathbb{R}^{n_i}$ ,  $h_i, A_{i,v}, B_{i,v}, \tau_{i,v}, \sigma_{i,v}$ ;
- $I_i \in \mathbb{R}$  sunt toate constante;
- $x_i(\cdot): \mathbb{R} \rightarrow \mathbb{R}^{n_i}$ ;  $e_i(\cdot), \xi_i(\cdot), y_i(\cdot): \mathbb{R} \rightarrow \mathbb{R}$  sunt funcții de timp;
- $x = dx/dt$  și  $f_i(\cdot): \mathbb{R} \rightarrow \mathbb{R}$  sunt funcții neliniare;
- $N_i$  este mulțimea vecinătăților celulei  $C_i$ ;

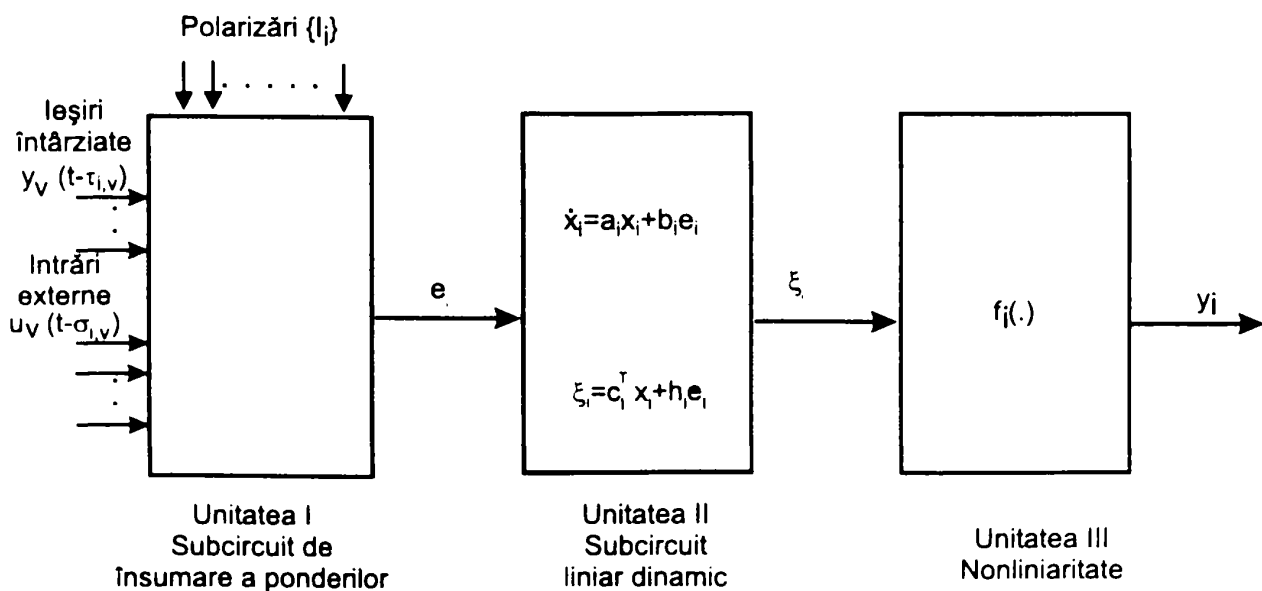


Fig. 2.5 Schema bloc a unei celule

După cum se observă fiecare celulă are două tipuri diferite de intrări  $u_i(t)$  și  $I_i$ . Intrarea de control  $u_i$  se aplică ponderat prin  $B_{iv}$  celulei  $C_i$  și celulelor vecine, în timp ce polarizarea  $I_i$  se aplică numai celulei  $C_i$ . Timpii de întârziere  $\tau_{iv}$  și  $\sigma_{iv}$  sunt introduși pentru a obține un model mai complex, realist, reflectând timpul de propagare necesar semnalului de reacție  $y_{iv}$  și a intrărilor de control  $u_{iv}$  de la celulele vecine pentru acțiunea asupra celulei  $C_i$ .

Pentru celulele aparținând vecinătății  $N_i$ , ponderile conexiunilor de control  $B_{iv}$ , respectiv cele de reacție  $A_{iv}$  se pot reprezenta sub formă unor matrici. Aceste matrici de conexiune pot fi aceleași sau pot diferi de la o celulă la alta.

### Definiția 3

O RNC cu matrici de control, reacție și polarizare invariante la transformările de coordonate, adică, independente de  $i$ , se numește *invariantă în spațiu*.

Rețelele feed-forward multistrat și rețelele Hopfield, abordate în Referatul 1 [5], pot fi considerate cazuri particulare ale acestui model. În cele ce urmează vom detalia studiul a două cazuri particulare, RNC analogice și RNC discrete.

## 2.2 Rețele neuronale celulare analogice

### 2.2.1 Dinamica rețelei

Rețelele neuronale celulare analogice (RNCA) au fost primele introduse, de L. Chua și L. Yang în 1988 [28]. Față de cazul RNCG se fac următoarele simplificări:

- rețeaua are straturi bidimensionale. Dimensiunea unui strat  $k$  fiind de  $M \times N$  celule, vom nota o celulă cu  $C_{i,j}$  unde  $i \in \{1, 2, \dots, M\}$  și  $j \in \{1, 2, \dots, N\}$ ;
- subcircuitul liniar dinamic este de ordinul întâi,  $t_i = 1$ ;
- neliniaritatea aplicabilă în unitatea de ieșire este o funcție continuă, liniară pe porțiuni, definită de relația:

$$f(x) = \frac{1}{2}(|x_i + 1| - |x_i - 1|) \quad (2.3)$$

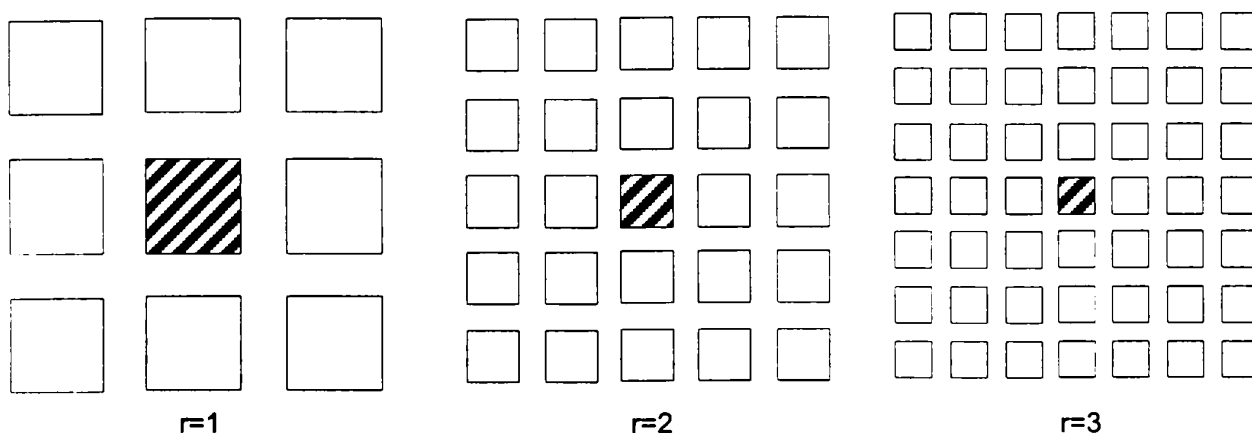
unde:  $|\cdot|$  reprezintă valoarea absolută ;

- $b_i = c_i = 1, h_i = 0$  ;
- constantele  $a_i = -a < 0$  sunt aceleași pentru toate celulele;
- fiecare celulă este conectată într-o vecinătate  $N_i(i,j)$  dată de distanța metrică  $d(i,j;k,l)$ :



$$d(i, j; k, l) = \max(|i - k|, |j - l|) \leq r \quad (2.4)$$

După cum se vede în Fig.2.6 vecinătatea unei celule pentru  $r=1$  conține 8 celule, pentru  $r=2$  conține 24 de celule, ș.a.m.d.



**Fig.2.6 Vecinătatea unei celule pentru  $r=1, r=2, r=3$**

În aceste condiții sistemul de ecuații (2.2) se poate scrie sub forma:

**Ecuația de stare**

$$C x_{i,j}(t) = -\frac{1}{R_x} x_{i,j}(t) + \sum_{C_{k,l} \in N_r(i,j)} A_{i,j;k,l} y_{k,l}(t) + \sum_{C_{k,l} \in N_r(i,j)} B_{i,j;k,l} u_{k,l}(t) + I$$

pentru  $1 \leq i, k \leq M$  și  $1 \leq j, l \leq N$

$$\text{Ecuația de intrare: } u_{i,j} = E_{i,j} \quad (2.5)$$

$$\text{Ecuația de ieșire: } y_{i,j}(t) = \frac{1}{2} (|x_{i,j}(t) + 1| - |x_{i,j}(t) - 1|)$$

$$\text{Condiții de stare: } |y_{i,j}(0)| \leq 1; |u_{i,j}| \leq 1$$

$$\text{Condiții asupra parametrilor: } A_{i,j;k,l} = A_{k,l;i,j}; C \geq 0, R \geq 0$$

unde:

- $u$  este intrarea independentă;
- $x$  este starea celulei;
- $y$  este ieșirea;
- $N_r$  este vecinătatea de rază  $r$  a celulei;
- $A$  este matricea de conexiune de reacție,  $B$  este matricea de conexiune de control și  $I$  matricea de polarizare;
- $R$ , și  $C$  sunt constante pozitive;

## 2.2.2. Structura unei celule

Sistemul descris de ecuațiile (2.5) poate fi modelat prin circuitul din Fig.2.7

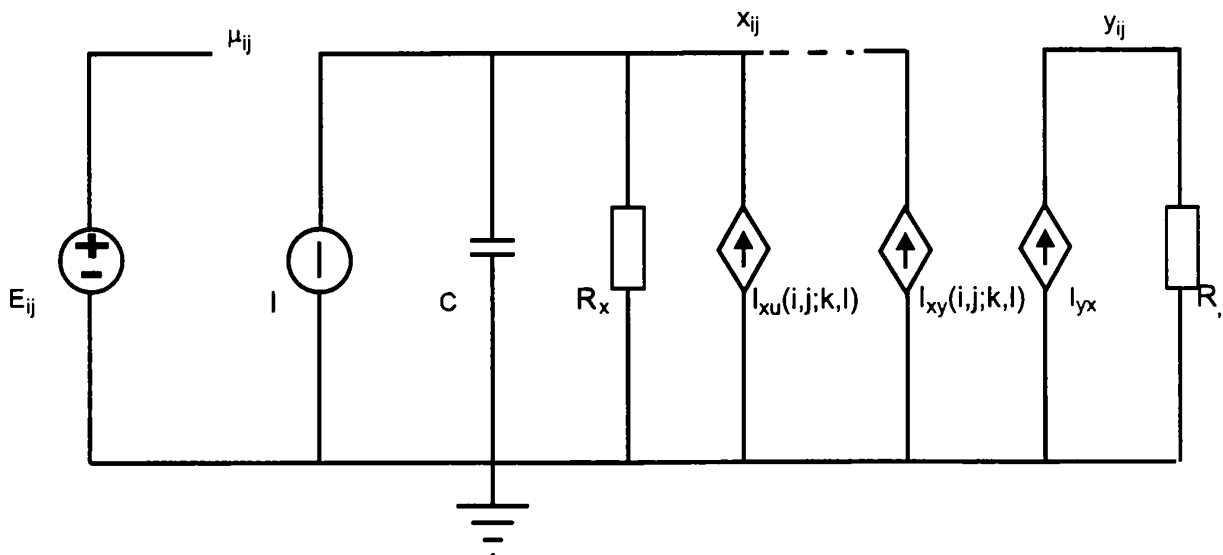


Fig. 2.7 Un exemplu de circuit pentru o celulă  $C_{i,j}$

Fiecare circuit  $C_{i,j}$  conține o sursă independentă de tensiune,  $E_{ij}$ , o sursă independentă de curent,  $I$ , o capacitate liniară  $C$ , două rezistențe liniare  $R_x$  și  $R_y$  și cel mult  $2m$  ( $m$  reprezintă numărul celulelor învecinate) surse liniare de curent controlate în tensiune, cuplate la celulele vecine prin tensiunea de control  $u_{kl}$  și reacția de la tensiunea de ieșire  $y_{kl}$  a tuturor celulelor vecine  $C_{kl}$ .

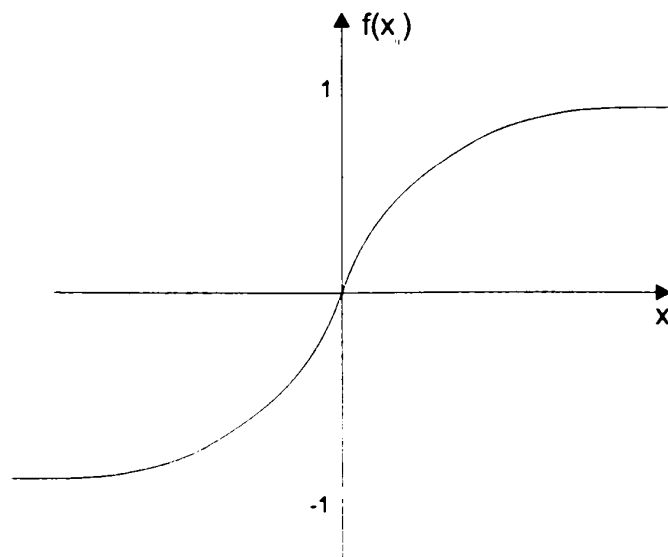


Fig. 2.8 Caracteristica sursei neliniare de curent controlate în tensiune

În particular, sursele de curent controlate în tensiune sunt liniare  $I_{xy}(ij,kl) = A_{ij,kl}$  și  $I_{xu}(ij,kl) = B_{ij,kl}$ . Singurul element neliniar este o sursă de curent controlată în tensiune  $I_{yx} = (1/R_y) f(x_{i,j})$  cu caracteristica  $f(\cdot)$  dată în Fig.2.8.

În practică, amplitudinea semnalului poate fi normalizată pentru a îndeplini condițiile parametrilor. Valorile rezistențelor pot fi convenabil alese, ele determinând și puterea disipată de circuit. Rezistențele  $R_x$  și  $R_y$  au valori cuprinse între  $1k\Omega$  și  $1M\Omega$ .  $CR_x$  este constanta de timp a circuitului și este uzual aleasă de ordinul  $10^{-5}$  -  $10^{-8}$ s.

## 2.3 Rețele neuronale celulare discrete

### 2.3.1 Dinamica rețelei

Aproximând ecuația diferențială de stare din sistemul (2.5) cu o ecuație cu diferențe finite, unde  $t=nT$ ,  $T$  fiind o constantă unitară de timp se poate obține sistemul de ecuații (2.6), care caracterizează o rețea neuronală în timp discret (RNCD) [47], [48]:

$$x_{i,j}[n+1] = \sum_{C_{k,l} \in N_r(i,j)} A_{i,j;k,l} y_{k,l}[n] + \sum_{C_{k,l} \in N_r(i,j)} B_{i,j;k,l} u_{k,l}[n] + I$$

ptr.  $1 \leq i \leq M$  ;  $1 \leq j \leq N$

(2.6)

$$y[n+1] = f(x[n]) = \begin{cases} 1 & \text{dacă } x[n] \geq 1 \\ -1 & \text{dacă } x[n] \leq -1 \end{cases}$$

Spre deosebire de RNCA sistemul discret are doar ieșiri binare ponderate prin operatorul de reacție. Ieșirea binară a unei celule este determinată de semnul tensiunii de stare a celulei, din iterația anterioară și este nedeterminată pentru o tensiune nulă. Practic însă, există întotdeauna un zgomot care determină o stare diferită de zero. Algoritmul este relativ insensibil la variațiile date de toleranța parametrilor reacției, controlului și polarizării, dacă este îndeplinită condiția:

$$\Delta = \min_{C_{k,l} \in N_r(i,j)} \left| \sum_{C_{k,l} \in N_r(i,j)} A_{(i,j;k,l)} y_{k,l}[n] + \sum_{C_{k,l} \in N_r(i,j)} B_{(i,j;k,l)} u_{k,l}[n] + I \right|$$

ptr.  $1 \leq i \leq M$  ;  $1 \leq j \leq N$

(2.7)

Comparativ cu modelul analogic, cel în timp discret are câteva avantaje importante:

- Datorită utilizării semnalelor binare, interconectarea mai multor chipuri este foarte simplă.
- Insensibilitatea parametrică dată de ecuația (2.7) determină robustețea rețelei la toleranțele de fabricație, dacă ponderile conexiunilor au fost proiectate corect.
- Viteza de propagare poate fi controlată într-un domeniu larg schimbând doar frecvența tactului. Aceasta ușurează și testarea circuitului.

### 2.3.2 Structura unei celule

Circuitul care modelează algoritmul dat prin ecuațiile (2.6) este prezentat în Fig.2.9.

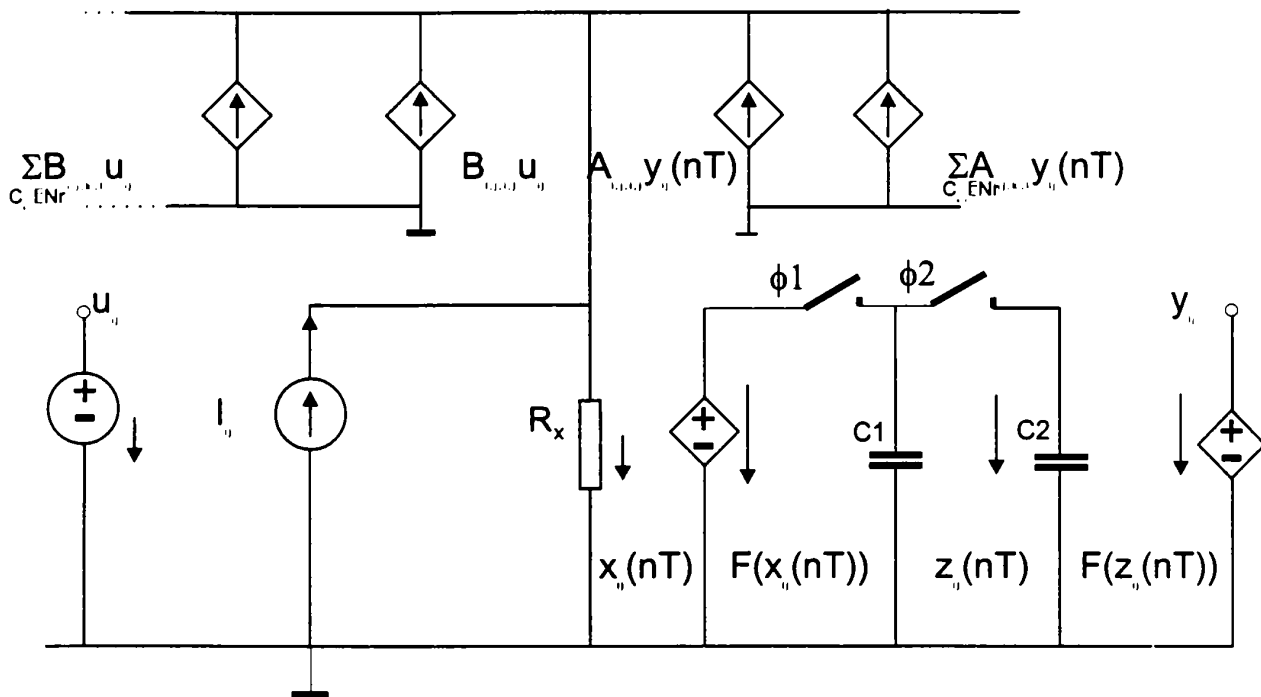


Fig. 2.9 Structura unei rețele neuronale celulare discrete

Variabilele sunt tensiuni în diferite momente de timp discret. Pasul de iterație este dat de perioada semnalelor  $\phi_1$  și  $\phi_2$ . Neglijând tranzițiile parazite, momentul  $nT$  poate fi ales arbitrar în timpul fazei de nivel "high" a semnalului  $\phi_1$  (Fig.2.10).

Multiplicarea este realizată de sursele liniare de curent controlate în tensiune. Curentul sumă este transformat în tensiunea  $x(nT)$  de rezistența  $R_x$ :

$$x_{i,j}(nT) = R_x \left( \sum_{C_{k,l} \in N_r(i,j)} A_{i,j;k,l} y_{k,l}(nT) + \sum_{C_{k,l} \in N_r(i,j)} B_{i,j;k,l} u_{k,l} + I \right) \quad (2.8)$$

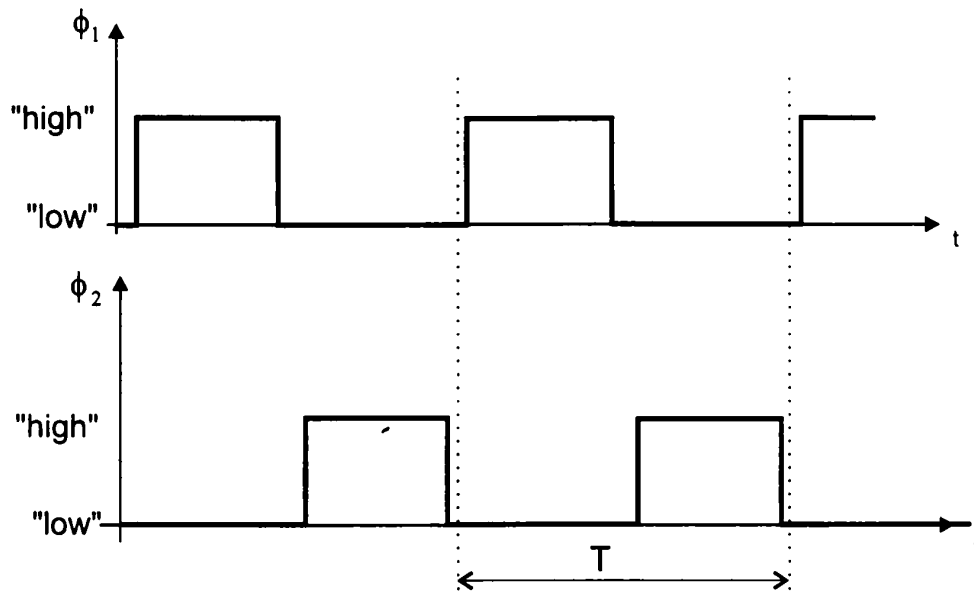
$$\text{ptr. } 1 \leq i \leq M ; 1 \leq j \leq N$$

Sursele de tensiune controlate în tensiune,  $F(x_{i,j}(nT))$  și  $F(z_{i,j}(nT))$  au caracteristică neliniară:

$$F(v) = \begin{cases} V_{\text{sat}} & \text{daca } v > 0 \\ -V_{\text{sat}} & \text{daca } v < 0 \end{cases} \quad (2.9)$$

În timpul fazei de nivel "high" a semnalului  $\phi_1$  condensatorul  $C_1$  se încarcă la tensiunea  $V_{\text{sat}}$  sau  $-V_{\text{sat}}$ . Pe durata fazei "high" a semnalului  $\phi_2$ , ( $\phi_1$  "low"), condensatoarele  $C_1$  și  $C_2$  sunt conectate în paralel. Tensiunea  $z_j(nT)$  se stabilizează după tranziție la:

$$z(nT) = \frac{C_1 F(x_{i,j}(n-1)T) + C_2 z_{i,j}(n-1)T}{C_1 + C_2} \quad (2.10)$$



**Fig. 2.10** Diagrama semnalelor de tact pentru rețelele neuronale celulare discrete

Deoarece  $|F(x_{i,j})| = V_{sat}$  și  $|z_{i,j}| \leq V_{sat}$ , pentru orice pas de iterație  $n$ , rezultă pentru  $C_1 > C_2$  că:

$$y_{ij}(nT) = F(x_{ij}(n-1)T) = \begin{cases} V_{sat} & \text{daca } x_{ij}(n-1)T > 0 \\ -V_{sat} & \text{daca } x_{ij}(n-1)T < 0 \end{cases} \quad (2.11)$$

Înlocuind  $x_{i,j}(n-1)T$  dat de ecuația (2.8), în relația (2.11), obținem:

$$y_{i,j}(nT) = F\left(R_x \left( \sum_{C_{k,l} \in N_r(i,j)} A_{i,j;k,l} y_{k,l}((n-1)T) + \sum_{C_{k,l} \in N_r(i,j)} B_{i,j;k,l} u_{k,l} + I \right)\right) \quad (2.12)$$

ptr.  $1 \leq i \leq M$ ;  $1 \leq j \leq N$

Prin normalizarea tensiunilor la  $V_{sat}$ , a curenților la  $V_{sat}/R_x$  și a elementelor pasive la  $R_x$  se obține sistemul de ecuații (2.6).

Pentru circuitul prezentat literatura oferă structuri analogice echivalente eficiente.

# Capitolul 3

## ASPECTE ALE PROIECTĂRII RETELELOR NEURONALE CELULARE ȘI MODALITĂȚI DE IMPLEMENTARE

### 3.1 Metode de determinare a matricilor de conexiune

Problema esențială care se pune în proiectarea RNC este determinarea matricilor de interconexiune, astfel încât rețeaua neuronală să implementeze o anumită aplicație.

Dinamica complexă a celulelor în interacțiune este determinată de matricile interconexiunilor de control, de reacție și de polarizare.

În majoritatea aplicațiilor în care RNC funcționează ca operatori (în prelucrările de imagini), aceste matrici sunt invariante în spațiu. Din punctul de vedere al programabilității, matricile ponderilor reprezintă subrutine. Ele pot lua valori discrete prestabilite sau pot fi controlate în timp. Întrucât schimbarea ponderilor poate fi realizată mult mai repede decât stabilizarea rețelei, se pot utiliza și matrici de conexiune variabile în timp.

Literatura consemnează mai multe metode de determinare a matricilor de conexiune între care:

- Translatarea automatului celular, a filtrării bidimensionale și a procesărilor de imagine de tip convoluție în matrici de conexiuni a fost utilizată cu succes încă de la apariția RNC [29].
- Utilizarea metodelor analitice, prin luarea în considerare a unor efecte dinamice simple, este o altă modalitate de construire a matricilor pondere, aplicată cu succes în modelul retinei și în microscopia electronică (la deblurring) [82], [70].
- Rezolvarea unui sistem de inegalități care asigură condiția ca ieșirea dorită să fie un punct de echilibru. Metoda nu garantează oprirea tranziției în minimul global, corespunzător ieșirii dorite. Procesarea se poate opri în minime locale. Metoda este adecvată în cazul în care condițiile inițiale nu sunt prea îndepărtate de ieșirea dorită.

- Minimizarea unor funcții de cost constituie ideea de bază a mai multor algoritmi de obținere a unor matrici de conexiuni adecvate. Cea mai simplă metodă abordează diferența dintre ieșirea dorită și ieșirea curentă .

### Observații:

- Funcția de cost poate fi dificil de minimizat, ea putând avea mai multe minime locale sau chiar globale.
- Dacă nu este garantată stabilitatea RNC aceasta poate oscila sau deveni chiar haotică.
- Alte dificultăți pot fi generate de faptul că funcția de cost nu este diferențiabilă.

Probleme importante sunt convergența algoritmului, unimodalitatea funcției de cost, selectarea unui set de date de antrenament adecvate pentru a asigura generalizarea.

Au fost dezvoltate versiuni modificate ale algoritmului cu retropropagare recurentă a erorii (recurrent back-propagation ) și ale celui cu retropropagare a erorii în timp (back-propagation through time) pentru a asigura nu numai evoluția înspre puncte fixe de echilibru, ci și după o anumită traiectorie.

Tehnicile de tip gradient necesită intuiție și experiență în alegerea adecvată a modelelor de antrenament și a dinamicii locale. Pentru RNCD în [64] se folosește o metodă de optimizare pentru determinarea traiectoriei, fiind în acest sens, singura de tip global. Pentru această metodă problema care se pune este câte modele sunt necesare pentru o bună generalizare. În [100] se obține și o limită superioară a numărului de eșantioane.

- Utilizarea algoritmilor genetici. Algoritmii genetici au performanțe bune în cazul în care operează cu puțini parametri liberi (ponderile interconexiunilor) și multe variabile dependente (valorile de ieșire ale pixelilor) [56], [94]. Ei pot găsi minimumul global în spații de căutare zgomotoase și discontinue, fără a utiliza informația de diferențiere a funcției de cost.

Diferite tipuri de ecuații diferențiale parțiale pot fi utilizate în forma lor discretizată spațial pentru a obține matricile de interconexiune . Cazuri speciale interesante sunt undele autogeneratoare (autowave) și modelul difuziei, ce utilizează următorul tip de ecuație:

$$\frac{\partial I(x, y, t)}{\partial t} = \mathbf{D} \cdot \text{divgrad}(g(I(x, y, t))) + \mathbf{F}(I(x, y, t)) \cdot \mathbf{L} \quad (3.1)$$

unde :

- $I(x,y,t)$  este o funcție de intensitate spațio-temporală;
- $g(.)$  și  $F(.)$  sunt funcții neliniare;
- $\mathbf{D}$  și  $\mathbf{L}$  sunt constante;

- In cazul RNC cu mai multe straturi se utilizează metode care combină în serie sau în paralel efectele cunoscute ale unor operatori simpli (considerând matricile de conexiuni subrutine) pentru a obține algoritmi analogici complecși. Acești algoritmi se pot reprezenta prin diagrame, limbaj de programare sau cod mașină și necesită intuiția și experiența proiectantului pentru descompunerea în elemente componente a aplicației.

Optimizarea ulterioară a parametrilor nominali a RNC, în raport cu toleranța parametrilor și a perturbațiilor modelelor de intrare reprezintă o altă problemă, un prim pas în luarea în considerare a constrângerilor de implementare hard. Pentru implementările VLSI concrete sunt posibile simplificări importante care pot fi luate în considerare în faza de proiectare [77].

Prin experiență, îmbinând analiza cu metodele euristice se poate dobândi o oarecare îndemânare.

### 3.1.1 Algoritmi pentru convergența la puncte fixe

Pentru simplitate se va adopta cazul unei rețele bidimensionale de dimensiune  $M \times N$ , relațiile extinse fiind valabile în cazul general multidimensional. Fiind dat modelul de ieșire dorit, pentru fiecare celulă  $C_{ij}$  ( $i=1,2,\dots,M, j=1,2,\dots,N$ ) aparținând unei RNCA de dimensiune  $M \times N$ , în regiunea de saturație  $|y_{i,j}^{\infty}| = 1$  și intrarea  $u_{ij}$ , se pune problema determinării matricilor de conexiune, astfel încât starea corespunzătoare  $|x_{i,j}^{\infty}| = 1$  să fie dată de:

$$x_{ij}^{\infty} = \sum_{kl \in N_r(ij)} A_{ij,kl} \cdot y_{kl}^{\infty} + \sum_{kl \in N_r(ij)} B_{ij,kl} \cdot u_{kl} + I_{ij} \quad (3.2)$$

iar derivata  $|x_{i,j}^{\infty}| = 0$ . Condițiile de convergență spre starea dorită  $y_{i,j}^{\infty}$  sunt:

$$\begin{aligned} x_{ij}^{\infty} &\geq +1 \quad \text{daca} \quad y_{ij}^{\infty} = +1 \\ x_{ij}^{\infty} &\leq -1 \quad \text{daca} \quad y_{ij}^{\infty} = -1 \end{aligned} \quad (3.4)$$

Cele două condiții se pot restrânge în inegalitatea:

$$y_{ij}^{\infty} \left( \sum_{kl \in N_r(ij)} A_{ij,kl} \cdot y_{kl}^{\infty} + \sum_{kl \in N_r(ij)} B_{ij,kl} \cdot u_{kl} + I_{ij} \right) \geq 1 \quad (3.4)$$

În general există  $P$  perechi de modele antrenament (modele de intrare  $u_{i,j}^{(p)}$  - modele de ieșire  $y_{i,j}^{(p)}$ ),  $P > 1$  ( $p=1,2,\dots,P$ ) și pentru fiecare se poate obține o inegalitate de forma (3.4). Sistemul de inegalități obținut se poate rezolva prin mai multe metode, ca de exemplu: metoda relaxării [66], algoritmul perceptronului [45], algoritmul ADATRON [73].

În [35] este evitată rezolvarea sistem de inegalități, care reprezintă partea cea mai laborioasă, prin eliminarea din start a zonelor care nu pot conține stări stabile. Toată procesarea se reduce la o simplă testare de semn și o căutare eficientă într-un arbore binar.



Pentru fiecare dintre algoritmi anterior menționați există o teoremă de convergență care garantează găsirea unei soluții dacă aceasta există.

În unele aplicații (ca de ex. procesarea de imagini) este necesară invarianța la rotație, sau izotropia matricilor de conexiune. Toți algoritmi anterior menționați pot fi modificați pentru a încorpora aceste constrângeri suplimentare.

Relațiile (3.2)-(3.4) date pentru rețelele neuronale analogice sunt adevărate și pentru cele discrete dacă se ia în considerare un moment de timp discret  $n$  suficient de mare ca rețeaua să se stabilizeze într-un punct fix.

În ambele cazuri nu s-au luat în considerare în procesul de învățare condițiile inițiale pentru stare  $x(0)$  și pentru ieșire  $y(0)$ , motiv pentru care nu există un control al bazinelor de atracție. În literatură se consemnează rezultate pozitive în această direcție, valabile doar în cazul în care tranzițiile rețelei sunt simplu monotone [77].

### 3.1.2 Metode de tip gradient pentru învățare de traiectorii prescrise pentru rețelele neuronale celulare analogice

Proiectarea evoluției înspre puncte fixe nu garantează comportarea corectă a sistemului dinamic, din moment ce nu întotdeauna stările inițiale se găsesc în bazinele de atracție ale stării de ieșire dorite.

Este necesară găsirea unui vector  $s$  ( $A, B, I$ ) astfel încât ieșirea RNC  $d^{(p)}(\infty)$  să fie cea dorită, pentru o stare inițială  $x^{(p)}(0)$  și o intrare dată  $u^{(p)}$ , pentru toate perechile de modele de antrenament ( $p=1,2, \dots, P$ ).

Uzual se definește o eroare sau o funcție de cost depentă de punctele fixe și de ieșirile dorite (retropropagarea recurentă a erorii - recurrent backpropagation) sau în general de traiectoria sistemului și traiectoria dorită (retropropagarea erorii în timp - backpropagation through time):

$$E(s) = \sum_{p=1}^P \sum_{i=1}^M \sum_{j=1}^N E_{ij}(x_{ij}^p, d_{ij}^p) \quad (3.5)$$

Derivata funcției de mai sus în raport cu ponderile poate fi utilizată pentru găsirea minimului local al erorii:

$$\frac{dE}{ds} = \sum_{p=1}^P \sum_{i=1}^M \sum_{j=1}^N \frac{dE_{ij}}{ds} (x_{ij}^p, d_{ij}^p) \quad (3.6)$$

Pentru simplitate se va omite indexul  $p$ , deoarece oricum gradientul se însumează pentru toate perechile de modele de antrenament. Datorită liniarității pe porțiuni a funcției de ieșire este mai bine să se definească eroarea ca o funcție de stare cu un parametru  $R$  [2], [89]:

$$e(v|R) = \begin{cases} \frac{1}{k} |v - (1 + R)|^k & \text{daca } v < (1 + R) \\ 0 & \text{in rest} \end{cases} \quad (3.7)$$

Funcția de eroare devine:

$$E(x_{ij}, d_{ij}) = e(x_{ij}d_{ij} | \mathbf{R}) \quad (3.8)$$

și derivata ei:

$$\frac{dE}{ds} = \frac{de(x_{ij}d_{ij} | \mathbf{R})}{dx_{ij}} \cdot d_{ij} \cdot \frac{dx_{ij}}{ds} \quad (3.9)$$

Parametrul  $R$  definește robustețea în spațiul stărilor, ceea ce înseamnă că eroarea raportată la o celulă va fi zero ori de câte ori ieșirea ei ajunge la cel puțin o distanță  $R$  față de limita regiunii de saturație.

**Retropropagarea recurentă a erorii (RRE)** este o generalizare a algoritmului cu retropropagare a erorii (backpropag). Eroarea se raportează la punctele fixe de stabilitate (presupunând că acestea sunt atinse):

$$E_{ij}(s) = e(x_{ij}(\infty)d_{ij} | \mathbf{R}) \quad (3.10)$$

și ecuațiile devin:

$$\frac{dE_{ij}}{ds} = \lambda_{ij} \cdot \frac{dF_{ij}}{ds} \Big|_{t \rightarrow \infty} \quad (3.11)$$

unde:

- $F_{ij}$  este membrul drept al ecuației de stare;
- $\lambda_{ij} \in \mathbb{R}^{M \times N}$  este un vector de eroare, procesat pentru sistemul dinamic asociat pornind de la orice condiție inițială  $\lambda_{ij}(0)$ ;

$$\dot{\lambda}_{ij} = -\lambda_{ij} + \sum_{kl \in N_r} A_{ij,kl} \cdot f'(x_{ij}(\infty)) \cdot \lambda_{kl} + e'(x_{ij}(\infty)d_{ij} | \mathbf{R}) \cdot d_{ij} \quad (3.12)$$

Sistemul asociat dinamic este introdus numai pentru a evita inversarea unei matrici. Dacă algoritmul găsește un vector parametru adecvat, el nu va învăța doar punctele fixe dorite ale sistemului dinamic, ci și traiectoriile de la o anumită stare inițială la acestea.

Problema algoritmului RRE este că RNC devine instabilă în timpul procedurii de învățare, dacă eșuează în găsirea soluției. Pentru a evita acest lucru, a fost introdusă retropropagarea erorii în timp [77].

### Retropropagarea erorii în timp (RET)

Acest algoritm permite atât învățarea punctelor de ieșire dorite cât și a unor traiectorii prescrise. Eroarea devine:

$$E_{ij}(s) = e_1(x_{ij}(T)d_{ij} | \mathbf{R}) \int_0^T e_2(x_{ij}(t)d_{ij} | \mathbf{R}) dt \quad (3.13)$$

și gradientul erorii:

$$\frac{dE}{ds} = \lambda \cdot \frac{dF_{ij}}{ds} \Big|_{t \rightarrow \infty} \quad (3.14)$$

utilizându-se sistemul asociat dinamic descris de :

$$\dot{\lambda}_{ij} = -\lambda_{ij} + \sum_{kl \in N_r} A_{ij,kl} \cdot f'(x_{ij}(t)) \cdot \lambda_{kl} - e_1(x_{ij}(T)d_{ij}|\mathbf{R}) \cdot e_2(x_{ij}(t)d_{ij}|\mathbf{R})d_{ij} \quad (3.15)$$

Sistemul asociat dinamic trebuie integrat înapoi în timp din moment ce valoarea  $\lambda_{ij}$  este cunoscută la momentul T:

$$\lambda_{ij}(T) = e_1(x_{ij}(T)d_{ij}|\mathbf{R}) \cdot d_{ij} \int_0^T e_2(x_{ij}(t)d_{ij}|\mathbf{R}) dt \quad (3.16)$$

Dependent de alegerea lui  $e_1$  respectiv  $e_2$  algoritmul poate învăța o traiectorie prescrisă la ieșirea dorită sau obține informații din traiectorie pentru a găsi un vector parametru pentru a atinge ieșirea dorită într-un interval de timp dat T.

### Observații

1. Toate tehnicile menționate necesită intuiția și experiența proiectantului pentru alegerea adecvată a modelelor de antrenament și specificarea dinamicii locale.

2. O problemă comună tuturor algoritmilor de tip gradient este cea a minimelor locale ale funcției de eroare. Astfel încât rezultatele sunt dependente de parametrii inițiali. Pentru evitarea minimelor locale se folosesc metodele clasice, aplicabile în cazul rețelelor neuronale, în general:

- selectarea aleatorie a ordinii modelelor de antrenament. Această metodă are un mecanism intrinsec de ieșire din minimele locale, dar existența mai multor minime globale poate genera probleme de convergență;
- repetarea învățării cu parametrii inițiali aleși aleator, până se atinge un minim global al erorii;

3. Rezultatele RRE și RET sunt superioare în raport cu varianta clasică RPE. Pentru ambele versiuni există și variante aplicabile RNCD.

### 3.1.3 Metode de învățare de traiectorii prescrise în rețelele neuronale celulare discrete

Metodele de tip gradient nu sunt aplicabile în cazul RNCD, deoarece gradientul erorii nu există peste tot în spațiul parametrilor rețelei, datorită funcției hard limitatoare (signum) utilizate.

Avantajul este că tranziția de la momentul n la momentul n+1 poate fi descrisă prin inegalități liniare. Fiind dată intrarea u și y(0) se aplică recursiv ecuația de stare și se obține următorul set de inegalități, în fiecare moment de timp n=0, ..., T-1:

$$\begin{aligned} x_{ij}[n] \geq 0 & \text{ dac\u0103 } y_{ij}[n+1] = +1 \\ x_{ij}[n] \leq 0 & \text{ dac\u0103 } y_{ij}[n+1] = -1 \end{aligned} \quad (3.17)$$

Din care se ob\u021bine :

$$y_{ij}[n+1] \cdot \left( \sum_{kl \in N_r(ij)} A_{ij,kl} \cdot y_{ij}[n] + \sum_{kl \in N_r(ij)} B_{ij,kl} \cdot u_{kl} + I_{ij} \right) > 0 \quad (3.18)$$

Se observ\u0103 c\u0103 exist\u0103 mai multe traiectorii posibile \u0219i valoarea 0 se poate \u00eenlocui \u00een rela\u021bii cu o constant\u0103 R pentru a asigura robuste\u021tea re\u021elei. Solu\u021bia sistemului de inegalit\u0103\u021bi nu se schimb\u0103, are loc doar o scalare a solu\u021biei sistemului original. Aceasta reflect\u0103 faptul c\u0103 spa\u021biul solu\u021biilor unui sistem general de inegalit\u0103\u021bi de forma (3.18) este un con convex :

$$\left\{ \mathbf{s} \in \mathbf{R}^{M \times N} : \mathbf{s}^{[p]} \cdot \mathbf{v}^{[p]} > R \geq 0; \mathbf{v}^{[p]} \in \mathbf{R}^{M \times N} \text{ pentru } 1 \leq p < P \right\} \quad (3.19)$$

Prin cre\u0219terea valorii lui R v\u0103rful conului se \u00eendep\u0103rteaz\u0103 de origine. O defini\u021bie a robuste\u021bii solu\u021biei s \u0219i a modului \u00een care se poate ob\u021bine cea mai robust\u0103 solu\u021bie este dat\u0103 \u00een \u0224 3.2.

### 3.1.4 Algoritmi globali pentru re\u021elele neuronale celulare discrete

Algoritmii cu \u00eenv\u0103tare global\u0103 pentru RNCD [76] proiecteaz\u0103 traiectoria, ei \u00een s\u0103\u0219i, printr-o procedur\u0103 de optimizare, determin\u0103nd parametrii re\u021elei care transform\u0103 un set de modele de intrare \u00eentr-un set de modele de ie\u0219ire dorite. Spre deosebire de algoritmii locali la care are semnifica\u021bie intrarea \u00eentr-o celul\u0103, aici imaginile de intrare se aplic\u0103 \u00entregii re\u021ele.

Probleme mai complexe pot fi rezolvate astfel \u0219i ob\u021binute traiectorii mult mai complicate. Dezavantajul acestei metode este faptul c\u0103 se pare c\u0103 apar\u021bine problemelor de tip combinatorial \u0219i deci costul computa\u021bional este ridicat.

Toate variantele au la baz\u0103 ideea definirii unei func\u021bii de cost care se minimizeaz\u0103, determin\u0103nd \u00eenv\u0103tarea.

RNCD au dou\u0103 comport\u0103ri posibile:

- fie converg spre un punct de echilibru stabil;
- fie oscileaz\u0103 periodic;

\u00c\n majoritatea aplica\u021biilor nu sunt tolerate oscila\u021biile, de aceea ele trebuie penalizate prin func\u021bia obiectiv.

Fie s vectorul parametru care con\u021bine ponderile interconexiunilor RNCD.

Se define\u0219te o distan\u021b\u0103 metric\u0103  $\Delta^{[p]}(s)$  \u0219i o func\u021bie de cost  $o(s)$  astfel:

$$\Delta^{[p]}(s) = \begin{cases} \frac{1}{4} \sum_i \sum_j \omega_{ij} (y_{ij,s}^{[p]}(\infty) - d_{ij}^{[p]}) & \text{pentru punctele de echilibru stabil} \\ 1 & \text{pentru oscila\u021biile periodice} \end{cases} \quad (3.20)$$

$$\mathbf{o}(\mathbf{s}) = \sum_{p=1}^P \Omega_p \cdot \Delta_p(\mathbf{s}) \quad (3.21)$$

unde:  $\omega_{ij} \in [0,1]$ ,  $\Omega \in [0,1]$  și reprezintă factori de ponderare normalizați care satisfac relațiile:

$$\sum_{i=1}^M \sum_{j=1}^N \omega_{ij} = 1 \quad (3.22)$$

$$\sum_{p=1}^P \Omega_p = 1 \quad (3.23)$$

În unele aplicații pot fi tolerate oscilații moderate. În acest caz, se utilizează o distanță metrică modificată  $\Delta_{|\mathbf{s}|}(\mathbf{s})$  în care se face o mediere într-un interval de timp al perioadei oscilațiilor.

Datorită neliniarității funcției signum a celulelor unei RNCD, funcția obiectiv are o proprietate nedorită: are platouri multidimensionale de valori constante și discontinuități (pante abrupte între acestea). Deci derivata funcției obiectiv este fie zero (pe platouri), fie nedefinită (pe discontinuități) astfel încât metodele clasice de optimizare ce utilizează derivate sunt inaplicabile.

Există și în acest caz mai multe alternative pentru soluționarea problemei:

- metode de optimizare care nu utilizează informație de gradient pentru minimizarea funcției obiectiv  $\mathbf{o}(\mathbf{s})$  [64];
- înlocuirea funcției neliniare de tip signum cu o funcție sigmoidă cu câștig variabil [64]. Sistemul devine astfel unul dinamic diferențiabil, pentru care metodele clasice de optimizare pot fi aplicate. Ideea este de a porni cu un câștig mic pentru funcția sigmoidă și de a găsi minimumul funcției de cost. Câștigul este apoi crescut puțin și din nou se minimizează funcția obiectiv. Procedura se repetă până când câștigul e foarte mare și funcția sigmoidă aproximează cu o anumită eroare neliniaritatea signum;
- aplicarea metodei “răcirii simulate” (simulated annealing) pornind de la observația că problema de optimizare are un spațiu al stărilor finit și poate deci fi abordată ca o problemă de optimizare combinatorială. Experiența a demonstrat că această metodă deși costisitoare computațional este robustă și găsește soluții bune chiar și în cazurile dificile [64];
- algoritmi genetici au fost și ei testați în învățarea globală atât pentru RNCA cât și pentru RNCD. Rezultatele nu sunt edificatoare. S-a stabilit însă faptul că decisivă pentru succes este codarea coeficienților, ceea ce este o direcție de cercetare deschisă;

Algoritmii cu învățare globală nu sunt înlocuitori ai algoritmilor locali, ci metode complementare pentru învățare în RNCD. Selecția adecvată a modelelor de antrenament reprezintă o problemă crucială pentru algoritmii globali. Rezultatele obținute depășesc performanțele algoritmilor locali.

### 3.1.5 Algoritmi orientați spre implementarea hard

Metodele de tip gradient prezentate sunt valabile pentru orice sistem descris de o ecuație generală de forma:

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{s}) \quad (3.24)$$

unde ideal  $\mathbf{F}$  este membrul drept al ecuației de stare a unei RNC.

Orice implementare fizică reală reprezintă o variantă a acestei ecuații, fiind posibile simplificări în cazul în care există descrierea exactă a circuitului real. Literatura consemnează o serie de realizări în această direcție [63], [69].

### 3.1.6. Algoritmi genetici

Proprietățile semnificative ale algoritmilor genetici sunt următoarele:

- operează cu o codare binară a setului de parametri (și nu cu parametri propriu-ziși);
- căutarea are loc într-un număr de puncte a spațiului parametrilor (ce formează o populație);
- în procesul de optimizare utilizează doar funcția de cost (nu derivata ei sau altă informație);
- utilizează reguli de tranziție probabilistice;

Algoritmii genetici sunt capabili să opereze cu funcții de cost dificil de minimizat, nediferențiable, cu minime locale sau chiar mai multe minime globale. Stabilitatea rețelei nu trebuie asigurată prin constrângeri. Performanțele sunt remarcabile în cazul în care funcționează cu puțini parametri liberi (valorile interconexiunilor) și mulți parametri dependenți (valorile pixelilor de ieșire). Pe măsură ce numărul parametrilor crește performanța rețelei scade.

Modul de codare al parametrilor reprezintă cheia succesului. "Codarea comună" eșuează în cazul unui număr ridicat de parametri și în minime locale. Rezultatele obținute cu codarea "încrucișarea aleatoare" (random crossover) nu sunt influențate de numărul parametrilor și performanțele obținute în cazul unor probleme complexe sunt încurajatoare. Au apărut și algoritmi genetici hibridi care încorporează și operatori specifici aplicației.

Prin utilizarea algoritmilor genetici au fost generate matrici de interconexiune pentru operatori simpli de imagine [56] (mediere, detecția unghiurilor convexe, determinarea

umbrei, conversia imaginilor cu nuanțe de gri în imagini alb-negru “half-toning” și invers, “inverse half-toning”) și au fost soluționate probleme complexe ce necesită o procesare masiv paralelă, cum este clasificarea și detecția texturii materialelor, extragerea defectelor din acestea [94].

### 3.2 Robustețea

Se poate discuta despre:

- robustețea relativă în spațiul modelelor de intrare respectiv în spațiul ponderilor;
- robustețea absolută;

Problema a fost abordată într-o serie de cazuri concrete, atât pentru RNCD cât și pentru RNCA (în absența reacției inverse de la celulele vecine), cu matrici de interconexiune invariante în spațiu [47].

Există mai multe modalități de definire a robusteții unei soluții  $s$  în raport cu diferite norme în spațiul  $\mathbf{R}^{M \times N}$ :  $s \in \mathcal{S} : \{s \in \mathbf{R}^{M \times N} : s^t v^{[p]} \geq 0; v^{[p]} \in \mathbf{R}^{M \times N} \text{ pentru } 1 \leq p \leq P\}$ .

De exemplu, robustețea relativă în spațiul vectorilor  $r_w(s)$  în raport cu norma euclidiană  $\|\cdot\|$  este definită de relația [92]:

$$\max(r) \text{ în raport cu } \forall \Delta s \in \mathbf{R}^{M \times N} : \|\Delta s\| = r\|s\| \Rightarrow (s + \Delta s) \in \mathcal{S} \quad (3.25)$$

Robustețea absolută  $R_w(s)$  se definește ca o soluție a următoarei probleme de optimizare:

$$\max_s r_w(s) = \max_s \min_{p=1,2,\dots,P} \frac{s^t v^{[p]}}{\|s\| \|v^{[p]}\|} \quad (3.26)$$

Într-un mod similar se poate defini și robustețea relativă în spațiul modelelor de intrare.

Se poate demonstra că  $r_w(s)$  este distanța minimă a vectorului  $s$   $\|\Delta s\|$  la planele definite de modelele  $v^{[p]}$ . Cea mai robustă soluție  $s^*$  se poate obține rezolvând ecuația:

$$\min \|s\| \text{ în raport cu } s^t v^{[p]} \geq \|s^{[p]}\| \text{ pentru } p = 1, 2, \dots, P \quad (3.27)$$

Evident soluția nu este unică, din moment ce o scalare pozitivă nu influențează robustețea. De aceea se poate adăuga o constrângere suplimentară  $\|s\| = 1$  problemei de optimizare. Se poate demonstra că dacă problema este solvabilă, obiectivul și constrângerile pot fi schimbate între ele, obținându-se o problemă echivalentă de programare pătratică cu constrângeri sub forma unor inegalități lineare:

$$\min \|s\| \text{ în raport cu } s^t v^{[p]} \geq \|s^{[p]}\| \text{ pentru } p = 1, 2, \dots, P \quad (3.28)$$

Soluția obținută este cea mai insensibilă atât la toleranțele ponderilor RNC datorate implementării hard imperfecte cât și la perturbațiile modelelor de intrare.

Algoritmul "Adaptive perceptron" găsește soluția optimă a ponderilor [76] RNCD și în cazul matricilor de interconexiune rotațional invariante pe o vecinătate  $r$ , pentru RNCA bidimensionale.

### Definiția 3

O matrice de conexiuni este *rotațional invariantă* dacă orice rotație  $k \cdot 90^\circ$ ,  $k \in \mathbb{N}$ , în jurul centrului și orice reflexie pe diagonală (față de centru) o lasă neschimbată.

În plus se demonstrează că dacă aplicația este invariantă în raport cu un grup, și ponderile optime obținute vor fi invariante [73]. Ca un efect colateral este posibil ca ponderile optime să fie invariante în raport cu un grup mai mare de transformări.

Problema robusteții a fost abordată și în cazul aplicațiilor de tip "câștigătorul ia totul" recent dezvoltate pe structura neuronală celulară [59]. În versiunea originală, dimensiunea rețelei era limitată la un număr de zece neuroni pentru a admite toleranțe de fabricație rezonabile. Ulterior, a fost propusă o extensie a algoritmului ce utilizează o funcție de activare unipolară și întreruperea temporară a reacției inverse (clipping feedback) ceea ce conferă rețelei robustețe atât la variațiile relative cât și ale celor absolute ale parametrilor rețelei.

## 3.3 Domeniul dinamic

### Definiția 4

*Domeniul dinamic* este valoarea maximă absolută a variabilelor de stare  $M \geq |x_{ij}(t)|$

În [28] s-a demonstrat că  $M$  este suma valorilor absolute a tuturor matricilor de conexiune plus unu:

$$M = x_{\max} = 1 + R_v |I| + R_x \max_{C_{k,t} \in \mathbb{N}_r(i,j)} (|A_{i,j;k,t}| + |B_{i,j;k,t}|) \quad (3.29)$$

O condiție similară s-a obținut și în cazul matricilor de conexiune neliniare [30] și cu întârziere în timp [82]. Fiind dat domeniul dinamic pentru o tehnologie dată 5V sau 15V, în tehnologia VLSI CMOS se poate determina domeniul valorilor interconexiunilor, pentru funcționarea corectă a unei RNC.



### 3.4 Stabilitatea

Stabilitatea este o problemă cheie a proiectării RNC.

#### Definiția 5

O RNC este *complet stabilă* dacă ea converge spre o stare cu o valoare constantă stabilă, pentru orice intrare și orice stare inițială.

Una dintre metodele eficiente în demonstrarea stabilității unei rețele neuronale celulare, aplicabilă și în cazul RNC, este metoda Lyapunov.

Un sistem dinamic este stabil dacă se poate găsi o funcție Lyapunov care caracterizează funcționarea sistemului, descrescătoare de-a lungul tuturor traiectoriilor posibile, adică  $\dot{E}(t) \leq 0$ . Funcția Lyapunov asociată unei rețele neuronale este o funcție de energie  $E(t)$ , dependentă de parametrii rețelei, similară cu energia sistemelor fizice. Pentru RNC se definește următoarea funcție Lyapunov:

$$E(\vec{y}) = -\frac{1}{2} \sum_{i,j} \sum_{C_{kl} \in N_r(i,j)} A_{ij,kl} y_{ij}(t) y_{kl}(t) + \frac{1}{2 \cdot R} \sum_{i,j} y_{ij}(t)^2 - \sum_{i,j} \sum_{C_{kl} \in N_r(i,j)} B_{ij,kl} y_{ij}(t) u_{kl}(t) - \sum_{i,j} I \cdot y_{ij}(t) \quad (3.30)$$

În articolul original [27] și în [44] se demonstrează că simetria matricii de reacție este o condiție suficientă ca această funcție să fie monoton descrescătoare, deci să fie asigurată stabilitatea RNCA.

În [44] se demonstrează în plus că pozitivitatea interconexiunilor este o condiție suficientă de stabilitate. S-a demonstrat și modul în care stabilitatea matricilor de conexiune simetrice și pozitive poate fi extinsă și la alte categorii prin transformări adecvate simple [86]. Pentru matricile de interconexiune cu semne opuse s-au găsit nu numai condițiile în care se generează stabilitate ci și comportare haotică [82]. În [25] se dă o condiție pentru stabilitatea completă a RNCD cu întârziere. S-a găsit un set de condiții în care întârzierea nu generează instabilitate (ca de ex. în cazul interconexiunilor pozitive ca semn). S-a demonstrat și că matricile simetrice ca semn sunt stabile aproape peste tot. S-a studiat posibilitatea existenței punctelor multiple de echilibru.

S-a demonstrat posibilitatea comportării oscilatorii și haotice, care au condus la noi aplicații.

Stabilitatea este, în concluzie, o problemă relativ bine fundamentată teoretic, pentru mai multe categorii de RNC. Pentru unele matrici de conexiune, experimental stabile, nu există încă o demonstrație a stabilității.

### 3.5 Generalizarea

#### Definiția 6

**Generalizarea** reprezintă abilitatea unei rețele neuronale de a opera corect și cu date de intrare care nu au fost incluse în setul de antrenament.

În contextul învățării din exemple unul dintre aspectele esențiale este determinarea numărului datelor de antrenament necesare pentru o generalizare corectă.

Ca o măsură a puterii de procesare a rețelei s-a introdus dimensiunea Vapnik-Chervonenkis [53].

#### Definiția 7

**Dimensiunea Vapnik-Chervonenkis** (VCdim) este dimensiunea celui mai mare set al datelor de antrenament VCdim, pentru care o rețea are puterea de a implementa  $2^{|S|}$  dihotomii în spațiul modelelor de ieșire, unde  $|S|$  reprezintă numărul eșantioanelor de antrenament din S.

Datorită similarității cu arhitectura rețelelor neuronale multistrat și a invarianței matricilor de interconexiune s-a determinat pentru o RNCD, cu NI intrări într-o celulă, măsura Vapnik-Chervonenkis [101]:

$$\text{VC dim} \leq \text{NI} + 1 \quad (3.31)$$

Considerând r raza vecinătății, se poate calcula NI,  $\text{NI} = 2 \cdot (2r + 1)^2$  și VC dim devine:

$$\text{VC dim} \leq 2(2r + 1)^2 + 1 \quad (3.32)$$

Un model matematic care operează într-un cadru probabilistic al învățării și generalizării este teoria aproximării probabil corecte (probably approximately correct PAC). Odată ce VCdim este cunoscută, aplicarea teoriei PAC permite calculul unei valori utile în practică numită, “complexitatea eșantioanelor” (sample complexity), o măsură puterii de procesare a rețelei, a abilității de a realiza transformări arbitrare asupra spațiului modelelor de intrare.

#### Definiția 8

**Complexitatea eșantioanelor** este o limită superioară a numărul datelor de antrenament necesare pentru o generalizare de încredere.

Datorită recurenței, RNCD poate realiza mai multe transformări decât o rețea feed-forward cu elemente liniare cu prag.

Presupunând funcționarea corectă a rețelei pentru datele de antrenament, complexitatea eșantioanelor  $m_0(\delta, \epsilon)$  ne dă o limită superioară a numărului modelelor de intrare pentru o generalizare corectă cu o eroare  $\epsilon$ , cu o probabilitate ce tinde spre cea a certitudinii, cu o eroare  $\delta$  (în condițiile în care atât datele de antrenament cât și cele de testare au o aceeași distribuție).

Adică dacă (pentru două valori  $\epsilon, \delta > 0$ ) la intrarea RNCD se aplică  $m \geq m_0(\delta, \epsilon)$  modele de antrenament atunci probabilitatea ca rețeaua să clasifice corect toate modelele de intrare cu excepția a  $\epsilon$  modele tinde spre certitudine cu o eroare  $\delta$ , ori de câte ori  $m \geq m_0(\delta, \epsilon)$ .

În literatură [101] se studiază două cazuri:

- rețeaua implementează transformări globale de la o intrare independentă constantă  $u(0)$  și o stare inițială  $y(0)$  la o ieșire  $y(T)$  după  $T$  pași:

$$m_0(\delta, \epsilon) \leq \frac{4}{\epsilon} \left[ (NI + 1) \lg \frac{12}{\epsilon} + \lg \frac{2}{\delta} \right] \quad (3.33)$$

- rețeaua învață o traiectorie a vectorului de ieșire  $y(t)$  discretizată pe axa timpului în  $T$  pași ( $y(1), y(2), \dots, y(T)$ ) pornind de la intrarea  $u(0), u(1), u(2), \dots, u(T-1)$  și starea de ieșire inițială  $y(0)$ .

Datorită naturii discrete și a actualizării paralele a strategiei de învățare a RNCD problema învățării unei traiectorii poate fi descompusă într-o problemă de  $T$  tranziții globale a  $M \times N$  celule.

$$m_0(\delta, \epsilon) \leq \frac{4}{\epsilon} \left[ \frac{1}{T \cdot M \cdot N} (NI + 1) \lg \frac{12}{\epsilon} + \lg \frac{2}{\delta} \right] \quad (3.34)$$

Cele două erori  $\delta, \epsilon$  pot fi estimate utilizând teoria PAC, inclusiv în condiții de zgomot [96].

### 3.6 Implementarea hard și soft

Încă de la apariție RNC au fost strict legate de implementarea fizică concretă. Primele încercări au fost VLSI, imediat după introducerea conceptului, deși nici implementările optice nu s-au lăsat prea mult așteptate. În prezent implementările VLSI au o tehnologie relativ bine fundamentată. Elementul esențial este transconductanța care realizează ponderea conexiunii.

#### 3.6.1 Implementarea soft

Pachetul de programe disponibil conține trei simulatoare, ce pot rula pe orice calculator IBM-AT 80286, sau compatibil [114], [115]:

- un simulator soft în virgulă fixă;
- un simulator soft în virgulă mobilă;

- un simulator pentru RNC variante în spațiu;

Cele trei simulatoare sunt un mediu de programare de tip user-friendly. Caracteristica lor esențială este posibilitatea de simulare a rețelelor multistrat. Cele două simulatoare soft oferă ceva mai multe facilități și o mai mare flexibilitate, dar la o viteză și la o capacitate de memorare mai mici decât cel bazat pe acceleratorul hard multiprocesor. Simulatorul în virgulă flotantă oferă precizie mai mare, manipulează conexiuni cu ponderi cu valori mai mari, cu prețul unei capacități de memorare mai mici decât cel în virgulă fixă. O variantă mai nouă permite și implementarea unor matrici de interconexiune variante în spațiu [115].

Simulatoarele pot implementa RNC cu mai multe straturi, atât analogice cât și discrete, conectate între ele arbitrar (numărul maxim al straturilor este de zece, pe simulatorul soft). Densitățile celulelor pot diferi de la un strat la altul. Conexiunile modelate pot fi liniare, neliniare sau cu întârziere în timp. Se pot utiliza succesiuni de matrici de conexiune în algoritmi complecși. Pentru descrierea acestor algoritmi se utilizează un limbaj simplu de programare. Se pot procesa operații logice și aritmetice. Se recunosc patru tipuri diferite de formate de fișiere pentru imagini. Se pot crea fișiere care să permită restaurarea mediului și chiar să redea evoluția întregului experiment (fără o nouă procesare). Se pot afișa pe ecran mai multe imagini, tranzițiile celulelor selectate, informațiile cele mai importante despre simularea curentă. Se poate utiliza și un adaptor multidisplay pentru maximum patru monitoare.

### 3.6.2 Acceleratorul hard

Realizările analogice VLSI sunt cele mai performante în ceea ce privește viteza de procesare. Numărul relativ redus al celulelor implementate pe chip (până în prezent de ordinul miilor), programabilitatea limitată, restrânge însă complexitatea aplicațiilor.

Dezvoltarea acceleratorului hard [85], cu 1 milion de pixeli, sub forma PC add-on board, transformă instantaneu un calculator personal într-o stație de lucru (workstation) capabilă să emuleze RNC la o mare viteză și la o fracțiune din prețul de cost al unui calculator de mare viteză care procesează identic.

Acceleratorul hard utilizează circuite integrate VLSI și reprezintă cea mai bună soluție din punctul de vedere al programabilității, flexibilității, complexității și al prețului de cost.

Arhitectura acceleratorului conține 4 DSP standard, pentru calculul răspunsului tranzitoriu al unei RNC cu un strat, ce are  $0,25 \cdot 10^6$  -  $0,75 \cdot 10^6$  celule neuronale analogice (dependent de tipul matricii de conexiune). Arhitectura și principiile de proiectare sunt independente de numărul procesoarelor

### 3.6.3 Workstation pentru rețele neuronale celulare

Continua dezvoltare a mediilor de programare existente a dus la conceperea unei arhitecturi ce conține [112]:

- un scanner optic;
- o videocameră interfațată cu calculatorul printr-un frame grabber;
- grafică interactivă;
- fișiere de tip ASCII;
- un simulator soft de uz general cu o bibliotecă de imagini și templates;
- un simulator dedicat chipului cu logică programabilă duală;
- un simulator bazat pe acceleratorul hard, care utilizează numere în virgulă mobilă;
- acceleratorul hard;

Sistemul de tip workstation este deci un pachet integrat hard-soft ce poate fi instalat pe orice computer compatibil IBM PC-AT, fiind un instrument util pentru proiectarea de noi algoritmi de învățare și matrici de conexiune în RNC .

### 3.6.4 RNC cu ponderi fixe

Conexiunile locale și structura geometrică regulată a RNC sunt două avantaje majore pentru implementarea fizică. În literatura de specialitate s-au publicat mai multe variante pentru versiunea analogică.

Primul chip testat, în 1991, a înregistrat o performanță remarcabilă, o constantă de timp de 100 ns la o densitate de 32 celule/mm<sup>2</sup>. Viteza de procesare de 1 Terra operații pe secundă calculată la o suprafață de siliciu de 1 cm<sup>2</sup> este remarcabilă. Versiunea analogică are și avantajul unui număr mai mic de componente, și deci o suprafață redusă a chipului (deoarece tranzistorele MOS nu sunt utilizate numai ca și comutatoare) [52]. Pe de altă parte, RNCA sunt mult mai sensibile la perturbații decât RNC discrete. O altă direcție de cercetare este deci implementarea discretă. Pentru versiunea discretă a RNC s-a realizat un chip în tehnologia capacităților comutate. Performanța cea mai bună raportată fost o frecvență de procesare de 10 Mz.

În prezent este disponibil un prim chip 80170 al companiei Intel, care implementează transconductanțe programabile.

Implementarea optică asigură procesarea cu viteze foarte mari (apropriate de viteza luminii). S-au realizat optic doar matrici de conexiune directe, cele de reacție fiind implementate cu componente discrete numerice.

### 3.6.5 RNC dual programabile

În 1990 a fost realizat primul chip programabil de RNC. Acesta combină operațiile logice cu cele analogice realizând o procesare duală. Spre deosebire de procesarea hibridă, aici nu există convertoare (care ar putea introduce erori). Singura codare este cea spațială, prin intermediul conexiunilor. În chipul de RNC cu logică duală, matricile de

conexiune pot fi programate continuu sau discret, și controlate de o logică standard. Se pot implementa algoritmi complecși, care conțin succesiuni de matrici de conexiune serie, sau paralel, prin memorarea locală a rezultatelor intermediare.

### 3.6.6 Mașina universală RNC

Cea mai semnificativă generalizare a RNC este cea mai recentă invenție din domeniu, mașina universală RNC (MU-RNC) și supercomputerul [81], [82], care încorporează la nivelul fiecărei celule, într-o mică suprafață a chip-ului, o memorie analogică locală, o memorie logică locală și o unitate de control și comunicare pentru interfațarea locală și cea globală. Această adevărată revoluție în concepție transformă RNC cu interconexiuni programabile într-un program memorat de RNC care moștenește caracteristicile programabile ale unui computer digital, dar are avantajul procesării în timp real. Au fost dezvoltate un limbaj RNC [102] și un sistem de operare RNC care permit utilizatorului scrierea unui program pentru procesarea unei imagini printr-o succesiune de straturi RNC, fiecare având un alt tip de conexiuni, pentru a îndeplini o anumită sarcină .

Deoarece fiecare “template” necesită pentru procesare 1  $\mu$ s, un mare număr de matrici de conexiune succesive poate fi procesat într-un interval de timp real.

Circuitele de memorare logică și analogică ocupă o suprafață foarte mică a chip-ului, deoarece sunt necesare doar câteva locații de memorie - ele sunt similare registrelor și bufferelor dintr-un calculator clasic. Spre deosebire de varianta clasică unde fiecare cuvânt este memorat ca un șir de numere binare, aici fiecare cuvânt din memoria logică conține o singură stare (“Da” sau “Nu”) Această memorie logică “primitivă” este elementul cheie al mașinii universale RNC.

Programul memorat pentru mașina universală RNC este numit “Analog-Logic” deoarece implică atât operații analogice (cu dinamică în timp real) cât și logice. O estimare grosieră a performanțelor mașinii demonstrează o capacitate de procesare mai mare decât supercomputerele actuale de scop general. S-a demonstrat că mașina universală RNC este o mașină Turing analogică [86], ea fiind capabilă să rezolve orice problemă ce poate fi soluționată cu un calculator clasic.

Mașina universală RNC poate fi realizată ca un chip monolitic. Acest chip nu e cu nimic mai prejos decât microprocesorul care a dus la expansiunea industriei calculatoarelor. Foarte probabil că MU RNC va fi utilizată ca un microprocesor analogic într-o nouă generație de computere inteligente cu procesare în timp real.

Ea este singura rețea neuronală cu o dinamică similară funcțiilor de “tip analogic” implementate în emisfera dreaptă a creierului și a funcțiilor de “tip discret” implementate în emisfera stângă a creierului uman. Cu alte cuvinte, mașina universală RNC este o primă încercare grosolană a implementării asimetriei creierului uman.

Este previzibilă apariția unei noi industrii de supercomputere bazate pe MU-RNC care să prelucreze informația într-o manieră similară creierului uman.

### 3.7 Concluzii

În acest capitol au fost trecute succint în revistă aspectele fundamentale ale proiectării RNC și modalitățile de implementare soft și hard ale acestora.

S-au prezentat diferite metode locale și globale de determinare a interconexiunilor. Trebuie remarcat faptul că ponderile optime odată determinate pentru o aplicație rămân fixe, ceea ce evident facilitează implementarea fizică. Reprezentarea datelor de intrare într-o formă adecvată, respectiv codarea lor, încorporarea unei informații locale specifice aplicației, sunt cheia succesului și, din păcate deocamdată mai mult o artă decât o știință. De aceea familiaritatea cu domeniul de aplicație, respectiv experiența în domeniul proiectării RNC, este esențială. Metoda “trial and error” a devenit un principiu al cercetării. Performanțele algoritmilor de tip global sunt superioare celor locali. În ultimii ani au apărut mai multe versiuni ale algoritmilor globali. Ei nu vor fi înlocuitori ai algoritmilor locali, ci complementari, datorită avantajelor implicate de cei din urmă.

Robustețea și stabilitatea sunt două aspecte importante în proiectarea unei RNC, abordate până în prezent doar în câteva cazuri concrete. O serie de modele viabile practic nu și-au găsit încă fundamentarea teoretică.

Pentru a putea face o comparație, timpul de procesare pentru diferitele implementări ale RNC, pentru o mie de celule este:

- 0.1– 1  $\mu$ s pentru circuitele VLSI analogice;
- 1–10 ms, pentru acceleratorul hard;
- 1–10 s pentru soft;

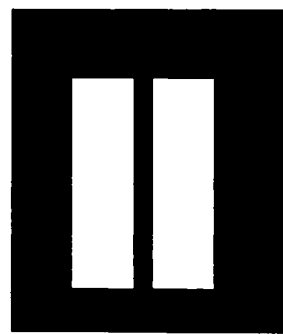
Pe lângă criteriul viteză de procesare, în alegerea unei anume modalități de implementare, într-o aplicație concretă, trebuie avute în vedere toate celelalte avantaje și dezavantaje, care au fost deja expuse.

Realizarea într-un chip monolitic a mașinii universale RNC și în perspectivă a unui supercomputer, al cărui element de bază va fi, deschide calea obținerii unui imens potențial de procesare și posibilitatea realizării unor noi și diverse aplicații .

În viitor, ca dispozitive de procesare a RNC, ar putea fi utilizate orice materiale cu proprietăți spațio-temporale descrise prin ecuații parțiale diferențiale, ce demonstrează o dinamică asemănătoare RNC, în forma lor discretizată spațial. Aspecte esențiale sunt modalitatea de furnizare a semnalului de intrare, respectiv de preluare a celui de ieșire. Promițătoare în această direcție sunt unele fenomene chimice care demonstrează capabilitatea de a procesa imagini. Țesuturile vii cu o structură geometrică regulată ar putea fi și ele utilizate în implementarea RNC .

Dezvoltarea unor algoritmi de învățare adaptivi și utilizarea în implementare a tehnologiilor optice, electromagnetice, biologice și cuantice va avea cu siguranță implicații spectaculoase asupra aplicațiilor RNC.

.....  
**Partea**



**Retele neuronale celulare  
,  
ca operatori**



# Capitolul 4

## REȚELELE NEURONALE CELULARE CA OPERATORI

### 4.1 Aplicații posibile

Există două direcții de proiectare a RNC în funcție de natura aplicației, care consideră întregul sistem ca un operator spațial, respectiv ca o memorie.

RNC, prin structura geometrică, spațial regulată, reprezintă cadrul ideal pentru procesările de imagini. Sute de publicații, rodul unei intense activități de cercetare au apărut în ultimii ani, având ca obiect cele mai diverse aplicații ale RNC ca și operatori locali de imagini:

- prelucrările simple de imagini: extragerea de contur, filtrarea, detectarea de contur, detecția de colțuri concave, de colțuri convexe, skeletonizarea imaginilor alb-negru și a celor cu nuanțe de gri, sortarea după orientare, după dimensiune, accentuarea contrastului, mărirea și micșorarea de imagine;
- detectarea de caracteristici (de exemplu detectorul de componente conectate, "connected component detector", care dă proiecția figurii, de exemplu pe orizontală, pe verticală, pe direcția de  $+45^\circ$  sau  $-45^\circ$ );
- îmbunătățirea rezoluției;
- eliminarea zgârieturilor din imagini, care pot proveni dintr-o fotocopiere defectuoasă;
- compresia și decompresia imaginilor și reconstrucția lor bidimensională și tridimensională;
- conversia imaginilor cu nuanțe de gri în imagini alb negru (halftoning) și invers, a imaginilor alb negru în imagini cu nuanțe de gri (inverse halftoning);
- clasificarea texturilor ;
- clasificarea obiectelor dintr-o imagine, după profunzime, direcție, mărime;
- numărarea obiectelor;

- determinarea drumului cel mai scurt dintre două puncte, cu evitarea punctelor interzise;
- simularea difuziei controlate;
- împiedicarea falsificării bancnotelor prin fotocopiere;
- simularea unor iluzii optice (de exemplu iluzia Muller-Lyer);
- implementarea unor funcții arbitrare booleene;
- analiza și detecția mișcării;
- extragerea unui anumit model dintr-o imagine statică sau în mișcare ;

RNC poate realiza orice transformări care implică convoluții și deconvoluții spațiale, fiind practic un filtru trece sus, local, în domeniul Fourier.

Unele dintre aplicațiile enumerate pot fi implementate printr-un singur template, care încorporează acțiunea unor operatori elementari (ca de exemplu clasificarea texturilor se realizează prin acțiunea unui operator care are atât funcția de detector de componente conectate cât și de conversie a imaginilor cu nuanțe de gri în imagini alb-negru), printr-o procesare masiv paralelă.

## 4.2 Analiza și detecția mișcării

Analiza și detecția mișcării este una dintre preocupările permanente ale cercetătorilor, datorită implicațiilor potențiale în întreaga sferă a activității umane.

Este incitant modul elegant, rapid, prin care evoluția ființelor vii a rezolvat problema detecției și estimării mișcării. O descoperire faimoasă a neurologilor Hubel și Wiesel arată că deplasarea unei bare într-o anumită regiune a câmpului vizual al unei pisici, într-o anumită direcție și cu o viteză în jurul unei anumite valori determină activarea unor anumiți neuroni în cortexul vizual.

Rezultatele cercetărilor arată că RNC au capacitatea abordării problemei într-o manieră similară neuronilor corticali.

Din punctul de vedere al RNC în analiza și detecția mișcării există cinci tipuri de probleme [78]:

- P.1.** Determinarea existenței mișcării având o viteză dată într-o anumită direcție;
- P.2.** Detecția existenței mișcării cu o valoare absolută a vitezei ce se încadrează într-un domeniu dat (direcția deplasării nu contează);
- P.3.** Estimarea vitezei mișcării după o direcție dată (orizontală sau verticală);
- P.4.** Estimarea componentelor  $v_x$  respectiv  $v_y$  atunci când obiectul se deplasează într-o direcție oarecare;
- P.5.** Separarea părților mobile respectiv a celor statice ale imaginilor în mișcare.

Abordarea analizei și detecției mișcării se poate face în două tipuri de circumstanțe posibile:

- Cazul ideal când fondul este alb iar obiectul în mișcare negru;
- Cazul real când trebuie să se țină cont de efectele disturbante posibile: imagini pe scara nuanțelor de gri, prezența reflexiilor de lumină, a zgomotului și blurring-ului, existența unor mici perturbații în direcția de mișcare, umbre sau incertitudini datorate rezoluției finite;

RNC poate funcționa în două moduri, în funcție de tipul imaginii de intrare, ce poate fi continuă sau eșantionată.

Literatura de specialitate oferă soluții pentru problemele P1, P2, [79], [84], atât pentru imaginile continue cât și pentru cele eșantionate. Există rezultate promițătoare și în detecția obiectelor în mișcare aflate într-un mediu zgomotos [84]. Separarea părții mobile a imaginilor ce conțin obiecte în mișcare a fost de asemenea abordată [84]. În acest paragraf se completează soluția dată la P5, prin proiectarea operatorului RNC pentru separarea părții statice a imaginilor în mișcare. Rezultatul poate fi folosit pentru analize complexe de imagini, care necesită succesiuni de matrici, aplicate într-un mod similar instrucțiunilor unui algoritm implementat pe un calculator clasic.

#### 4.2.1 Cadrul general al rețelelor neuronale celulare pentru mișcări continue

În cazul imaginilor aflate în mișcare, este necesară utilizarea matricilor de conexiune cu întârziere în timp. Fazele consecutive ale mișcării sunt generate de matricile de întârziere.

În modul de funcționare continuu, dinamica unei celule  $C_{i,j}$ , a unei RNC de dimensiune  $M \times N$  este descrisă de următoarea ecuație de stare:

$$C_{i,j} \mathbf{x}_{i,j}(t) = -\frac{1}{R_x} \mathbf{x}_{i,j}(t) + \sum_{C_{k,l} \in N_r(i,j)} \mathbf{A}_{i,j;k,l} \mathbf{y}_{k,l}(t) + \sum_{C_{k,l} \in N_r(i,j)} \mathbf{B}_{i,j;k,l} \mathbf{u}_{k,l}(t) + \sum_{C_{k,l} \in N_r(i,j)} \mathbf{A}^{\tau}_{(i,j;k,l)} \mathbf{y}_{k,l}(t - \tau) + \sum_{C_{k,l} \in N_r(i,j)} \mathbf{B}^{\tau}_{i,j;k,l} \mathbf{u}_{k,l}(t - \tau) + \mathbf{I} \quad (4.1)$$

unde:

- $\mathbf{A}^{\tau}$  matricea de interconexiune de reacție cu întârziere în timp;
- $\mathbf{B}^{\tau}$  este matricea de control cu întârziere în timp;
- celelalte mărimi au aceeași semnificație ca cea dată în § 2.2;

Ecuatiile de intrare și de ieșire, constrângerile impuse asupra stării celulelor, respectiv parametrilor rețelei sunt aceleași ca în cazul RNCA date în § 2.2.

Pixelii imaginii au valori între +1 pentru nivelul de negru și -1 pentru nivelul de alb.

Obiectul în mișcare este reprezentat de imaginea  $P=P(t)$  (de aceeași dimensiune cu cea a RNC), la care intensitatea pixelilor variază continuu în timp.  $P(t)$  se aplică la

intrarea RNC, astfel încât  $u_{ij} = P(t)$ . Starea inițială a rețelei este uzual identică cu imaginea de intrare la  $t=0$ ,  $x_{ij}(0) = P(0)$ . Fazele consecutive ale mișcării sunt generate de matricile de întârziere. RNC trebuie să fie mult mai rapidă decât mișcarea însăși, astfel încât timpul de întârziere trebuie ales în concordanță cu viteza mișcării. Relația dintre timpul de întârziere  $\tau$  și timpul de stabilizare a circuitului  $t_s$  este  $\tau \gg t_s$ .

Pentru a face diferența dintre două imagini consecutive  $P_0$  și  $P_1$ , cele două imagini se aplică independent, ca intrare a rețelei  $u_{ij} = P_0$ , respectiv ca stare  $x_{ij} = P_1$ . Operația de extragere a unei imagini din cealaltă se realizează conform tabelului de adevăr 4.1, unde  $P_d$  este  $y_{ij}$  ieșirea celulei după stabilizarea circuitului.

Pentru a implementa Tabelul 4.1 se poate utiliza următoarea succesiune de matrici de conexiune:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = -1 \quad (4.2)$$

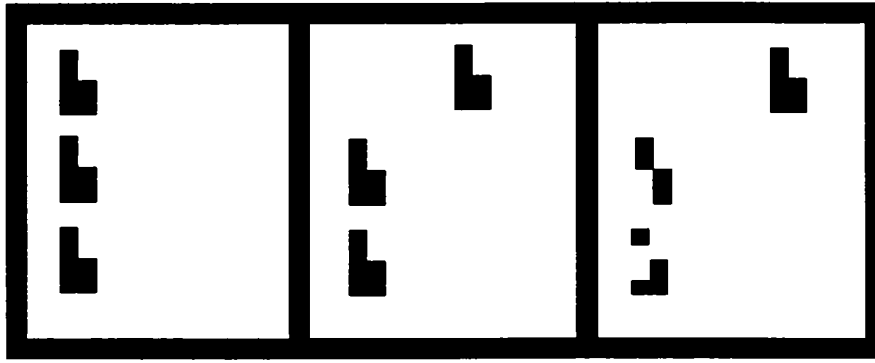
**Tabelul 4.1**  
**Tabelul de adevăr pentru diferența a două imagini**

$P_0 / u_{ij}$	$P_1 / x_{ij}$	$P_d / y_{ij} (\infty)$
-1	-1	-1
-1	1	1
1	-1	-1
1	1	-1

În Fig.4.1 este prezentat rezultatul diferenței a două imagini. În cazul imaginilor care variază continuu, nu putem utiliza stările inițiale ca intrări independente, și pentru diferența lor sunt necesare întârzieri [79]. Un template care extrage imaginea întârziată  $P(t-\tau)$  din cea curentă  $P(t)$  și furnizează continuu diferența în timp este următorul:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = -3 \quad (4.3)$$

$$A^i = [0], \quad B^i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



**Fig. 4.1 Diferența a două imagini succesive**

Într-o situație reală, imaginea de intrare are reflexii de lumină și alte perturbații. Este necesar un template mai complex, pentru a face o diferență și o filtrare a zgomotului.

Matricile pe care le căutăm trebuie să fie independente de direcție, deci simetrice față de centru:

$$\begin{aligned}
 \mathbf{A} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} a & a & a \\ a & b & a \\ a & a & a \end{bmatrix}, \quad \mathbf{I}, \quad \mathbf{A}^t = [0] \\
 \mathbf{B}^t &= \begin{bmatrix} c & c & c \\ c & d & c \\ c & c & c \end{bmatrix}, \quad a, b, c, d, \mathbf{I} \text{ constante}
 \end{aligned}
 \tag{4.4}$$

Aceste matrici vor genera RNC stabile întotdeauna, fiind simetrice. În plus se poate demonstra că tranzițiile sunt monotone în timp (fiecare variabilă este strict descrescătoare sau strict crescătoare în intervalul  $0 < t < t_s$  (vezi Anexa 2).

#### **4.2.2 Metoda de determinare a matricilor de conexiune pentru separarea părții statice a imaginilor ce conțin obiecte în mișcare**

În problemele de detecție apare de multe ori necesitatea separării părților statice de cele mobile ale imaginii în mișcare, în vederea unor prelucrări ulterioare.

Se prezintă o metodă de determinare a matricilor de conexiune pentru construirea unui operator ce implementează un tabel de adevăr oarecare, particularizată pentru separarea părții statice a imaginilor (Tabelul 4.2).

Prima coloană este pentru imaginea întârziată  $P(t-\tau)$ , coloana doi pentru imaginea curentă  $P(t)$ , iar coloana trei pentru rezultatul tranziției  $P_J$ .

**Tabelul 4.2**  
**Tabelul de adevăr pentru separarea părților statice a imaginilor în mișcare**

$P(t-\tau) / u_{ij}(t-\tau)$	$P(t) / u_{ij}(t)$	$Pd / y_{ij}(\infty)$
-1	-1	-1
1	1	1
-1	1	-1
1	-1	-1

În conformitate cu primul rând, dacă un pixel este alb atât în starea curentă cât și în cea întârziată, indiferent de vecini, celula  $C_{ij}$  va genera la ieșire tot nivelul de alb. Ținând cont de scrierea ecuației (4.1) sub formă matriceală și de condiția de monotonie a tranziției, vom obține inegalitatea:

$$\dot{x}_{ij} = -x_{ij}(t) + y_{ij}(t) + I - b \pm 8a - d \pm 8c < 0 \quad (4.5)$$

Semnul  $\pm$  arată că inegalitatea trebuie să aibă loc indiferent de starea vecinilor.

Similar, pentru rândurile doi, trei, patru din Tabelul 4.2 sunt valabile inegalitățile:

$$\dot{x}_{ij} = -x_{ij}(t) + y_{ij}(t) + I + b \pm 8a + d \pm 8c > 0 \quad (4.6)$$

$$\dot{x}_{ij} = -x_{ij}(t) + y_{ij}(t) + I + b \pm 8a - d \pm 8c < 0 \quad (4.7)$$

$$\dot{x}_{ij} = -x_{ij}(t) + y_{ij}(t) + I - b \pm 8a + d \pm 8c < 0 \quad (4.8)$$

Orice tabel de adevăr se poate transpune sub forma unui sistem de inegalități, în conformitate cu această metodă. Dacă inegalitățile sunt contradictorii, nu există soluție. Se pot obține condiții suplimentare luând în considerare starea admisă a pixelilor dintr-o vecinătate pentru o situație concretă. Dacă inegalitățile sunt redundante, evident se pot neglija cele inutile. Dacă se consideră inegalitățile (4.5) - (4.8), la momentul  $t=0$ , vom avea  $x_{ij}(0) = y_{ij}(0)$ . Dacă inegalitățile sunt satisfăcute la  $t=0$ , ele vor fi adevărate și la  $0 < t < t_c$ , în concordanță cu monotonia tranziției. Astfel se obțin următoarele inegalități:

$$I - b \pm 8a - d \pm 8c < 0 \quad (4.9)$$

$$I + b \pm 8a + d \pm 8c > 0 \quad (4.10)$$

$$I + b \pm 8a - d \pm 8c < 0 \quad (4.11)$$

$$I - b \pm 8a \mp d \pm 8c < 0 \quad (4.12)$$

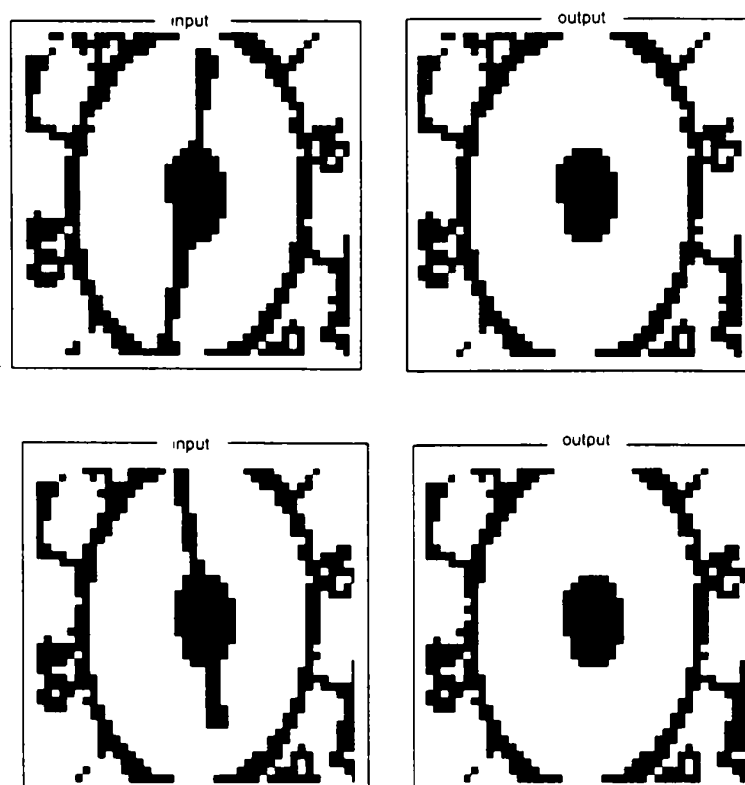
Aceste inegalități determină un poliedru convex în spațiul cu cinci dimensiuni al necunoscutelor. Cei cinci parametri necunoscuți  $a$ ,  $b$ ,  $c$ ,  $d$  și  $I$  vor fi localizați în interiorul poliedrului. Alegând  $a=c$ , o soluție adecvată poate fi S1:  $a=0$ ,  $b=3$ ,  $c=0$ ,  $d=3$ ,  $I=-3$ .

Într-o situație reală imaginea extrasă are reflexii de lumină și alte perturbații. Sunt necesare matrici de conexiune mai complexe pentru a selecta partea statică și pentru a filtra zgomotul. Se propune soluția S2 care realizează în plus o mediere a zgomotului:  $a=0.125$ ,  $b=3.2$ ,  $c=-0.125$ ,  $d=3.2$ ,  $I=-3$ .

Pentru selectarea părții mobile a unei imagini în mișcare trebuie implementat Tabelul de adevăr nr 4.1. Una dintre soluțiile ce pot fi obținute în urma rezolvării sistemului de inegalități dezvoltat din tabelul de adevăr este:  $a=0.25$ ,  $b=2$ ,  $c=-0.25$ ,  $d=-2$ ,  $I=-4.75$ .

### 4.2.3 Rezultatele simulării

Fig.4.2 prezintă imaginea unui ceas în funcționare, în două momente de timp diferite. Imaginea evoluției ceasului, în timp, a fost luată cu o videocameră și transformată într-un fișier de date accesibil simulatorului soft, și aplicată acestuia ca intrare. Arătătorul se mișcă, în timp ce restul ceasului este static. Rezoluția imaginii este de  $44 \times 44$ . S-a ales soluția S2 care asigură separarea părții statice și eliminarea zgomotului datorat reflexiilor de lumină și a micilor perturbații în sistemul de preprocesare.



**Fig. 4.2** Selectarea părții statice a unui ceas, în mișcare, în două momente de timp diferite

Fig.4.3 prezintă rezultatele procesării imaginii în mișcare de către operatorul de extragere a părții mobile.

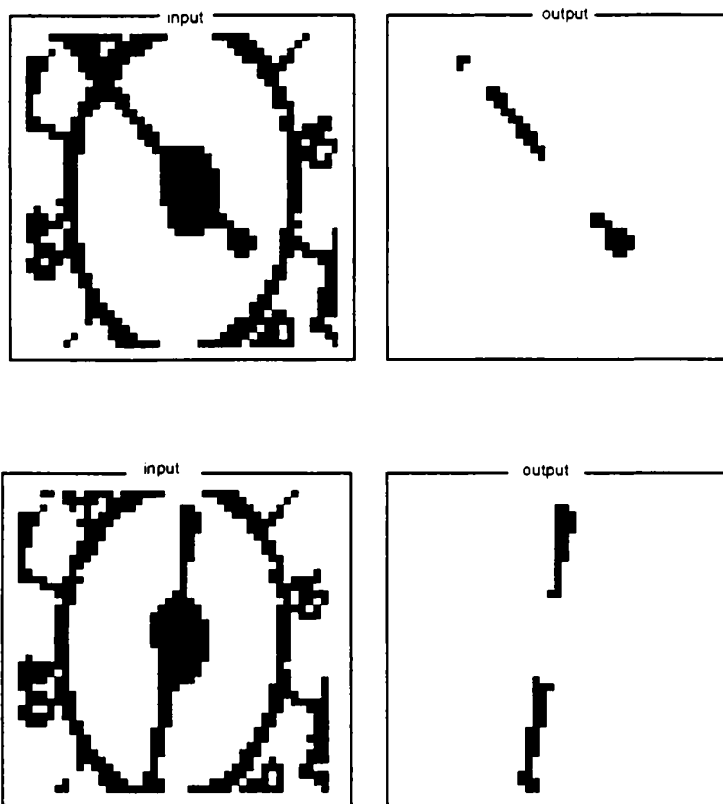


Fig. 4.3 Selectarea părții mobile a imaginii unui ceas, în două momente de timp diferite

### Observații

- Aplicând imaginea diferență la intrarea unor straturi succesive se poate continua analiza mișcării.
- O variantă a acestei metode de proiectare a fost utilizată cu succes pentru detecția mișcării unui obiect în mișcare continuă independentă de direcție, atunci când viteza sa se încadrează într-un domeniu dat (de ex. 1 pixel / perioadă de eșantionare). Matricile de interconexiune obținute sunt:

$$\begin{aligned}
 \mathbf{A} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \mathbf{B} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \mathbf{I} &= -2 \\
 \mathbf{A}^{\tau} &= \begin{bmatrix} 0.68 & 0.68 & 0.68 \\ 0.68 & 0.68 & 0.68 \\ 0.68 & 0.68 & 0.68 \end{bmatrix}, & \mathbf{B}^{\tau} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 0 \end{bmatrix}
 \end{aligned} \tag{6.13}$$



### 4.3 Reconstrucția obiectelor tridimensionale prin interpolare și aproximare

O suprafață descrisă prin câteva puncte (specificate dispersat) poate fi reconstruită prin metoda interpolării sau prin cea a aproximării folosind calculatoarele digitale, dar cu o viteză de calcul redusă datorată procesării secvențiale.

Utilizând rețele neuronale celulare, calculele vor avea loc în paralel cu o viteză semnificativ mai mare.

În cazul interpolării suprafața determinată se potrivește exact în punctele specificate, în timp ce în cazul aproximării suprafața nu este reconstruită exact prin punctele date, ci aproximativ.

#### 4.3.1 Modelul pentru descrierea unei suprafețe

Există o infinitate de suprafețe care pot fi definite prin punctele date, dar dintre acestea doar una singură este cea corectă. Pentru a obține suprafața dorită, se consideră un model fizic, și anume cel care descrie comportarea unei plăci subțiri supusă unor constrângeri speciale.

În cazul interpolării, constrângerile sunt date prin fixarea suprafeței în punctele specificate. În cazul aproximării constrângerile sunt date prin salturi față de punctele specificate.

Utilizând rețeaua neuronală celulară se simulează potențialul de energie al unei plăci rotunde subțiri. La echilibru, potențialul de energie al unui sistem mecanic este minim. Suprafața plăcii subțiri se găsește deci în minimul potențialului de energie și corespunde constrângerilor date.

Descriind suprafața cu o funcție  $z=v(x,y)$  următoarea relație definește potențialul de energie al plăcii :

$$E_{\text{placa}} = \iint_{\Omega} \left(\frac{1}{2}\right) (\Delta v)^2 - (1 - \sigma) (v_{xx}v_{yy} - v_{xy}v_{xy}) dx dy - \iint_{\Omega} g v dx dy - \int_{\partial\Omega} p(s) v ds - \int_{\partial\Omega} m(s) v_n ds + \frac{1}{2} w \sum_i [v(x_i, y_i) - z_i]^2 \quad (4.14)$$

Funcția de energie are cinci termeni:

- Primul termen este energia internă a plăcii. Factorul  $\sigma$  reprezintă *constantă Poisson*. El este folosit când au loc modificări în dimensiune, ca de exemplu întinderea longitudinală a plăcii;
- Cel de al doilea termen este dat de acțiunea gravitației;
- Următorii doi termeni reprezintă energia distribuită de-a lungul marginilor plăcii unde:
  - $p(s)$  reprezintă forțele externe aplicate pe perimetrul plăcii;
  - $m(s)$  este momentul de curbură al perimetrului plăcii;
  - $\Omega$  reprezintă suprafața plăcii;
  - $\delta\Omega$  este perimetrul plăcii;

- $s$  este lungimea arcului de-a lungul perimetrului;
- $v_n$  reprezintă derivata direcțională a vectorului  $v$ , normală la perimetrul plăcii;
- Ultimul termen reprezintă energia salturilor, unde  $w$  este saltul și  $x_i, y_i, z_i$  sunt cele trei coordonate ale punctului dat.

În cazul aproximării, nu sunt aplicate forțe externe și nu sunt luate în considerare efectele gravitației. Relația (4.14) devine:

$$E_{\text{aproximare}} = \iint (v_{xx}^2 + 2v_{xy}^2 + v_{yy}^2) dx dy + w \sum_i [v(x_i, y_i) - z_i]^2 \quad (4.15)$$

În cazul interpolării, salturile nu sunt utilizate și astfel relația se simplifică:

$$E_{\text{aproximare}} = \iint (v_{xx}^2 + 2v_{xy}^2 + v_{yy}^2) dx dy \quad (4.16)$$

### 4.3.2 Discretizarea modelului

Fie suprafața descrisă de funcția  $z=v(x,y)$ . Pentru reconstrucția suprafeței cu o rețea neuronală celulară este necesară discretizarea acesteia. În locul funcției  $z=v(x,y)$  se utilizează o grilă bidimensională de valori  $v_{(i,j)}$ , unde  $v_{(i,j)}$  reprezintă altitudinea punctului de coordonate  $(i,j)$ . Dacă dimensiunea grilei este de  $m \times m$ , se specifică  $m \times m$  valori  $v_{(i,j)}$ . Acestea pot fi date sub forma unui vector  $v$ .

Pentru discretizarea funcției de energie, termenii diferențiali se aproximează prin diferențe de termeni:

$$\begin{aligned} v_{xx}^2(i, j) &= \frac{1}{h^2} [v_{(i+1,j)} - 2v_{(i,j)} + v_{(i-1,j)}]^2 + O(h^2) \\ v_{yy}^2(i, j) &= \frac{1}{h^2} [v_{(i,j+1)} - 2v_{(i,j)} + v_{(i,j-1)}]^2 + O(h^2) \\ v_{xy}^2(i, j) &= \frac{1}{4h^2} [v_{(i-1,j-1)} - v_{(i+1,j-1)} - v_{(i-1,j+1)} + v_{(i+1,j+1)}]^2 + O(h^2) \end{aligned} \quad (4.17)$$

Integralele se aproximează cu sume. Funcția de energie devine :

$$\begin{aligned} E(\underline{v}) &= \sum_{i=1}^{m-2} \sum_{j=0}^{m-1} (v_{(i+1,j)} - 2v_{(i,j)} + v_{(i-1,j)})^2 + \sum_{i=0}^{m-1} \sum_{j=1}^{m-2} (v_{(i,j+1)} - 2v_{(i,j)} + v_{(i,j-1)})^2 + \\ &+ 2 \sum_{i=0}^{m-2} \sum_{j=0}^{m-2} (v_{(i,j)} - v_{(i+1,j)} - v_{(i,j+1)} + v_{(i+1,j+1)})^2 + w \sum_i (v_{(x_i, y_i)} - z_i)^2 \end{aligned} \quad (4.18)$$

Relația dată este utilă în cazul aproximării. În cazul interpolării ultimul termen al relației (4.18) lipsește .

### 4.3.3. Găsirea minimumului prin utilizarea unei metode de tip gradient. Interpolarea

Pentru determinarea vectorului  $\underline{v}$  (care reprezintă suprafața căutată) trebuie minimizată funcția sa de energie  $E(\underline{v})$ . Se utilizează o metodă de tip gradient folosind următoarea regulă de iterație:

$$\underline{v}(k+1) = \underline{v}(k) - \beta \text{grad}E(\underline{v}) \Big|_{\underline{v}_k}, \quad \lim_{k \rightarrow \infty} \underline{v}(k) = \underline{v}_{\min} \quad (4.19)$$

unde  $\beta$  este o constantă.

Pentru o abordare prin RNC se folosește o formulă modificată. Starea staționară a următoarei ecuații permite determinarea vectorului corespunzător energiei minime:

$$\frac{d\underline{v}}{dt} = -\beta \cdot \text{grad}E(\underline{v}) \quad (4.20)$$

Funcția de energie poate fi exprimată ca suma produselor dintre două elemente ale grilei și o constantă:

$$E(\underline{v}) = \frac{1}{2} \sum_{(i,j)} \sum_{(k,l) \in N_r(i,j)} C(ij,kl) v_{(i,j)} v_{(k,l)}, \quad r = 2 \quad (4.21)$$

O coordonată a gradientului este dată de suma produselor dintre elementele grilei și o constantă. Vectorul  $\underline{v}$  are  $m \times m$  coordonate, deci și gradientul are  $m \times m$  valori corespunzătoare.

$$\frac{\partial E}{\partial v_{(i,j)}} = \sum_{(k,l) \in N_r(i,j)} C(ij,kl) v_{(k,l)}, \quad r = 2 \quad (4.22)$$

Derivând relația (4.23) în raport cu  $v_{(k,l)}$  se obține:

$$C(ij,kl) = \frac{\partial^2 E}{\partial v_{(i,j)} \partial v_{(k,l)}} \quad (4.23)$$

Comparând ecuația obținută prin metoda gradientului (4.23) cu formula ecuației diferențiale a unei celule (2.5), (starea unei celule fiind corespunzătoare funcției de energie  $E(\underline{v})$ ), se pot obține valorile interconexiunilor de reacție  $A_{ij,kl}$ :

$$A_{ij,kl} = -C(ij,kl) + 1 \quad (4.24)$$

Rețeaua neuronală celulară de interpolare are următoarele caracteristici:

- Curentul de polarizare  $I$  este zero;
- Matricea de control  $B$  este zero;
- Funcția neliniară aplicată stării unei celule este  $f(x) = x$ ;
- Matricea de reacție  $A$  este variantă în spațiu și elementele ei se determină cu relația (4.24);

- Punctele specificate se mențin ca stări fixe ale celulelor RNC corespunzătoare;

#### 4.3.4 Aproximarea

Prin interpolare se reconstruiește suprafața exact prin punctele date. Dacă printre punctele date un singur punct este greșit, suprafața obținută este deformată față de cea dorită. În acest caz, suprafața va corespunde aproximativ punctelor date.

Se dorește ca suprafața să fie netedă și să se potrivească bine în punctele date. Un factor de pondere  $w$  va determina importanța relativă a celor două cerințe. Dacă  $w$  este mic, suprafața se va potrivi aproape exact prin punctele date, deci procesarea va avea un rezultat similar interpolării. Dacă  $w$  este mare netezimea suprafeței este mai importantă, potrivirea suprafeței nu este exactă prin punctele date.

R.N.C pentru aproximare prezintă următoarele diferențe față de cea de interpolare:

- Pentru celula  $C_{ij}$  corespunzătoare unui punct dat, elementul central al matricii de reacție  $A$  este mai mic decât la rețeaua de interpolare:

$$a_{\text{aprox (central)}} = a_{\text{interp (central)}} - 2 \cdot w$$

- Curentul de polarizare  $I$  este diferit de zero:  $I_{\text{aprox}} = 2w \cdot z_i$
- Nici o celulă nu are starea menținută fixă.

#### 4.3.5 Rezultatele simulării

Utilizând relațiile (4.22), (4.23) și (4.24) au fost determinate în “Mathcad 6 Plus” interconexiunile de reacție ale RNC de rază  $r=2$  care realizează reconstrucția unei suprafețe prin interpolare, pentru diferite dimensiuni ale suprafeței simulate.

Rezultatele procesării sunt sintetizate în Tabelul 4.3, care prezintă matricea de interconexiune a celulei de coordonate  $i,j$ . Literele  $a, b, c, d, e, f$  desemnează tipurile de matrici de bază, prezentate în cele ce urmează. Litera  $x$  atașată unei matrici semnifică reflexia acesteia după axa  $x$ , iar litera  $y$  reflexia după axa  $y$ . Litera  $t$  are semnificația de transpunere a matricii la care este atașată.

Tabelul 4.3

$i/j$	0	1	2	3	...	m-3	m-2	
0	f	d	c	c	.	b	ey	dtx
1	dt	e	b	b	.	b	ey	dt <sub>y</sub>
2	ct	bt	a	a	.	a	b <sub>t</sub> y	ct <sub>y</sub>
3	ct	bt	a	a	.	a	b <sub>t</sub> y	ct <sub>y</sub>
m-3	ct					a	b <sub>t</sub> y	ct <sub>y</sub>
m-2	dt <sub>x</sub>	e <sub>x</sub>	b <sub>x</sub>	b <sub>x</sub>	.	b <sub>x</sub>	e <sub>x</sub> y	dt <sub>x</sub> y
m-1	f <sub>x</sub>	d <sub>x</sub>	c <sub>x</sub>	c <sub>x</sub>	.	c <sub>x</sub>	d <sub>x</sub> y	f <sub>x</sub> y

Matricile de bază determinate sunt următoarele:

$$\begin{aligned}
 \mathbf{a} &= \begin{bmatrix} 0 & 0 & -2 & 0 & 0 \\ 0 & -4 & 16 & -4 & 0 \\ -2 & 16 & -39 & 16 & 0 \\ 0 & -4 & 16 & -4 & 0 \\ 0 & 0 & -2 & 0 & 0 \end{bmatrix} ; & \mathbf{b} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -4 & 12 & -4 & 0 \\ -2 & 16 & -37 & 16 & -2 \\ 0 & -4 & 16 & -4 & 0 \\ 0 & 0 & -2 & 0 & 0 \end{bmatrix} \\
 \mathbf{c} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & 12 & -21 & 12 & -2 \\ 0 & -4 & 12 & -4 & 0 \\ 0 & 0 & -2 & 0 & 0 \end{bmatrix} ; & \mathbf{d} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & -19 & 12 & -2 \\ 0 & -4 & 12 & -4 & 0 \\ 0 & 0 & -2 & 0 & 0 \end{bmatrix} \\
 \mathbf{e} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -4 & 12 & -4 & 0 \\ 0 & 12 & -35 & 16 & -2 \\ 0 & -4 & 16 & -4 & 0 \\ 0 & 0 & -2 & 0 & 0 \end{bmatrix} ; & \mathbf{f} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -7 & 8 & -2 \\ 0 & 0 & 8 & -4 & 0 \\ 0 & 0 & -2 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

Fie de exemplu matricea de interconexiune corespunzătoare celulei din poziția (0,m-2), dtx. Ea este o matrice de tipul d, transpusă și apoi reflectată după axa x.

Transpusa matricii d este:

$$\mathbf{dt} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & -4 & 0 \\ 0 & 0 & -19 & 12 & -2 \\ 0 & 0 & 12 & -4 & 0 \\ 0 & 0 & -2 & 0 & 0 \end{bmatrix}$$

În urma reflexiei după axa x se obține:

$$\mathbf{dtx} = \begin{bmatrix} 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 12 & -4 & 0 \\ 0 & 0 & -19 & 12 & -2 \\ 0 & 0 & 8 & -4 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

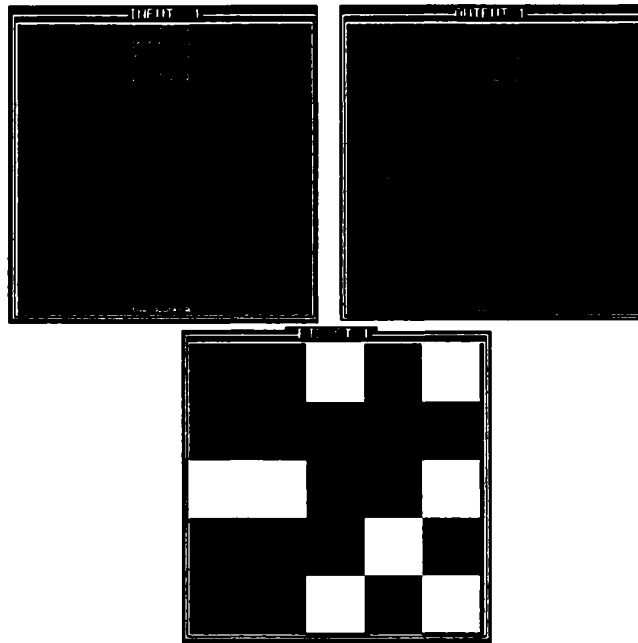
Respectând condițiile date anterior, în paragraful 4.3.3 s-a experimentat în simulatorul soft variant în spațiu reconstrucția unor suprafețe de diferite dimensiuni.

Punctele date ale suprafeței, în spațiul tridimensional sunt specificate prin trei coordonate i, j, z. Coordonatele i și j sunt coordonatele cuantizate pe axa x și axa y, coordonatele z reprezintă altitudinea suprafeței în punctul (i,j) al grilei.

# Interpolarea

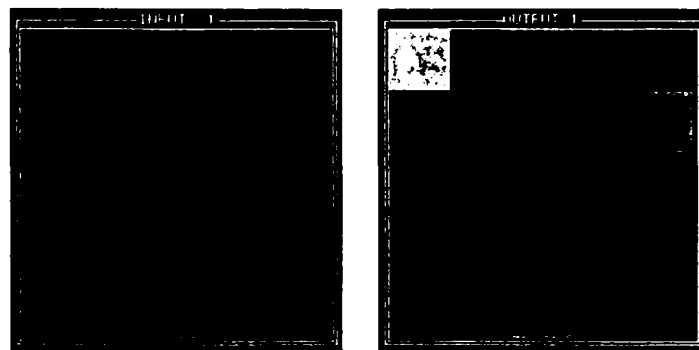
## Exemplul 1

Se dorește reconstrucția prin interpolare a unei suprafețe plane de dimensiune  $5 \times 5$ , specificându-se opt puncte, cu valori în intervalul  $(0,0.5)$ . Punctele care definesc suprafața sunt date prin starea inițială a celulelor corespunzătoare RNC. Aceste stări pentru interpolare, trebuie menținute fixe. Matricea de interconexiune pentru dimensiunea de  $5 \times 5$  este cea obținută din Tabelul 4.3.

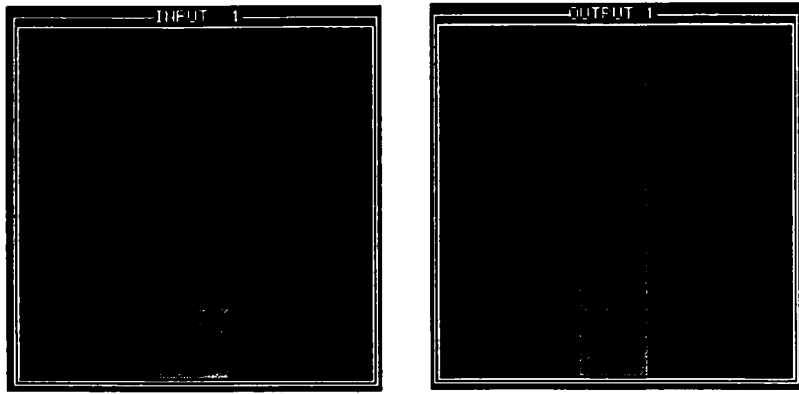


**Fig. 4.4** Intrarea, ieșirea și starea fixă a RNC în momentul inițial

Fig. 4.5 prezintă starea RNC de interpolare după  $100\tau$ , unde  $\tau$  este constanta de timp a sistemului. RNC se stabilizează după 900 de pași la starea ilustrată în Fig.4.6

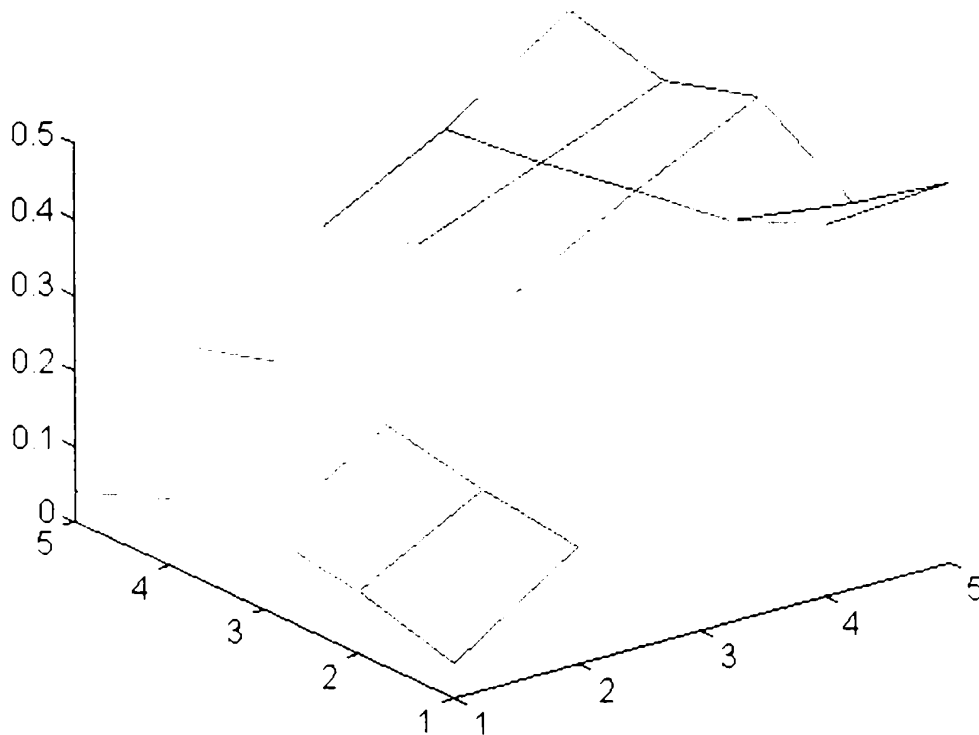


**Fig.4.5** Intrarea și ieșirea RNC după 100 de iterații



**Fig. 4.6** Intrarea și ieșirea RNC de interpolare după stabilizare

Folosind mediul de calcul “Matlab” pentru reprezentarea stărilor celulelor RNC după stabilizare, se poate vizualiza suprafața obținută (Fig.4.7).



**Fig. 4.7** Suprafața interpolată

Se observă că suprafața obținută nu este una netedă. Aceasta se datorează dimensiunii reduse a rețelei și intervalului de valori restrâns  $[-1, 1]$  în care operează simulatorul.

### Exemplul 2

Se reconstruiește o piramidă de dimensiune  $7 \times 7$ , prin metoda interpolării, prin 10 puncte cu valori în intervalul  $(0,0.8)$ . Fig 4.8 prezintă o stare intermediară a RNC după  $17\tau$ , iar Fig.4.9 starea finală stabilă după  $250\tau$ .

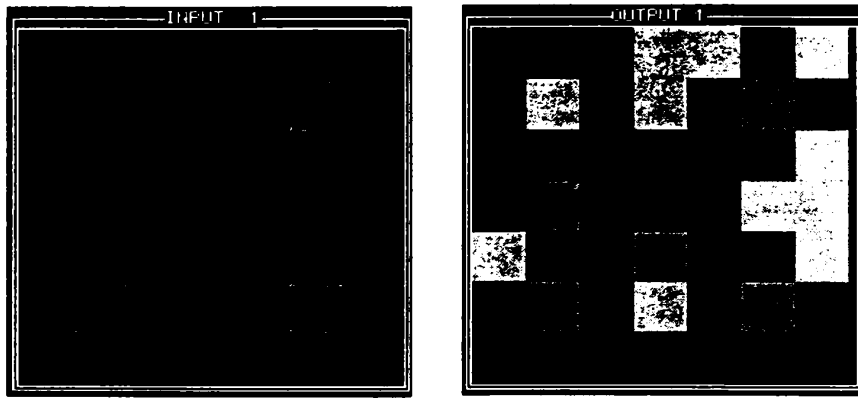


Fig. 4.8 Intrarea și ieșirea RNC după  $17\tau$

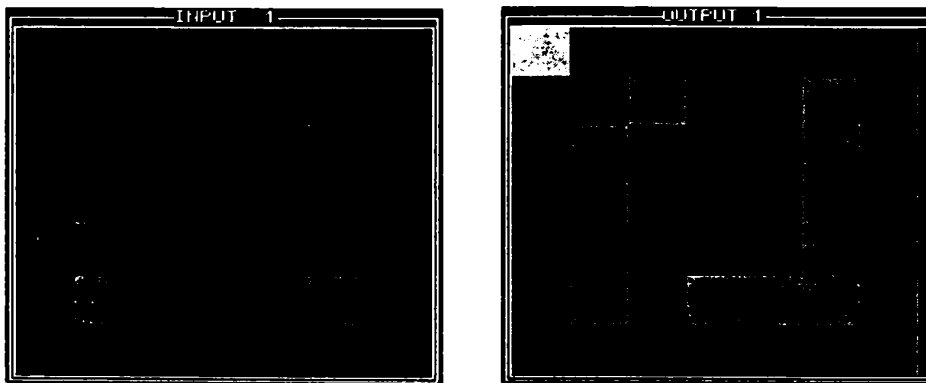


Fig. 4.9 Intrarea și ieșirea RNC după stabilizare

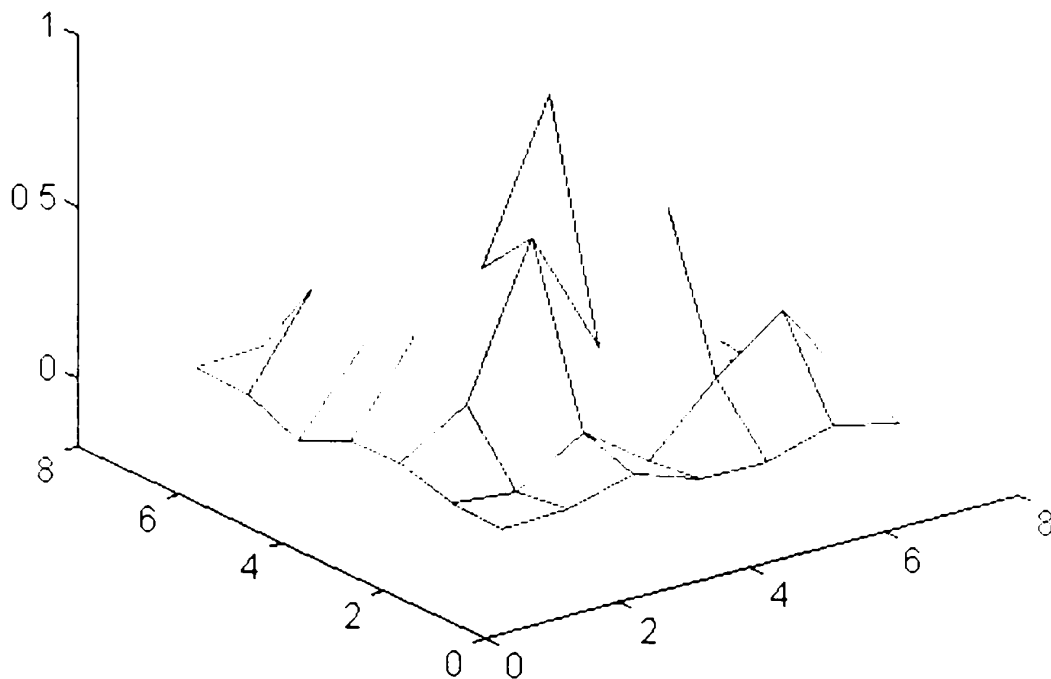
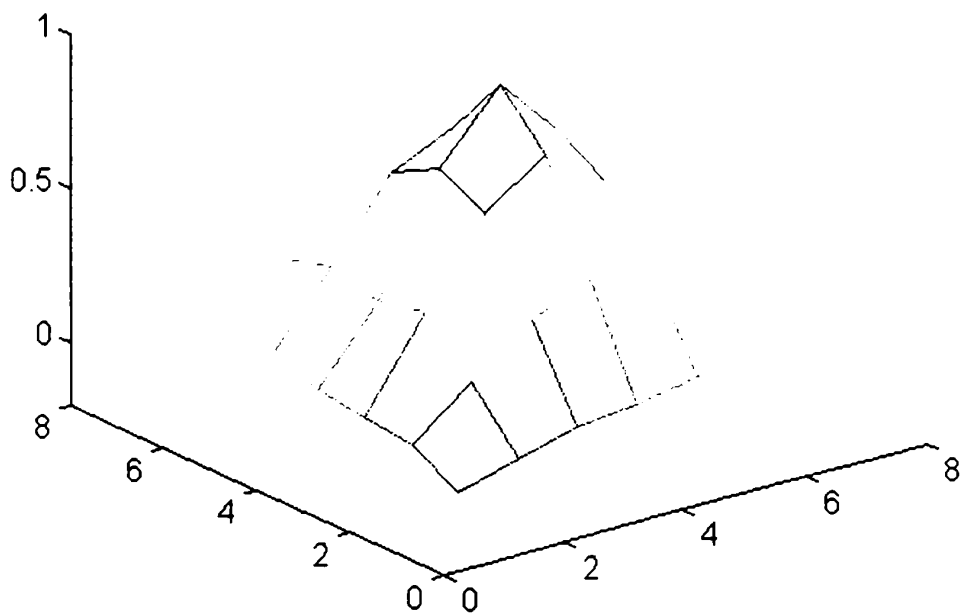


Fig. 4.10 Suprafața interpolată după  $17\tau$





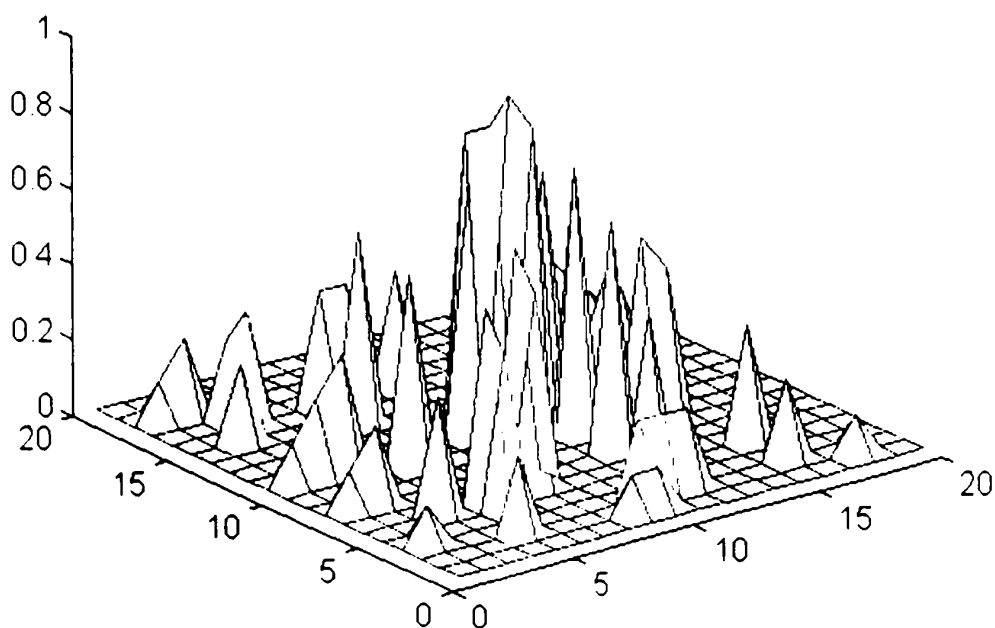
**Fig. 4.11 Suprafața interpolată după stabilizare**

Fig.4.10 ilustrează suprafața interpolată într-o fază intermediară a procesării (la  $17\tau$ ) și Fig.4.12 după stabilizare.

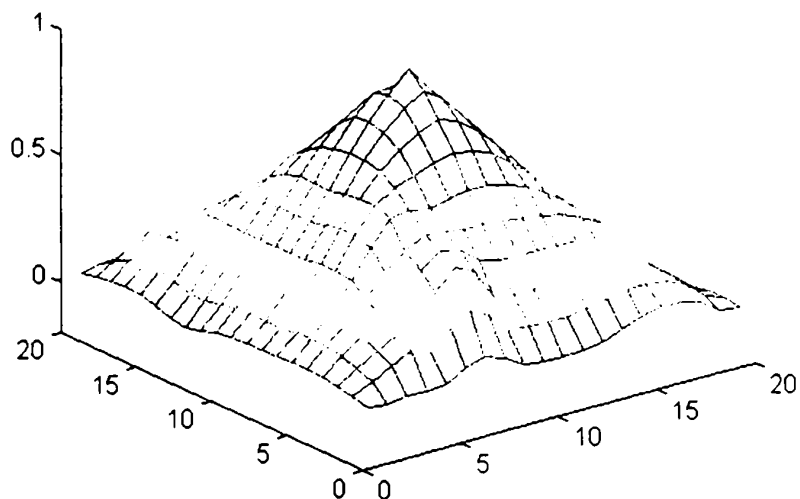
### Exemplul 3

Se reconstruiește o piramidă de dimensiune  $20 \times 20$ , precizându-se 60 de puncte (15%), cu valori în intervalul  $(0,0.8)$ . Fig 4.12 este reprezentarea tridimensională a stării inițiale RNC. Graficul tridimensional 4.13 reprezintă suprafața simulată după 2000 de iterații.

Se poate observa că precizia de reconstrucție este mult mai bună, suprafața obținută fiind mai netedă.



**Fig. 4.12 Reprezentarea tridimensională a stării inițiale a celulelor RNC pentru interpolarea unei piramide de dimensiune  $20 \times 20$**



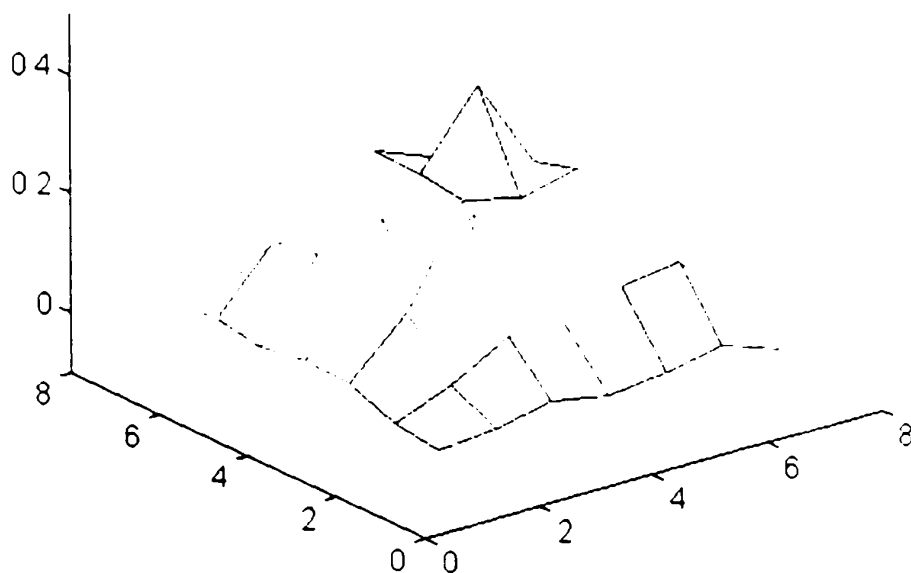
**Fig. 4.13 Suprafața interpolată**

## Aproximarea

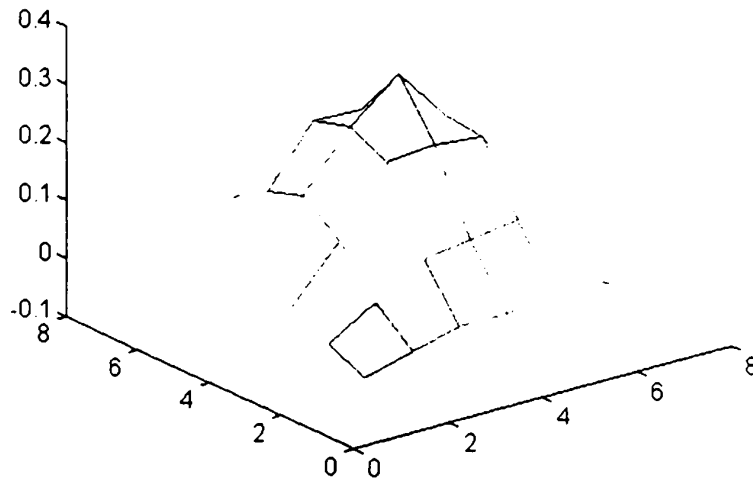
### Exemplul 4

Se dorește, ca pe baza punctelor de intrare date, să se reconstruiască prin metoda aproximării cu o RNC de dimensiune  $7 \times 7$  o suprafață piramidală, prin aceleași opt puncte ca în exemplul 2.

Matricile de interconexiune de reacție și de polarizare utilizate în acest caz diferă de cele de la interpolare în conformitate cu precizările din paragraful 4.3.3. Se introduc în RNC aceleași stări inițiale pentru celule, ca în exemplul 2 dar nici una nu este menținută fixă. Se alege parametrul  $w=5$ . Fig. 4.14 ilustrează suprafața piramidală construită.



**Fig. 4.14 Suprafața aproximată cu  $w=5$**

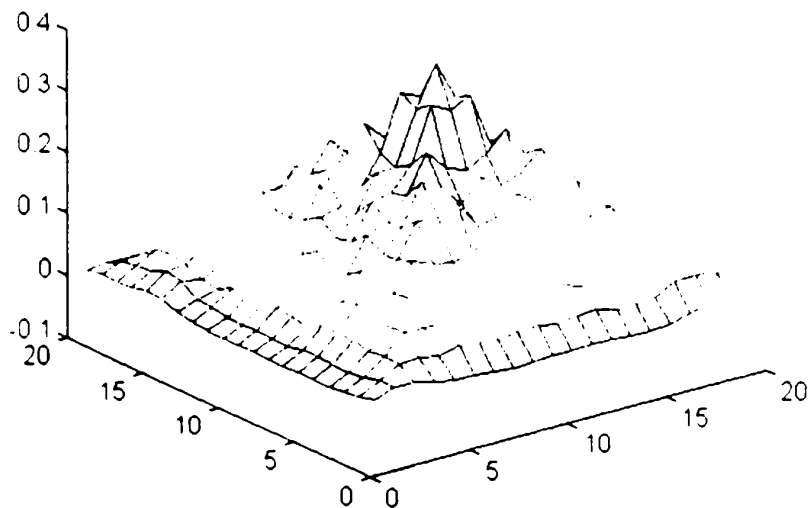


**Fig. 4.15 Suprafața aproximată cu  $w=2$**

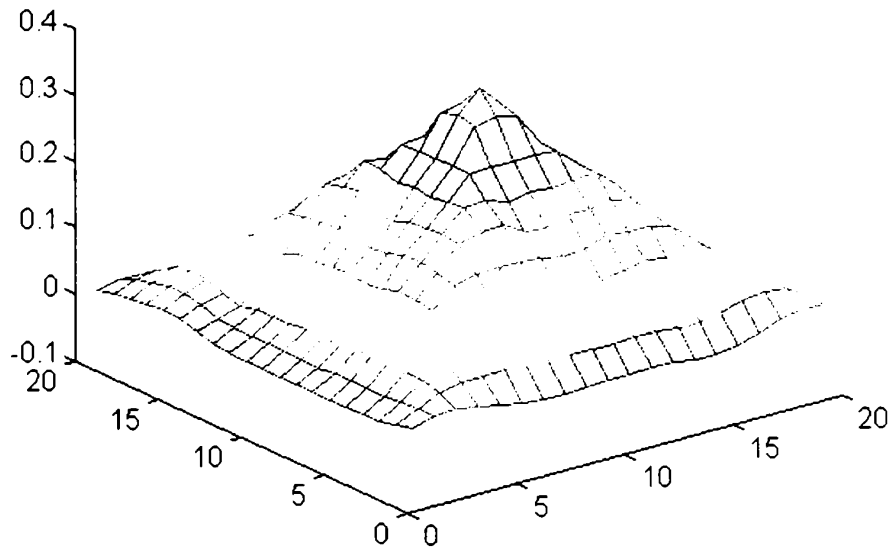
RNC s-a stabilizat după  $402\tau$ , deci după un interval de timp mai îndelungat decât în cazul interpolării. Suprafața obținută nu este la fel de netedă ca în cazul interpolării. Eroarea de reconstrucție este mult mai mare. Dacă se alege  $w$  mai mic, de exemplu  $w=2$ , eroarea de aproximare este mai mică după cum se vede în Fig. 4.15.

### Exemplul 5

Fig. 4.16 ilustrează reconstrucția prin aproximare a unei piramide de dimensiune  $20 \times 20$  dată prin starea inițială din exemplul 2 și  $w=2$ . În cazul în care  $w$  este mai mic,  $w=0.5$ , precizia aproximării este mult mai bună, deci suprafața obținută este mai netedă așa cum se poate observa în Fig. 4.17.



**Fig. 4.16 Suprafața aproximată cu  $w=2$**



**Fig. 4.17 Suprafața aproximată cu  $w=0,5$**

### Observații

- Reconstrucția prin interpolare și aproximare cu RNC are doi parametri importanți:
  - eroarea de reconstrucție
  - timpul de stabilizare
- Este evident faptul că eroarea de reconstrucție scade odată cu creșterea numărului de constrângeri inițiale (punctele date). Cu cât numărul punctelor specificate inițial crește, cu atât timpul de stabilizare al RNC crește și el. Într-o situație reală, timpul de procesare este esențial, astfel încât trebuie găsit un raport optim între precizia reconstrucției și viteza de procesare, de la caz la caz.
- RNC de dimensiuni mai mari au un potențial de procesare mai mare decât cele de dimensiuni mai reduse. Precizia de reconstrucție este mai bună în detrimentul timpului de stabilizare.
- În urma experimentelor efectuate am constatat că procesarea depinde și de densitatea și poziția punctelor specificate. Dacă densitatea punctelor pentru diferite zone variază, eroarea de reconstrucție este mai mare decât în cazul în care densitatea are valoarea medie (corespunzătoare mediei densității diferitelor zone).

## 4.4 Rotația obiectelor

În procesele de recunoaștere (unde pentru a fi recunoscut modelul este rotit) rotația este o operație uzuală, variantă în spațiu și simetrică.

Structura RNC folosită pentru această aplicație, prezentată în Fig.4.18, este constituită din două straturi: unul care conține informația variantă în spațiu codată, numit strat de control, și cel de-al doilea strat ce conține obiectul care trebuie rotit, cel de referință. În exemplul ilustrat în Fig.4.18 se utilizează o cale circulară în stratul de control.

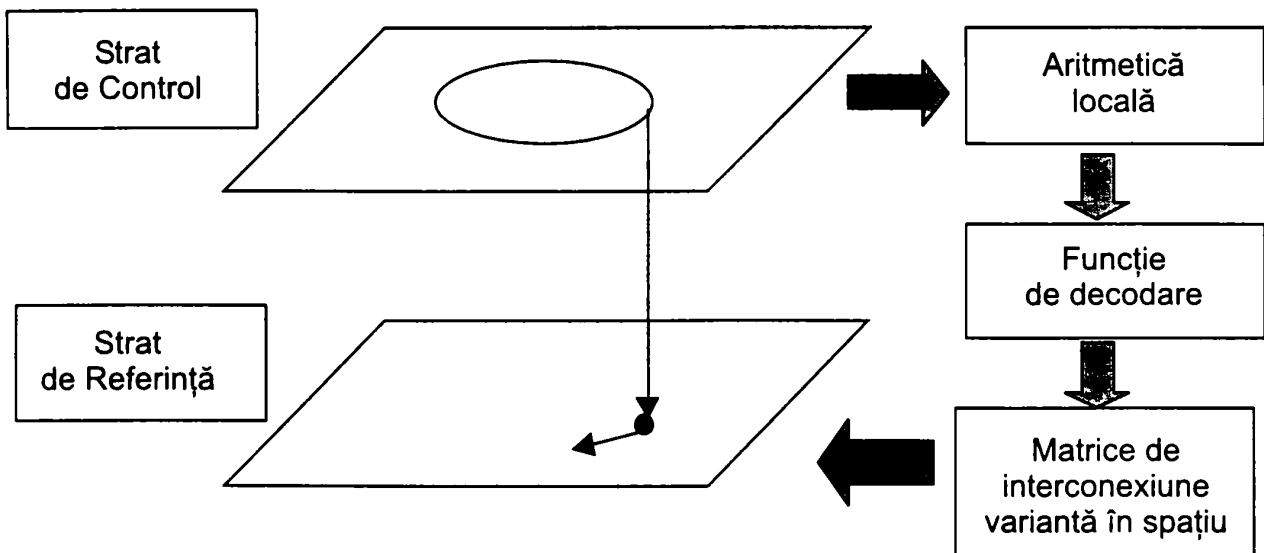


Fig. 4.18 Structura unei RNC care realizează rotația obiectelor

Argumentul funcției de decodare  $f$  este determinat printr-un calcul local, de exemplu prin diferența valorilor pixelilor vecini,  $y_{ij} - y_{kl}$ .

Ieșirea acestei funcții generează elementele matricii neliniare de interconexiune de reacție variante în spațiu, care este aplicată stratului de referință .

$$A_{ij,kl} = f(y_{ij,con} - y_{kl,con}) \quad (4.23)$$

Structura prezentată este descrisă de următoarea ecuație de stare:

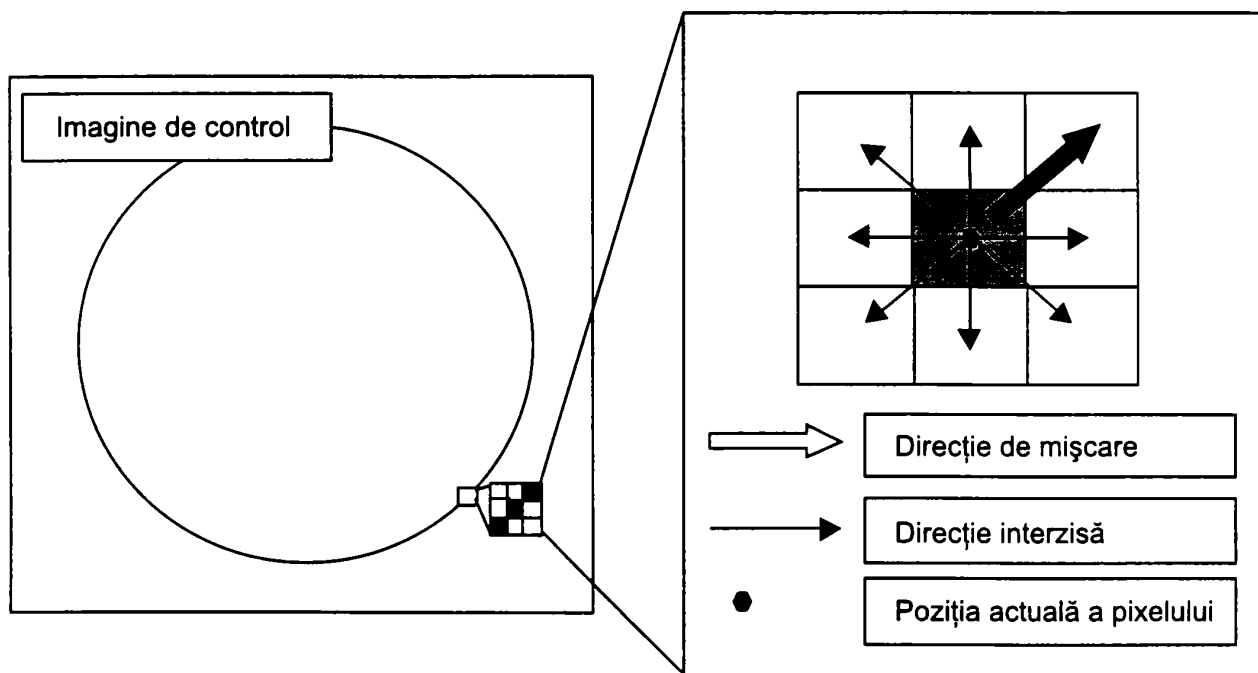
$$x_{ref}(k+1) = (1-h)x_{ref}(k) + h \left[ \sum_{C_{kl} \in N_r} A_{ij,kl} \cdot y_{ij,ref} + \sum_{C_{kl} \in N_r} B_{ij,kl} \cdot u_{ij,ref} \right] + I \quad (4.24)$$

unde  $h$  este pasul de iterație.

### 4.4.1 Deplasarea unui singur pixel negru pe o cale circulară

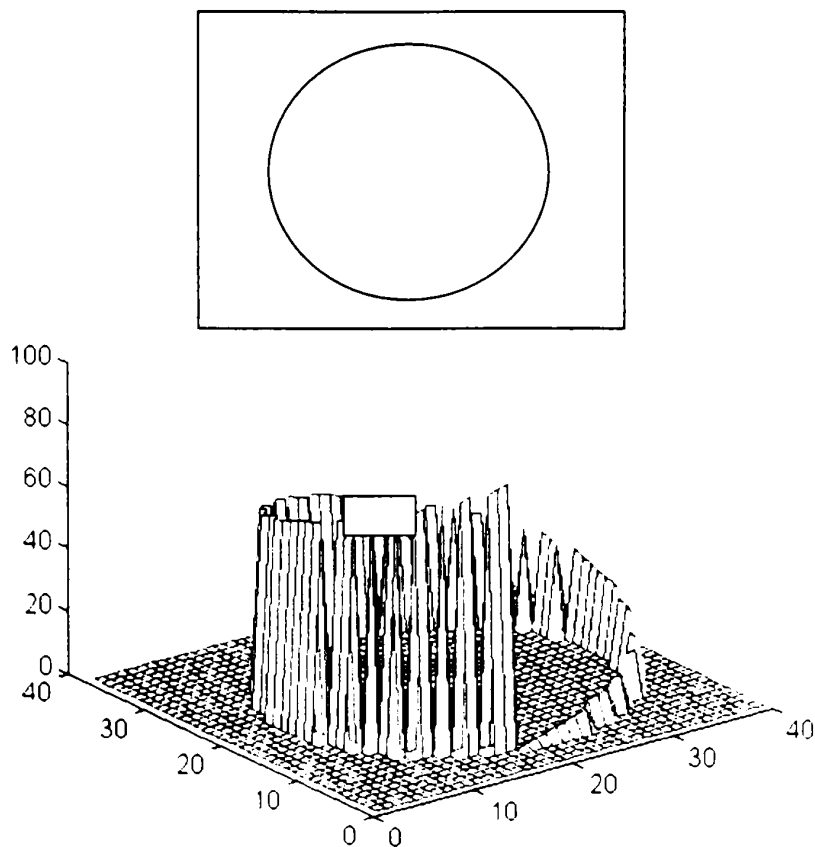
Fie un obiect format dintr-un singur pixel negru, de valoare  $+1$ , situat undeva pe imagine (necentral). Scopul este de deplasa pixelul pe o cale circulară în jurul centrului.

Direcția și viteza mișcării depinde de poziția pixelului față de centru. Aceasta informație poate fi codată într-o fereastră de 3x3 (Fig.4.19).



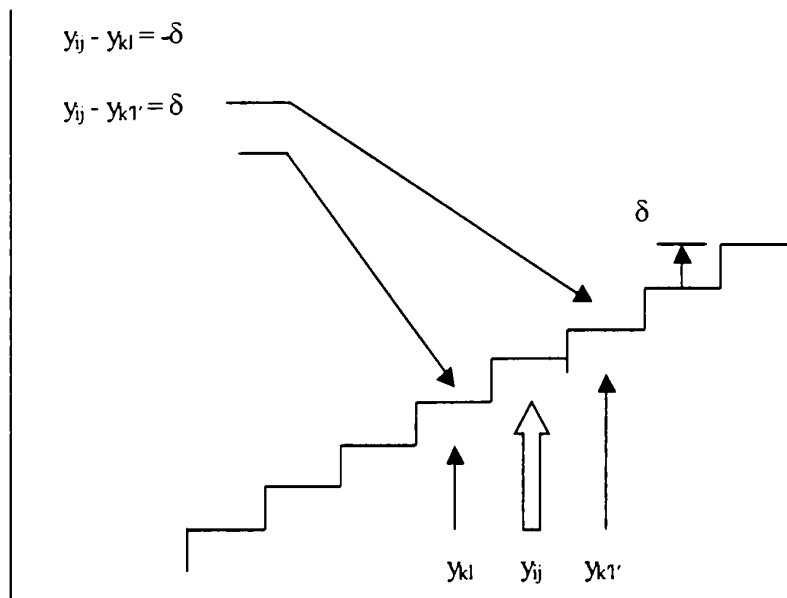
**Fig 4.19 Codarea locală a informației necesare rotației, în imaginea de control**

Codarea direcției necesare pentru deplasarea pixelului pe o cale circulară poate fi inclusă într-o funcție circulară în trepte, ilustrată în Fig. 4.20 .



**Fig.4.20 Imaginea de control și reprezentarea ei 3D**

Pasul acestei funcții are o valoare bine determinată  $\delta$ . În fereastra din figura 4.21 diferența dintre valoarea pixelului elementului central și valoarea pixelului adiacent este:  $\delta$  în direcția de mișcare,  $-\delta$  în direcția opusă, și mult mai mare în toate celelalte direcții (cele interzise).



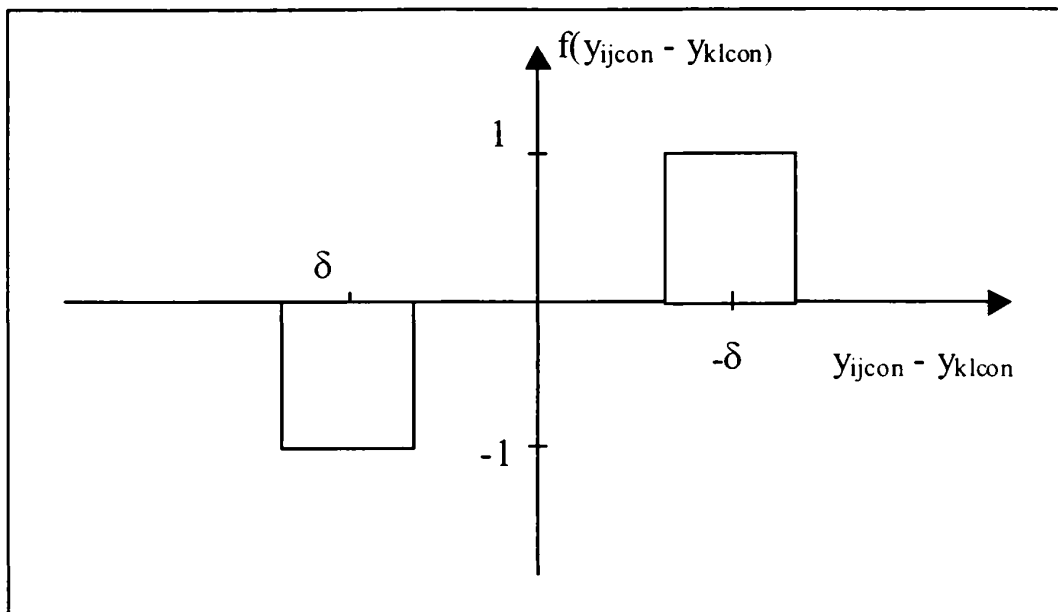
**Fig. 4.21 Detaliu corespunzător ferestrei din Fig.4.20.Reprezentarea stării celulelor adiacente din imaginea de control**

Informația de direcție poate fi obținută din imaginea de control într-un mod similar cu modul de procesare al binecunoscutului operator detector de componente conectate (DCC) (connected component detector). El conține doar trei valori diferite de zero: 2 în centru, -1 în direcția de mișcare și 1 în direcția opusă. DCC mută un punct pe o dreaptă. Dacă s-ar putea realoca valorile -1 și 1 în funcție de poziție, s-ar putea transforma DCC într-un operator variant în spațiu.

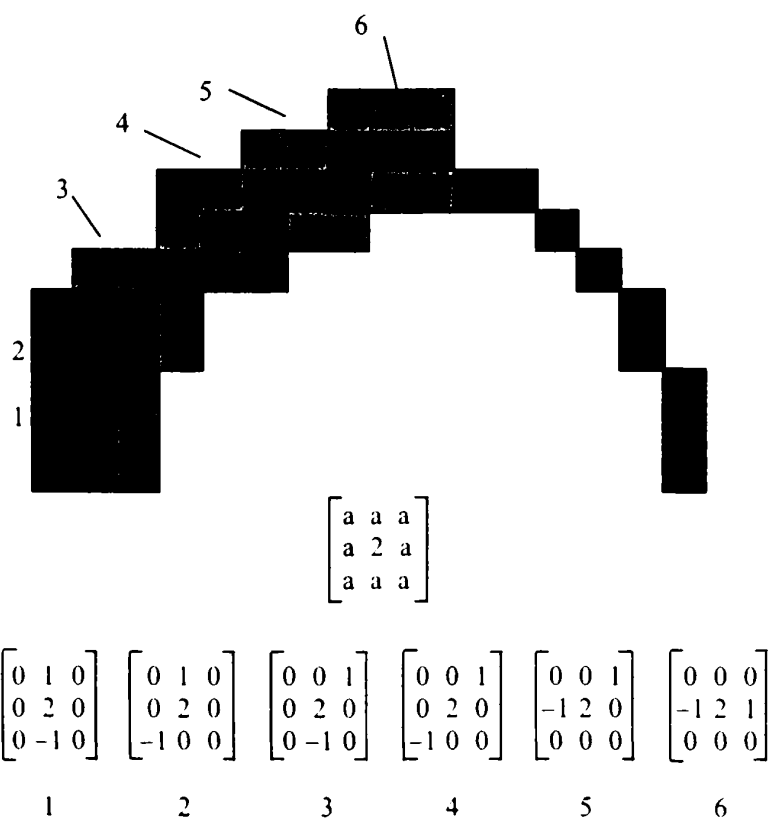
Diferențele posibile într-o vecinătate dată corespunzătoare ieșirilor funcției de decodare sunt:

$$A_{ij,kl} = f(v_{ij,con} - v_{kl,con}) = \begin{cases} 0, & \text{pentru } y_{ij,con} - y_{kl,con} = 0 \\ 1, & \text{pentru } y_{ij,con} - y_{kl,con} = \delta \\ -1, & \text{pentru } y_{ij,con} - y_{kl,con} = -\delta \\ 0, & \text{pentru } y_{ij,con} - y_{kl,con} \gg \delta \end{cases} \quad (4.25)$$

Fig. 4.23 prezintă funcția neliniară de control capabilă să alocă valori adecvate elementelor matricii de reacție, în concordanță cu diferențele dintre valorile pixelilor din imaginea de control. Fig.4.24 ilustrează modul în care funcția de decodare alocă valori potrivite într-o fereastră de dimensiune 3x3.



**Fig. 4.23 Funcția neliniară de control**



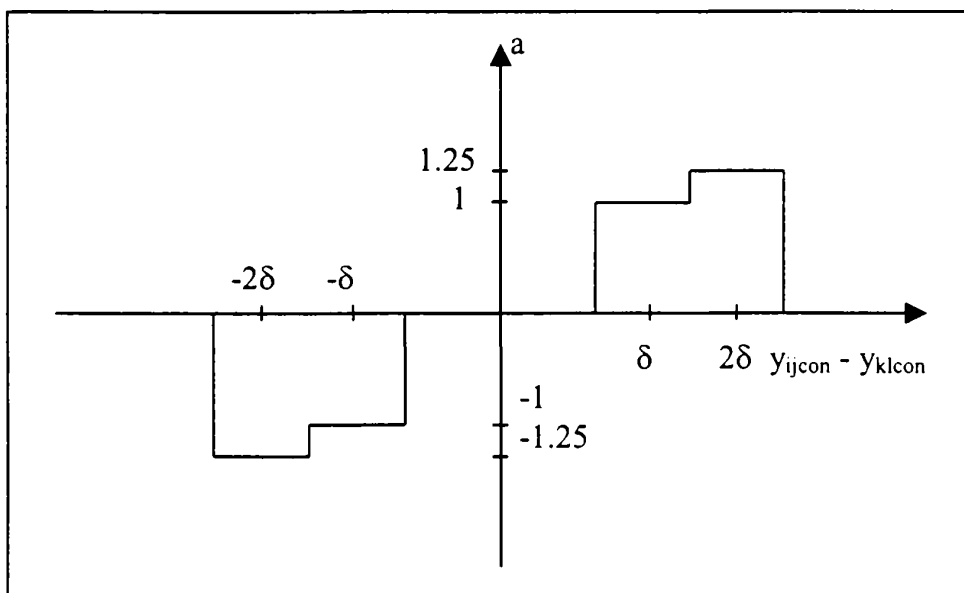
**Fig. 4.24**

#### 4.4.2 Rotirea a doi pixeli

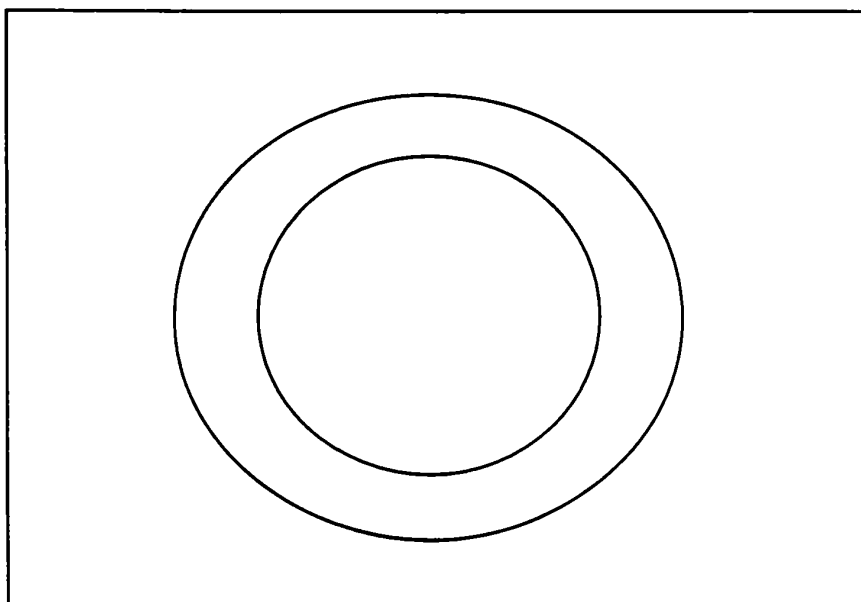
Dacă se exprimă poziția pixelului în coordonate polare cu centrul de rotație ca origine, direcția de deplasare depinde de o variabilă unghiulară, în timp ce viteza pixelului depinde de distanța acestuia față de centru. Această ultimă proprietate poate fi exprimată



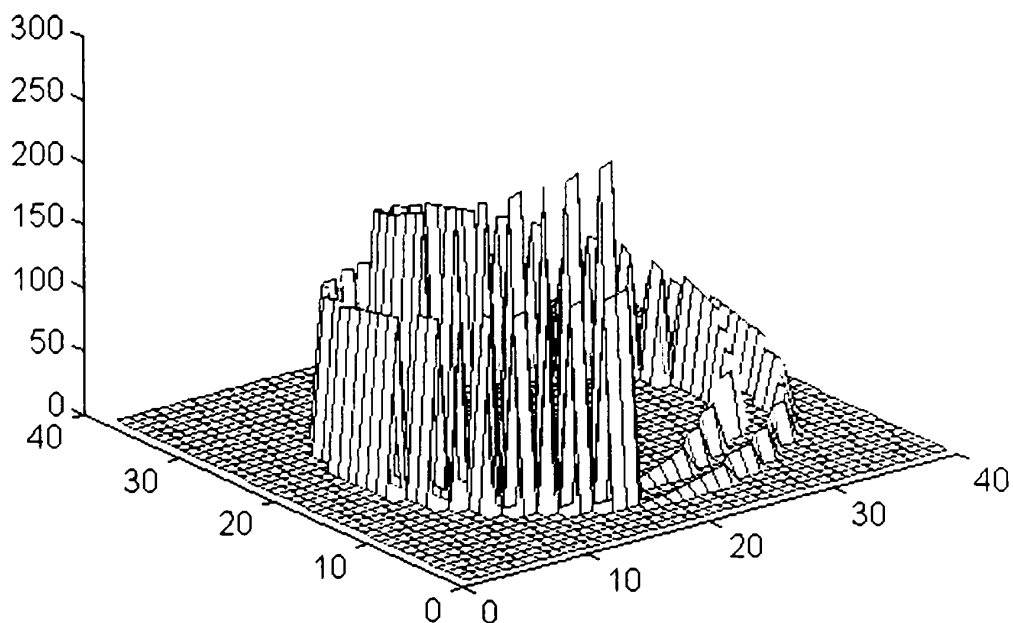
printr-o funcție neliniară cu trepte multiple, în care viteza este codată prin mărimea treptelor (cuantelor). Fig.4.25 prezintă funcția de decodare în cazul a două puncte cu distanțe radiale diferite. Imaginea de control utilizabilă este cea din Fig. 4.26.



**Fig.4.25 Funcția neliniară care codează distanța radială pentru rotirea a doi pixeli aflați la distanțe diferite de centru, cu aceeași viteză unghiulară**



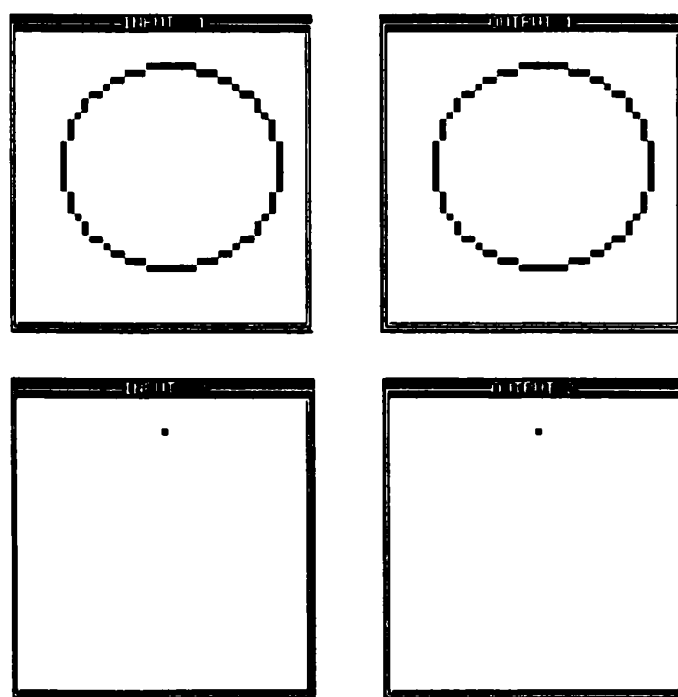
**Fig. 4.26 Imaginea de control pentru rotirea a doi pixeli**



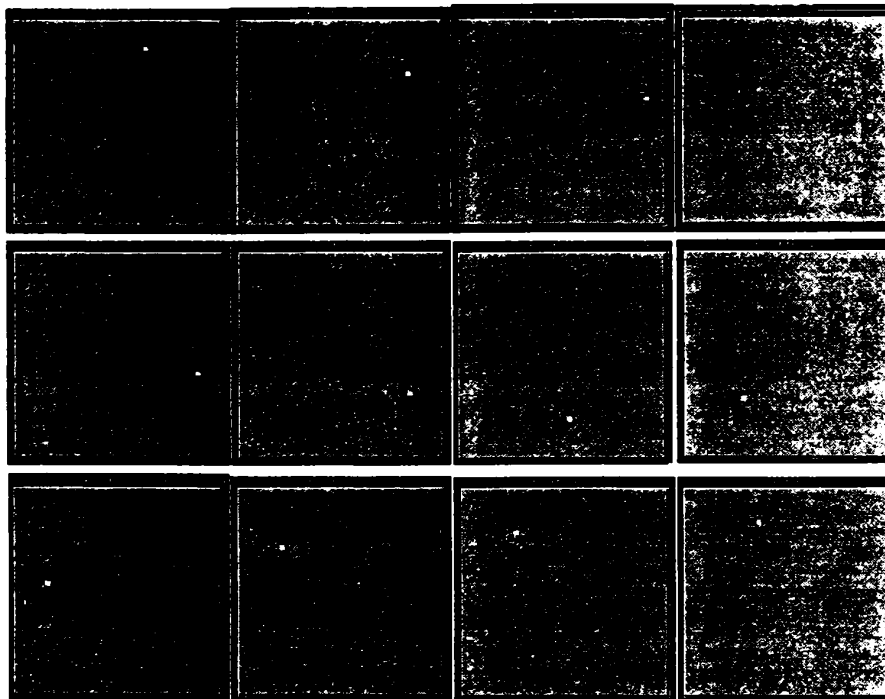
**Fig. 4.27** Reprezentarea tridimensională a imaginii de control pentru rotirea a doi pixeli

#### 4.4.3. Rezultatele simulării

Fig. 4.28 reprezintă starea inițială a RNC care realizează rotația unui pixel. Funcția dată în §4.4.2 este aplicată atât pentru generarea interconexiunilor de reacție din stratul 2 cât și ca funcție neliniară în stratul 2. În timpul procesării, pixelul negru din stratul de referință este deplasat pe conturul din stratul de control, cu o "urmă" devansată de un pixel negru și unul alb. Câteva dintre stările intermediare din timpul procesării sunt reprezentate în Fig. 4.29.



**Fig. 4.28** Starea inițială a RNC pentru rotirea unui pixel



**Fig. 4.29** Câteva stări intermediare ale RNC în timpul rotației unui pixel

## 4.5 Concluzii

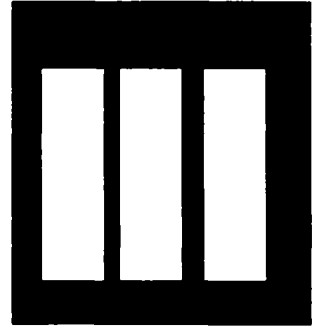
În acest capitol se prezintă câteva aplicații ale RNC ca operatori în prelucrările mai complexe de imagini. Scopul este exploatarea vitezei mari de calcul a RNC dată de dinamica lor analogică paralelă.

Se descrie o metodă generală de determinare a matricilor de conexiune pentru implementarea unui tabel de adevăr oarecare, care caracterizează comportarea rețelei în timp. Se prezintă rezultatele experimentării matricilor de conexiune obținute în detecția părții statice, respectiv a celei mobile a unei imagini în mișcare. Se abordează detecția mișcării unui obiect în condiții de zgomot. Aplicând imaginea obținută la intrarea unor straturi succesive se poate continua prelucrarea ei. Rezultatul procesării va fi același și în cazul imaginilor cu nuanțe pe scara de gri.

Se studiază reconstrucția obiectelor 3D prin interpolare și aproximare cu RNC și se prezintă rezultatele simulărilor. Se abordează problema rotației obiectelor 3D. Rotația poate fi implementată cu două straturi RNC invariante în spațiu. Metoda expusă poate fi generalizată în continuare prin implementarea altor operații variate în spațiu care pot fi codate local (micșorarea, mărirea obiectelor). Este posibilă utilizarea metodei și în algoritmi recursivi, cu o imagine de control în mișcare. Comportarea deosebit de complexă a acestor algoritmi necesită însă studii aprofundate.

Pentru o situație reală implementarea aplicațiilor prezentate poate fi făcută fie pe acceleratorul hard, fie VLSI, beneficiind de avantajul programabilității chipurilor de RNC.

# Partea



# Memorii asociative

# Capitolul 5

## MEMORII ASOCIATIVE. NOȚIUNI FUNDAMENTALE

### 5.1 Generalități

Într-o bază de date (memorie) implementată pe un calculator clasic memorarea se face sub formă de liste imbricate. Apelul unui model din memorie se face secvențial, ceea ce implică un timp îndelungat de acces, dependent de numărul elementelor memorate.

Într-o memorie asociativă neuronală apelul unui model se face printr-un proces de relaxare, de minimizare a unei funcții de energie asociate. Timpul de acces nu depinde de numărul modelelor memorate, ci este o caracteristică a arhitecturii rețelei.

Modelele memorate se găsesc în minimele energetice locale ale funcției Lyapunov care caracterizează sistemul (Fig.5.1). Fiind dat un model de intrare, rețeaua va converge spre una dintre stările sale stabile posibile. Găsirea unui minim local, cel mai apropiat de starea inițială a sistemului, oprește procesarea. Acest mecanism de căutare reprezintă avantajul major al memoriilor asociative neuronale față de cele clasice. El este similar recunoașterii de modele în sistemele nervoase biologice și explică de ce timpul necesar recunoașterii figurilor familiare este același la vârsta de cinci, douăzeci și cinci, sau cincizeci de ani.

Aglomerarea modelelor memorate poate afecta precizia clasificării, deoarece modelele similare pot avea bazine de atracție care se suprapun.

Se știe că dimensiunea rețelei trebuie să depășească numărul de modele pe care dorim să le memorăm, pentru ca sistemul să nu învețe modele noi cu prețul uitării altora.

În ultima decadă s-a depus un efort intens de cercetare în direcția memoriilor asociative implementate cu rețele neuronale celulare. Au fost elaborate modele analogice și discrete, ce implică atât conexiuni de ordinul întâi, cât și de ordin superior.

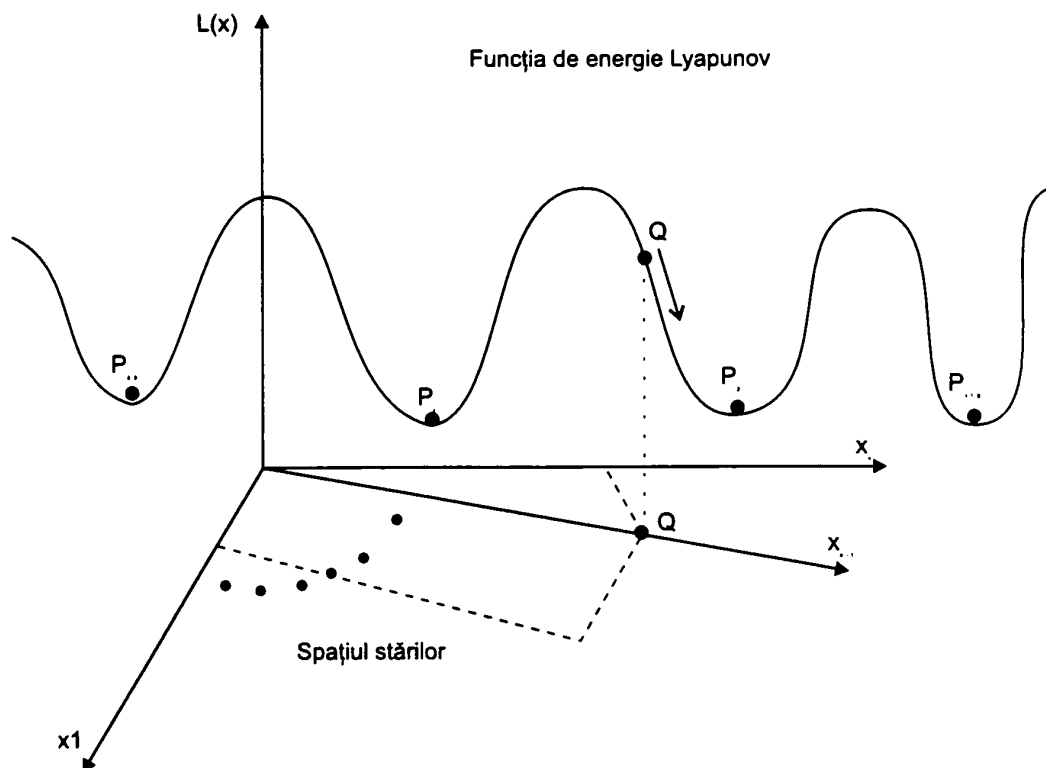


Fig. 5.1 Funcția Lyapunov care caracterizează o memorie asociativă

### Definiția 9

**Memoriile asociative** implementează o transformare între o mulțime de modele aparținând spațiului de intrare și o mulțime de modele aparținând spațiului de ieșire. Ori de câte ori la intrare se aplică un model particular, la ieșire se obține modelul asociat acestuia.

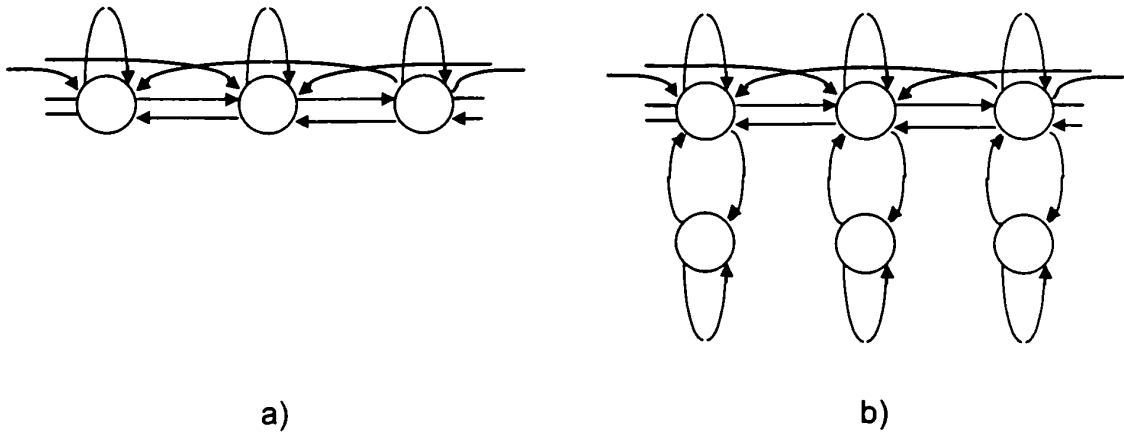
Un caz particular este **autoasocierea**, când un model este asociat cu el însuși. Scopul procesării este completarea de model sau eliminarea zgomotului, asociindu-se modelul de intrare, incomplet, sau afectat de zgomot cu el însuși la ieșire.

Există trei modalități de proiectare a rețelelor neuronale ca și memorii asociative, care asigură **convergența** către:

- puncte de echilibru stabil;
- soluții periodice;
- traiectorii haotice;

Fig. 5.2a prezintă structura unei rețele neuronale formată din unități stabile iar Fig.5.2b cea a unei RN formate din oscilatoare. Conexiunile de autoreacție și cele dintre diferitele unități sunt reprezentate prin linii, iar sensul legăturii prin săgeți.

Literatura de specialitate consemnează rezultate doar în implementarea unor memorii asociative cu puncte de echilibru stabil. Celelalte două modalități de implementare rămân subiecte de cercetare deschise.



**Fig. 5.2 Rețea neuronală celulară**  
**a) cu unități stabile; b) cu oscilatoare**

În implementarea unei rețele neuronale asociative există două faze:

- învățarea celor p perechi de modele de asociat;
- aplicarea la intrarea rețelei a oricăruia dintre modelele memorate va genera la ieșire modelul asociat (recall);

Scopul este obținerea unei memorii asociative cu :

- **capacitate de memorare** cât mai mare. Capacitatea de memorare este numărul modelelor distincte pe care sistemul le poate învăța cu precizie și rememora, deci coda și decoda.
- **capabilitate de corecție a erorilor**. Dacă se aplică la intrarea rețelei unul dintre modelele de intrare, afectat de o eroare, rețeaua ar trebui să convergă către modelul asociat dorit ( în condițiile unei erori cât mai mari);
- **evitarea modelelor nedorite** (spurious patterns). Parametrii rețelei trebuie ajustați astfel încât rețeaua să memoreze doar modelele dorite, nu și altele, parazite;
- **număr redus de interconexiuni**. Sunt de dorit interconexiuni de ordinul întâi, știut fiind faptul că cele de ordin superior generează probleme în implementările VLSI;
- **algoritm de determinare a ponderilor interconexiunilor rapid și aditiv** în cazul necesității încorporării unui model nou.

Figurile Fig.5.3a și Fig.5.3.b ilustrează caracteristicile unei MA performante, respectiv ale unei MA ineficiente . Modelele care reprezintă informația utilă sunt reprezentate prin cercuri goale, iar cele nedorite prin cercuri pline. Este de dorit ca în cazul în care modelele nedorite nu pot fi evitate ele să fie cât mai îndepărtate de punctele în care se memorează informația utilă.

Memoriile asociative pot fi implementate ca și sisteme dinamice continue:

$$\dot{x}(t) = F_w(x(t), u) \tag{5.1}$$

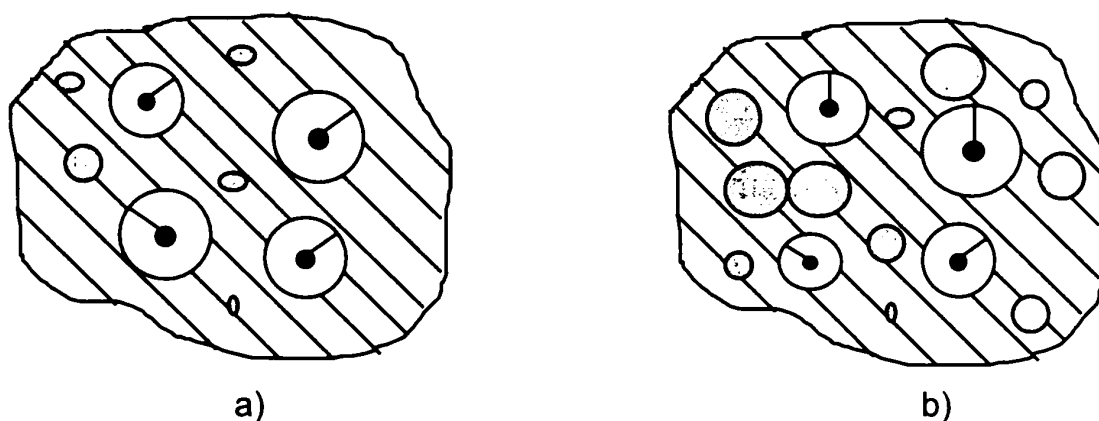
sau discrete:

$$x[n + 1] = F_w(x[n], u) \quad (5.2)$$

parametrizate de o matrice  $W$  (sau mai multe ) care codează asocierile, unde:

- $x$  este starea sistemului;
- $u$  este intrarea, care poate fi independentă sau aplicată ca o condiție inițială  $x(0)$ , respectiv  $x[0]$ .

Fiecare celulă dintr-o rețea de  $M \times N$  are o aceeași structură simplă, descrisă de ecuațiile (2.5) pentru RNCA sau (2.6) pentru RNCD.



**Fig. 5.3 Memorie asociativă**  
a) cu performanță ridicată; b) cu performanță redusă

## 5.2 Capacitatea de memorare

Există încă controverse asupra modului de definire a capacității de memorare.

Uzual capacitatea de memorare se referă la comportarea asimptotică a memoriei asociative atunci când numărul neuronilor tinde la infinit.

În [67] se propune următoarea definiție :

### **Definiția 10**

**Capacitatea de memorare** este numărul maxim al modelelor aleatoare ce se pot înmagazina cu o probabilitate mai mare decât un prag adecvat (de ex. 0.95).

Datorită localității conexiunilor este de așteptat ca în cazul RNC să avem o capacitate de memorare mai mică decât în cazul rețelei Hopfield care este total conectată.

Nu există o modalitate teoretică de determinare a capacității de memorare a unei memorii asociative implementate cu o RNC în cazul general. Experiența demonstrează că, în esență, capacitatea de memorare depinde de tipul interconexiunilor dintre celulele rețelei neuronale.



Ea a fost determinată în câteva cazuri particulare, pentru fiecare caz în parte (vezi cap.6, cap.7, cap.8, cap.9).

Pentru RNCD variante în spațiu, neregiproce există un studiu [67] care fundamentează estimarea empirică a capacității de memorare dependent de tipul, mărimea vecinătății și lățimea bazinului de atracție (cunoscută și sub denumirea de margine de stabilitate):

$$C = b \cdot c^\Delta \quad (5.3)$$

cu b și c date de relațiile:

$$b = s \cdot d + t$$

$$b = s' \cdot d + t'$$

unde:

- d este numărul de conexiuni;
- $s \cong 1.8$ ,  $t \cong -7.6$ ,  $s' \cong 0.0013$  și  $t' \cong 0.15$ ;
- $\Delta$  este marginea de stabilitate;

Relația a fost determinată în absența autoreacției, pentru a menține numărul stărilor nedorite la minimum, și în cazul considerării unor interconexiuni neregiproce, pentru a avea un grad de libertate cât mai mare.

### 5.3 Determinarea bazinului de atracție

Problema determinării bazinului de atracție prezintă o deosebită importanță în implementarea memoriilor asociative fiind legată atât de robustețea rețelei cât și de analiza convergenței și a stabilității. Pentru o rețea neuronală este deosebit de important ca atunci când la intrare se aplică un anumit model să existe un singur punct de echilibru în care să se stabilizeze rețeaua.

În [90] se abordează pentru prima oară această problemă în cazul unei RNC cu 2 celule. O extensie a metodei pentru estimarea a bazinului de atracție, pentru fiecare punct de echilibru al unei RNC de ordin mai mare, s-a publicat recent [68]. Metoda constă în determinarea așa numitului **arbore de regiuni** cuplate la fiecare punct de echilibru și se bazează pe următoarele definiții și teoreme:

#### Definiția 11

*O RNC operează în regiunea liniară*, subspațiul RL, dacă variabilele sale de stare nu sunt saturate

$$RL = \{x \in R^n : |x_i| < 1, \quad i = 1, 2, \dots, n\} \quad (5.4)$$

### Definiția 12

O RNC operează în regiunea  $k$  de saturație (RS)  $k=1,2,\dots,2^n$ , dacă toate variabilele sale de stare sunt saturate .

$$RS = \{x \in \mathbb{R}^n : |x_i| > 1, \quad i = 1, 2, \dots, n \} \quad (5.5)$$

### Definiția 13

O RNC operează într-o regiune de saturație parțială (RSP)  $k$ ,  $k=1,2,\dots,3^n-2^n-1^n$ , dacă unele dintre variabilele sale de stare sunt saturate și altele nu.

$$RSP = \{x \in \mathbb{R}^n : |x_i| > 1, |x_j| < 1, \quad i \in P, j \in Q \} \text{unde } P \cup Q = N, P \cap Q = \emptyset \quad (5.6)$$

### Definiția 14

*Arborele de regiuni* asociat unei regiuni de saturație cu un punct fix de echilibru (rădăcina) este format din rădăcină și ramuri, care sunt regiuni ale spațiului stărilor ce îndeplinesc următoarele condiții:

- regiunea de rădăcină conține un punct de echilibru stabil;
- fiecare ramură a arborelui constă din regiuni ce reprezintă o cale a traiectoriei de stare către regiunea rădăcină;

### Definiția 15

*Vectorul destinație a traiectoriei* are:

- dimensiunea egală cu numărul hiperplanurilor de separare a regiunii prin care trece:
  - pentru o RL dimensiunea este  $2n$ ;
  - pentru o RS dimensiunea este  $n$ ;
  - pentru o RSP dimensiunea este dependentă de tipul regiunii RSP, pe care o traversează traiectoria și poate fi  $n+1, n+2, \dots, n+n+1$ ;
- fiecare dintre componentele vectorului corespunde unui hiperplan de separare a regiunii și are valoarea:
  - 1 dacă traiectoria este direcționată spre exteriorul regiunii;
  - -1 dacă traiectoria este direcționată spre interiorul regiunii;
  - 0 dacă există traiectorii care sunt direcționate spre exteriorul regiunii și traiectorii direcționate spre interiorul regiunii;

### **Teorema 1**

Într-o regiune de saturație există cel mult un punct de echilibru.

### **Teorema 2**

Dacă există un punct de echilibru într-o regiune de saturație, atunci întreaga regiune de saturație aparține bazinului de atracție a punctului de echilibru.

### **Teorema 3**

Dacă este îndeplinită condiția  $A_{ii} > 1$  atunci orice punct de echilibru dintr-o regiune de saturație parțială și din regiunea liniară este instabil.

### **Teorema 4**

Toate stările unei RNC sunt mărginite în orice moment de timp  $t > 0$ .

### **Teorema 5**

Dacă există un punct de echilibru într-o regiune de saturație dată, atunci toate componentele traiectoriei  $n$  dimensionale sunt egale cu  $-1$ .

### **Teorema 6**

Se consideră componentele vectorului destinație a traiectoriilor pentru regiunile RL, RSP și RS fără puncte de echilibru. Dacă este îndeplinită condiția  $A_{ii} > 1$ , există cel puțin o componentă a vectorului destinație cu valoarea  $0$  sau  $+1$ .

### **Criteriu**

Dacă cel puțin una dintre componentele vectorului destinație a traiectoriei are valoarea  $0$  sau  $1$ , pentru o anumită regiune, atunci acea regiune va fi adăugată arborelui (dacă traiectoria aflată la limita dintre o anumită regiune și alte regiuni ale arborelui poate părăsi regiunea în cauză).

### **Teorema 7**

Dacă o regiune aparține unui singur arbore, toate regiunile ramurilor conectate prin această regiune cu rădăcina arborelui, aparțin bazinului de atracție a punctului de echilibru stabil (din rădăcină).

Demonstrațiile teoremelor 1-4 se găsesc în [28] și ale teoremelor 5-7 în [68].

## Procedura de determinare a bazinului de atracție

Fie regiunea de rădăcina RS care conține punctul de echilibru stabil. În conformitate cu Teorema 5, nivelul 2 al arborelui va conține toate cele n regiuni vecine regiunii de saturație considerate. Se iau în discuție acum toate regiunile vecine fiecărei regiuni de pe nivelul 2 al arborelui. Dacă este satisfăcut criteriul 1, regiunile considerate vor fi adăugate nivelului 3 al arborelui. Aceste regiuni pot fi de tip RSP, RS fără puncte de echilibru sau RL, și în conformitate cu Teorema 6, traiectoria stării va găsi o cale de a le părăsi. Procedura se oprește când toate regiunile posibile au fost vizitate și pentru fiecare punct de echilibru a fost generat un arbore.

Metoda este utilă și permite determinarea traiectoriei, respectiv a ieșirii în cazul în care se cunosc condițiile inițiale ce sunt situate într-o zonă ce aparține unui singur arbore de decizie. Ea permite determinarea regiunilor care aparțin în întregime bazinului de atracție al fiecărui punct de echilibru și a regiunilor în care se găsesc limitele dintre bazinele de atracție.

Metoda de estimare a bazinului de atracție este de ordinul întâi, pentru că atunci când condițiile inițiale sunt situate în zone ce aparțin mai multor arbori de decizie nu se știe unde va ajunge traiectoria ce pornește din ele.

## 5.4 Metode de proiectare

### 5.4.1 Pentru memorarea informației în punctele de echilibru stabil

Uzual, matricile de conexiune de control, de reacție și de polarizare sunt determinate astfel încât perechile de modele dorite să reprezinte puncte de echilibru stabil ale rețelei, cu un bazin de atracție de distanță metrică dată.

Numărul punctelor de echilibru pentru o RNC cu N celule este de  $2^N$ . Practic condițiile de proiectare fiind mai dificile decât în cazul binecunoscutei rețele complet conectate Hopfield, care are o limită a capacității de memorare dată, de  $0.14N$ , numărul maxim al modelelor ce se pot memora este mai mic (vezi §5.2)

Fiind dat modelul de memorat, pentru celula  $C_{ij}$ , din regiunea de saturație  $|y_{i,j}^c| = 1$  și intrarea  $u_{ij}$ , trebuie determinate matricile de conexiune astfel încât starea corespunzătoare  $|x_{i,j}^c| = 1$  să fie dată de:

$$x_{ij}^c = \sum_{k \in N_r(ij)} A_{ij,kl} \cdot y_{kl} [n] + \sum_{k \in N_r(ij)} B_{ij,kl} \cdot u_{kl} [n] + I_{ij} \quad (5.7)$$

iar derivata  $|x_{ij}^c| = 0$ . Condiția de convergență spre starea dorită  $y_{ij}^c$  sunt

$$\begin{aligned} x_{ij}^c &\geq +1 & \text{daca} & y_{ij}^c = +1 \\ x_{ij}^c &\leq -1 & \text{daca} & y_{ij}^c = -1 \end{aligned} \quad (5.8)$$

Cele două condiții se pot restrânge în inegalitatea:

$$y_{ij}^{\infty} \left( \sum_{kl \in N_r(ij)} A_{ij,kl} \cdot y_{ij}^{\infty} + \sum_{kl \in N_r(ij)} B_{ij,kl} \cdot u_{kl} + I_{ij} \right) \geq 1 \quad (5.9)$$

În general există P modele de memorat, date de  $|y_{i,j}^{\infty[q]}|$  și  $|u_{i,j}^{[q]}|$  modele de intrare posibile ( $q=1,2,\dots,P$ ). Pentru fiecare pereche de modele, model de intrare - model de ieșire dorit, se poate obține o inegalitate de forma (5.9). Sistemul de inegalități obținut se poate rezolva prin mai multe metode, ca de exemplu: metoda relaxării [67], algoritmul perceptronului [45], prin algoritmul ADATRON [73]. Pentru fiecare dintre acești algoritmi există o teoremă de convergență care garantează găsirea unei soluții dacă aceasta există.

În unele aplicații (ca de exemplu procesarea de imagini) este necesară invarianța la rotație, sau izotropia matricilor de conexiune. Toți algoritmi anterior menționați pot fi modificați pentru a încorpora aceste constrângeri suplimentare.

#### 5.4.2 Pentru memorarea informației ca oscilații periodice

Memorarea informației sub formă de oscilații oferă noi posibilități:

- cu același număr de parametri pot fi proiectate sisteme cu mult mai multe soluții periodice (potențiale modele de memorat) decât numărul punctelor de echilibru posibile. În cazul extrem, sistemele haotice au un număr infinit de soluții periodice instabile.

Aplicând la intrare un model identic sau unul apropiat de unul memorat, traiectoria sistemului va fi blocată în ciclul periodic corespunzător, sau va vizita mai des zona în care modelul corespunzător este memorat.

- Oscilatoarele permit și operația de “segmentare”.  
Fie două modele binare în care proporția de 1 este mult mai mică decât cea de 0. Dacă se aplică la intrarea rețelei unul dintre modele, rețeaua îl va regăsi în maniera anterior prezentată. Dacă se prezintă intrării un model care este suma celor două, traiectoria sistemului va trece de la un model la celălalt. Aceasta reprezintă cea mai simplă versiune a segmentării.
- Sistemele haotice au și o altă proprietate utilă, o soluție se “plimbă” în jurul atractorului.
- În cazul nerecunoașterii modelului, traiectoria parcurge toate modelele fără să demonstreze o corelație mai puternică cu unele decât cu celelalte. Eșecul căutării poate fi interpretat ca absența unui model anumit în memorie, în timp ce sistemele bazate pe puncte fixe de echilibru se vor bloca în stări nedorite. Fenomenul este similar celui din cortexul olfactiv, unde în absența stimulului traiectoria se “plimbă” într-o stare bazală haotică, care este un fel de attractor cu aripi multiple.

Exploatarea potențialului oscilațiilor stabile sau instabile pentru regăsirea unui model este oricum dificilă. Au fost propuse mai multe metode pentru stabilizarea soluțiilor

periodice instabile ,care formează teoria “controlului haosului”. Deși au fost raportate câteva rezultate pozitive în această direcție [100], [1], se știe foarte puțin despre memorarea unui anumit model dorit, despre proiectarea unui sistem cu orbite periodice dorite.

## 5.5 Concluzii

În acest capitol s-au introdus și discutat noțiunile fundamentale ale MA și metodele specifice de determinare a matricilor de interconexiuni.

A fost dată o relație aproximativă, empirică, de estimare a capacității de memorare dependentă de tipul , mărimea vecinătății și lărgimea bazinului de atracție, în absența autoreacției. Capacitatea de memorare a fost calculată doar pentru câteva cazuri concrete, care vor fi trecute în revistă în capitolele ce urmează, determinarea ei teoretică în cazul unei RNC oarecare rămânând o problemă de cercetare deschisă.

A fost abordată și problema bazinului de atracție, expunându-se o metodă de ordinul întâi pentru determinarea traiectoriei, respectiv a ieșirii în cazul în care se cunosc condițiile inițiale ce sunt situate într-o zonă ce aparține unui singur arbore de decizie.

În ceea ce privește modelele nedorite, în cele ce urmează, se va preciza pentru fiecare metodă în parte, dacă procedura de sinteză permite sau nu evitarea lor.

Metodele uzuale de proiectare a MA prin RNC urmăresc memorarea informației ca puncte fixe de echilibru ale sistemului. S-au înregistrat rezultate promițătoare în memorarea informației ca oscilații periodice stabile și nestabile (generate de sisteme haotice). Cercetările neurologice (în special în simțul mirosului) din ultimii ani vin să confirme ipoteza că modul procesare al informației în creierul uman este de tip “haotic”. Din păcate până în prezent se știe foarte puțin despre proiectarea unui sistem haotic care să memoreze modele dorite. Exploatarea potențialului de procesare a sistemelor haotice deschide noi perspective MA.

Algoritmii genetici nu pot fi folosiți în cazul MA implementate prin RMC. Aceștia funcționează bine cu un număr redus de necunoscute, dar chiar și atunci necesită un timp îndelungat de procesare. În cazul memoriilor asociative, se lucrează cu matrici de interconexiune variabile în spațiu, astfel încât numărul parametrilor de decizie este mare, fiecare celulă având propriul ei template.

# Capitolul 6

## MEMORII ASOCIATIVE CE UTILIZEAZĂ REGULA LUI HEBB

### 6.1 Modelul Vandewaale

#### 6.1.1 Caracteristicile rețelei și condițiile impuse modelelor de memorat

Pentru simplitatea analizei se consideră exclusiv versiunea în timp discret a RNC [86] fără reacție, cu polarizare nulă și actualizare sincronă:

$$x_{ij}[n + 1] = f\left(\sum_{C_{kl} \in N_r(ij)} B_{ij,kl} u_{kl}[n]\right) \quad (6.1)$$

unde  $f(\cdot)$  este funcția  $\text{sign}(\cdot)$ ;

Fie un set de  $P$  modele binare, de dimensiune  $M \times N$ , și  $b$  o distanță de regăsire dată ( $b$  se consideră în sensul distanței Hamming, astfel încât un model diferit de un model oarecare  $q$ , prin distanța de regăsire  $b$ , ar trebui să conducă la regăsirea modelului  $q$ ). Cea mai simplă metodă de determinare a conexiunilor astfel încât modelele binare date să reprezinte puncte fixe stabile ale rețelei cu bazinul de atracție  $b$  este regula lui Hebb [50]. Dacă proiectarea garantează realizarea bazinelor de atracție de mărime prescrisă, pentru fiecare punct stabil, aspecte ca stările parazite stabile, noțiunea de completă stabilitate devin irelevante. Cel mai important aspect devine capacitatea de memorare.

Considerând cazul RNC cu vecinătate unu vom face o analiza locală, asupra submodelului elementar de dimensiune  $3 \times 3$  celule, rezultatul putând fi extins asupra întregii rețele [97]. Submodelul centrat în jurul celulei  $C_{ij}$  este :

$$N_{ij}(\mathbf{X}) = \begin{bmatrix} x_{i-1,j-1} & x_{i,j-1} & x_{i+1,j-1} \\ x_{i-1,j} & x_{i,j} & x_{i+1,j} \\ x_{i-1,j+1} & x_{i,j+1} & x_{i+1,j+1} \end{bmatrix} \quad (6.2)$$

Pentru toate cele P modele de memorat 1, 2, ..., P, se poate defini o succesiune de submodele  $X^1, X^2, \dots, X^P$ , pentru fiecare celulă  $C_{ij}$ ,  $i=1, 2, \dots, M, j=1, 2, \dots, N$

$$X_{ij}^q = N_{ij}(q) \quad \text{unde } q = 1, 2, \dots, P \quad (6.3)$$

Aplicând regula lui Hebb, ponderile conexiunilor de control ale celulei  $C_{ij}$  pot fi determinate cu:

$$B_{ij,kl} = \sum_{q=1}^P X_{kl}^q X_{ij}^q \quad \text{ptr. } i-1 \leq k \leq i+1 \text{ si } j-1 \leq l \leq j+1 \quad (6.4)$$

Se observă că regula lui Hebb este locală, în sensul că orice pondere este determinată de stările celulelor învecinate și aditivă, astfel încât orice model adițional poate fi ușor încorporat în proiectare, prin adunarea ponderilor determinate de noul model la matricea de conexiune a rețelei deja proiectate.

Fie  $h$  cea mai mică distanța Hamming (definită conform Anexei 1) dintre submodelele

$X^1, X^2, \dots, X^P$ .

### Teorema 8

Condiția necesară și suficientă de regăsire, în bazinul de atracție de rază  $b$ , a P submodele memorate cu regula lui Hebb este dată de inegalitatea [97]:

$$9 - 2b - (P - 1)(9 - 2h + 2b) > 0 \quad (6.5)$$

### Demonstrație:

Se observă că pentru cele P submodele corespunzând celulei  $C_{ij}$ , pentru  $i-1 \leq k \leq i+1$  și  $j-1 \leq l \leq j+1$ , avem:

$$\sum_{kl} X_{kl}^q X_{kl}^t = \begin{cases} 9 - 2h \quad \text{ptr. } q \neq t \quad (q, t = 1, 2, \dots, p) \\ 9 \quad \text{ptr. } l = t \quad (q, t = 1, 2, \dots, p) \end{cases} \quad (6.6)$$

Fie  $\bar{X}^t$  un model dat, la distanța Hamming  $b$  de submodelul  $X^t$ . Conform definiției, se poate vedea imediat că:

$$\sum_{kl} X_{kl}^q \cdot \bar{X}_{kl}^t = \begin{cases} \leq 9 - 2h + 2b \quad \text{ptr. } q \neq t \quad (q, t = 1, 2, \dots, p) \\ \geq 9 - 2b \quad \text{ptr. } q = t \quad (q, t = 1, 2, \dots, p) \end{cases} \quad (6.7)$$

Dacă  $\bar{X}^t$  este aplicat ca stare inițială, considerând polarizarea nulă, intrarea în celula  $C_{ij}$  va fi:



$$U_{ij} = \sum_{kl} B_{ij,kl} \cdot \bar{X}_{kl} = \sum_{kl} \left( \sum_{q=1}^p X_{kl}^q X_{jkl}^q \right) \cdot \bar{X}_{kl} = \sum_{q=1}^p X_{ij}^q \left( \sum_{kl} X_{kl}^q \bar{X}_{kl} \right) \quad (6.8)$$

== Pentru ca rețeaua să convergă către submodelul  $X^t$ , atunci când la intrare se aplică  $X^t$ , este necesar să fie îndeplinite următoarele două inegalități:

$$(a) X_{i,j}^t = +1 \quad U_{i,j} \geq (9 - 2b) \cdot (+1) - (p - 1) \cdot (9 - 2h + 2b) > 0 \quad (6.9)$$

$$(b) X_{i,j}^t = -1 \quad U_{i,j} \leq (9 - 2b) \cdot (-1) + (p - 1) \cdot (9 - 2h + 2b) < 0 \quad (6.10)$$

Din inegalitățile (6.9) și (6.10) obținem condiția (6.5), q.e.d.

Acest rezultat local poate fi extins la întreaga rețea, de dimensiune  $M \times N = 3m \times 3n$ . Fie  $b_{\min}$ ,  $b_{\max}$ , distanța minimă, respectiv maximă a regăsirii locale, iar  $p_{\min}$ ,  $p_{\max}$ , numărul minim, respectiv maxim al submodelelor memorate. Din teorema 1 rezultă ca numărul modelelor care pot fi memorate,  $C$ , de întreaga rețea cu distanța de regăsire  $d$ , este:

$$p_{\min}^{m \cdot n} \leq C \leq p_{\max}^{m \cdot n} \quad (6.11)$$

unde  $d$  este limitată în domeniul dat de relația:

$$m \cdot n \cdot b_{\min} \leq d \leq m \cdot n \cdot b_{\max} \quad (6.12)$$

Analiza locală a relevat câteva restricții ale memoriei asociative. Dacă fiecare submodel de dimensiune  $3 \times 3$  trebuie să memoreze cel puțin două submodele, atunci din relația (6.5) rezultă că  $h$ , cea mai mică distanța Hamming dintre submodele trebuie să îndeplinească condiția:  $h \geq 2b$ .

De exemplu dacă  $b=3$ ,  $h \geq 6$ . Se constată că deși potențial capacitatea de memorare este relativ mare (de ex. ptr. o rețea de  $30 \times 30$  și ptr. o distanță de regăsire  $b=2$ ,  $C=2^{100}$  cu  $d=200$ ), practic există limitări serioase.

### 6.1.2 Rezultatele simulărilor

Experimentul realizat a constatat în memorarea celor trei modele ideale  $P_1$ ,  $P_2$ ,  $P_3$ , din Fig.6.1, de dimensiune  $12 \times 12$ , raportate și în [97], aplicate ca intrare în RNC. S-a utilizat o valoare de  $+1$  pentru pixelii negri și  $-1$  pentru cei albi. Matricile interconexiunilor de control au fost determinate în Borland C, utilizând regula lui Hebb. Stările inițiale zgomotoase au fost obținute prin inversarea pixelilor din modelele dorite. Pentru verificarea memoriei proiectate a fost folosit simulatorul soft de RNC [115], variant în spațiu. Modelele de intrare, aplicate ca intrare independentă, au avut 11% din pixeli

inversați față de cele ideale. Se poate constata că la ieșirea rețelei RNC au fost obținute modelele ideale.

Practic, memoria are limitări serioase, în ceea ce privește condițiile ce trebuie îndeplinite de modelele de memorat, astfel încât ea prezintă doar interes istoric, fiind unul dintre primele MA implementate.

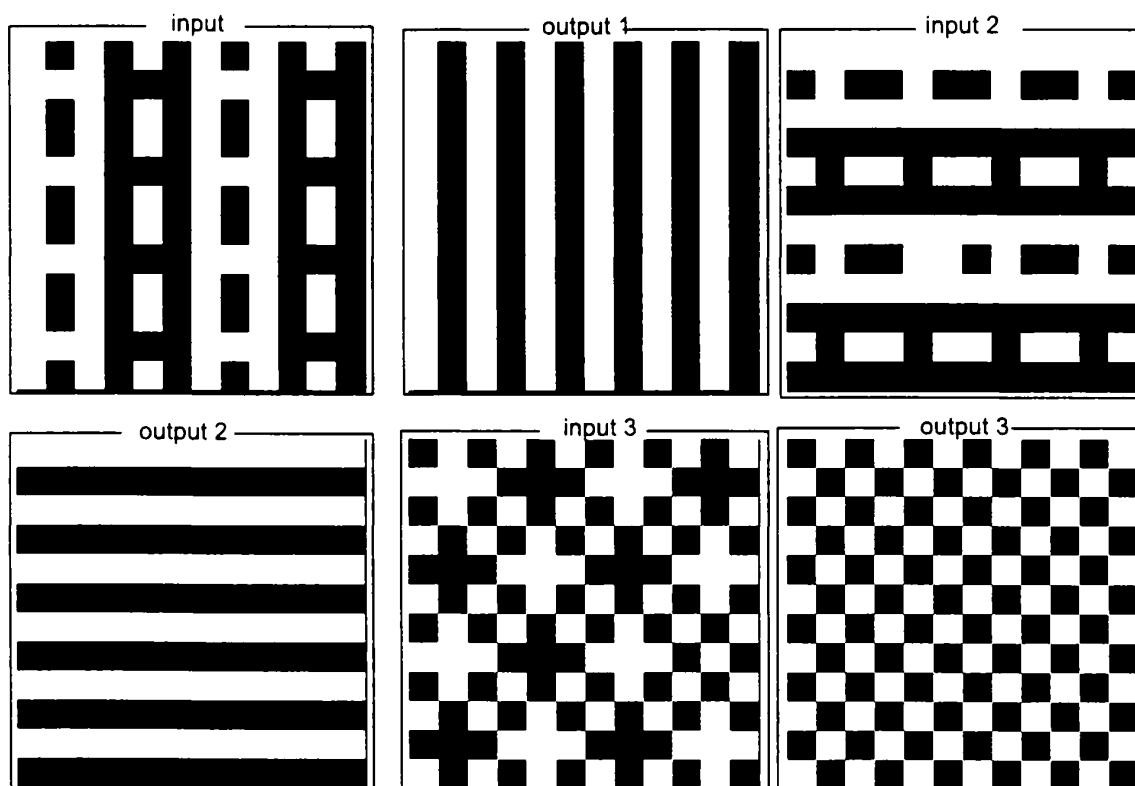
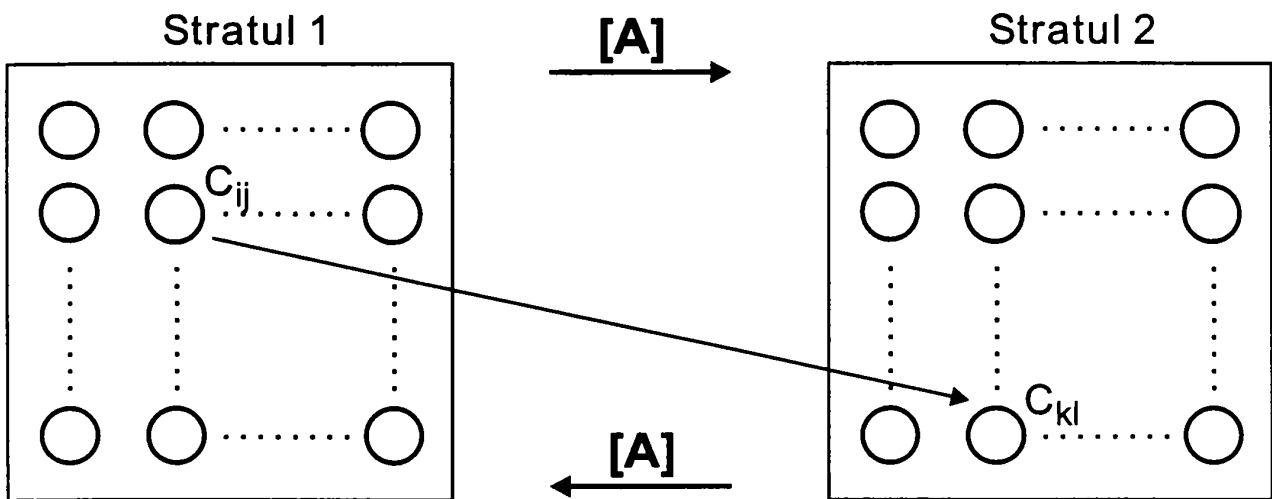


Fig. 6.1 Un exemplu de simulare a unei memorii asociative cu modelul Vandewaale

## 6.2 Memorie asociativă cu reacție inversă

### 6.2.1 Structura rețelei și dinamica

Pornind de la ideea memoriei asociative bidirecționale [57], se propune o structură neuronală celulară discretă cu două straturi, având o reacție inversă între cele două straturi dată de regula lui Hebb [11] și [12], ilustrată în Fig.6.2. . Fiecare strat are  $M \times N$  celule cu o aceeași structură simplă.



**Fig.6.2 Memorie asociativă bidirecțională**

Se poate deduce ușor că ecuațiile de stare care caracterizează dinamica rețelei sunt următoarele:

$$x_{1,ij}[n+1] = \sum_{C_{k,l} \in N_r(i,j)} A_{2,ij,kl} y_{2,kl}[n] \quad (6.13)$$

$$x_{2,ij}[n+1] = \sum_{C_{k,l} \in N_r(i,j)} A_{1,ij,kl} y_{1,kl}[n] \quad (6.14)$$

for  $1 \leq i, k \leq M$  ;  $1 \leq j, l \leq N$

unde:

- $x_{1,ij}$  ( $x_{2,ij}$ ) este starea celulei  $C_{i,j}$  al primului (celui de al doilea) strat ;
- $A_{2,ij,kl}$  este matricea ponderilor conexiunii de la ieșirea  $y_{2,kl}$  la intrarea celulei  $C_{1,ij}$ , adică :

$$A_{ij,ij} = a \cdot \sum_{q=1}^P y_{ij}^q \cdot y_{ij}^q$$

- $b$  și  $a$  constant,  $b \in (0,1)$ ;
- $y_{1,kl}$  ( $y_{2,kl}$ ) este ieșirea celulei  $C_{k,l}$  al primului (celui de al doilea) strat. Funcția de ieșire este signum ;
- $N_r$  este vecinătatea celulei  $C_{i,j}$  de dimensiune  $r$ ;

Se poate observa ușor că, utilizând regula lui Hebb în determinarea matricilor de conexiune, se implementează binecunoscuta regulă de învățare din memoria asociativă

bidirecțională ( $A=X^T \cdot X$ ), aplicată pe vecinătatea fiecărei celule (structura elementară locală) a RNC.

Ecuatiile (6.13) și (6.14) sunt adevărate pentru orice celulă  $C_{ij}$  și în orice moment discret de timp  $n$ . Celulele unui strat sunt actualizate sincron. Ponderea conexiunii între celula  $C_{i,j}$  și celula  $C_{kl}$  este calculată cu regula lui Hebb și este egală cu cea a conexiunii inverse de la  $C_{kl}$  la  $C_{ij}$ . Fiecare conexiune este bidirecțională și există o singură pondere pentru fiecare legătură. Matricile de conexiune de reacție inversă pentru fiecare celulă  $C_{ij}$  sunt dependente de coordonatele celulei, astfel încât este necesară utilizarea simulatorul soft pentru RNC variante în spațiu [115].

## 6.2.2 Caracteristicile rețelei și condiții impuse modelelor de memorat

Se dorește determinarea matricilor de conexiune astfel încât  $P$  modele dorite  $1, 2, \dots, P$  să fie puncte de echilibru stabil ale unei RNCD cu două straturi bidimensionale  $M \times N$ , conectate printr-o reacție inversă într-o vecinătate  $r$ . Dacă metoda de proiectare garantează dimensiunea bazinului de atracție aspecte ca modele nedorite sau stabilitatea completă devin neimportante. Capacitatea de memorare devine cel mai important aspect [97].

Se consideră structura elementară de dimensiune  $3 \times 3$  în jurul celulei  $C_{ij}$  dată de relația (6.2) respectiv secvența de submodele  $X^1, X^2, \dots, X^P$ , pentru toate cele  $P$  modele de memorat definită de (6.2). Matricea de conexiune pentru celula  $C_{ij}$  se determină cu:

$$A_{ij,kl} = \sum_{q=1}^P X_{k,l}^q X_{ij}^q \quad \text{ptr. } i-1 \leq k \leq i+1 \text{ și } j-1 \leq l \leq j+1 \quad (6.15)$$

Fie  $h$  cea mai mică distanță Hamming între submodelele de memorat  $X^1, X^2, \dots, X^P$ , pentru celula  $C_{ij}$ .

Condiția de convergență spre aceste puncte de echilibru poate fi scrisă ca în relația [77]:

$$y_{i,j}[n+1] \sum_{C_{kl} \in N_r} A_{ij,kl} y_{k,l}[n-1] > 0 \quad (6.16)$$

În condițiile date, relațiile (6.6) și (6.7) rămân valabile. Dacă  $\bar{X}^t$  este starea inițială de ieșire a celulei elementare de  $3 \times 3$ , deoarece modelele de memorat sunt binare și funcția de ieșire este signum, starea  $X_{ij}$  va fi:

$$X_{ij} = \sum_{k,l} A_{i,j;k,l} \cdot \bar{X}_{kl}^t = \sum_{k,l} \left( \sum_{q=1}^P X_{k,l}^q X_{i,j}^q \right) \cdot \bar{X}_{kl}^t = \sum_{q=1}^P X_{ij}^q \left( \sum_{k,l} X_{k,l}^q \bar{X}_{kl}^t \right) \quad (6.17)$$

pentru  $i-1 \leq k \leq i+1, j-1 \leq l \leq j+1$

Astfel condiția de convergență (6.16) (la  $X^t$ , când intrarea este  $\overline{X}^t$ ) poate fi scrisă:

$$(a) Y_{i,j}^1 = +1 \quad X_{i,j} \geq (9 - 2b) \cdot (+1) - (p - 1) \cdot (9 - 2h + 2b) > 0 \quad (6.18)$$

$$(b) Y_{i,j}^1 = -1 \quad X_{i,j} \leq (9 - 2b) \cdot (-1) + (p - 1) \cdot (9 - 2h + 2b) < 0 \quad (6.19)$$

Condiția necesară și suficientă ca P modele să fie memorate cu un grad de regăsire locală b, utilizând regula lui Hebb poate fi dedusă din ultimele două inegalități :

$$9 - 2b - (p - 1)(9 - 2h + 2b) > 0 \quad (6.20)$$

Dacă fiecare structură elementară de 3x3 trebuie să memoreze cel puțin două submodele, atunci din (6.20) se observă că distanța minimă dintre submodele trebuie să satisfacă condiția  $h \geq 2b$ . Pentru un grad de regăsire locală  $b \geq 2$  ( $0 < b < 9$ ) distanța minimă dintre modele trebuie să fie mai mare decât patru. Astfel distanța dintre două modele poate fi calculată cu relația:

$$d = b \cdot m \cdot n \quad (6.21)$$

Aceasta reprezintă o limitare serioasă asupra modelelor de memorat.

### 6.2.3 Capacitatea de memorare

Considerând b gradul de regăsire locală al unei RNC, cu o vecinătate r egală cu 1, se poate obține o formulă simplă pentru calculul capacității de memorare:

$$C = b \cdot m \cdot n \quad (6.22)$$

Practic capacitatea de memorare este chiar mai mică datorită celulelor de margine, pentru care vecinătatea trebuie completată cu celule de stare nulă.

### 6.2.4 Rezultatele simulărilor

Programul pentru determinarea matricilor de conexiune de reacție a fost implementat în Borland C++, utilizându-se regula lui Hebb. S-a utilizat o valoare de +1 pentru pixelii negri și -1 pentru cei albi. Pentru celulele din marginea rețelei, pentru construirea matricilor de interconexiune, s-au extins modelele cu pixeli de valoare zero (în vecinătatea aleasă). Modelele de intrare au fost aplicate ca stare a RNC. Aceasta a permis modificarea dinamică a stării RN sub acțiunea interconexiunilor de reacție .

Memoria asociativă obținută a fost testată cu simulatorul soft de RNC variante în spațiu.

MAB, obținută utilizând regula lui Hebb, a fost testată pentru diferite modele, care respectă sau nu condițiile restrictive ale algoritmului (6.20), modele de diferite dimensiuni, diferite vecinătăți și în diferite condiții de zgomot (gaussian și binar). Modelele afectate de zgomot binar au fost obținute din cele ideale prin inversarea valorii pixelilor.

## Exemplul 1

Am testat memorarea a cinci modele de dimensiune 10x10 (conform ieșirilor din Fig.6.3), care satisfac condițiile de proiectare (6.20), utilizând o vecinătate de  $r=1$ , în condiții de zgomot gaussian cu medie nulă și dispersie  $\sigma \in (0,1)$  și de zgomot binar.

Cele cinci modelele au fost corect apelate din memorie pentru un zgomot gaussian  $\sigma=0.9$  (Fig.6.3). Pentru  $\sigma=1$  patru modele au fost corect apelate din memorie și unul cu o eroare de 3 pixeli. Fig.6.4 ilustrează modelele de intrare aplicate, având o eroare de inversare a pixelilor de 25% și modelele de ieșire spre care a convers RNC, reprezentând modelele ideale memorate.

Prin memorarea modelelor complementare s-a obținut o aceeași matrice de interconexiune, după cum era de așteptat [57].

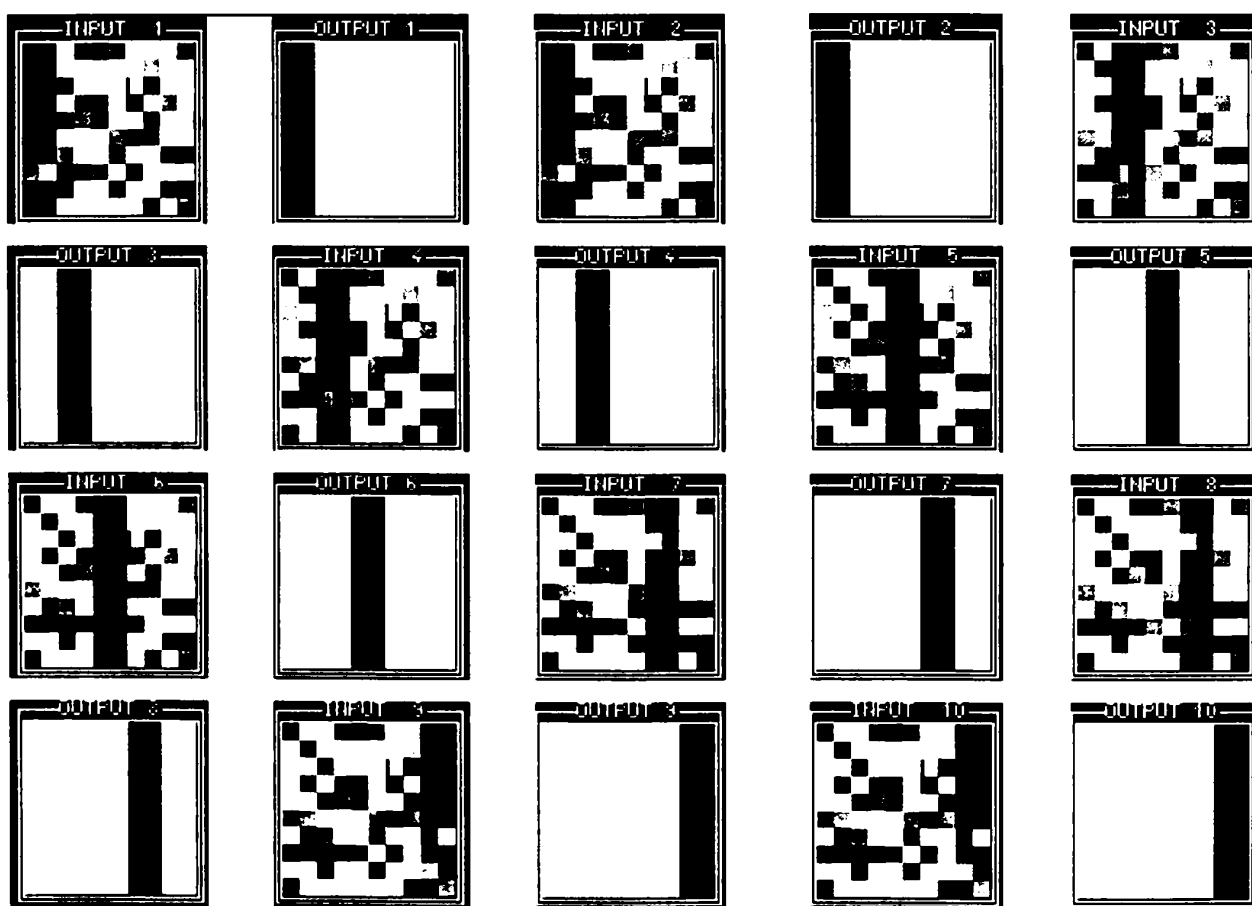


Fig. 6.3 Modelele de intrare afectate de un zgomot gaussian cu medie nulă și dispersie  $\sigma = 0.9$  și ieșirile corespunzătoare unei MAB de dimensiune 10x10

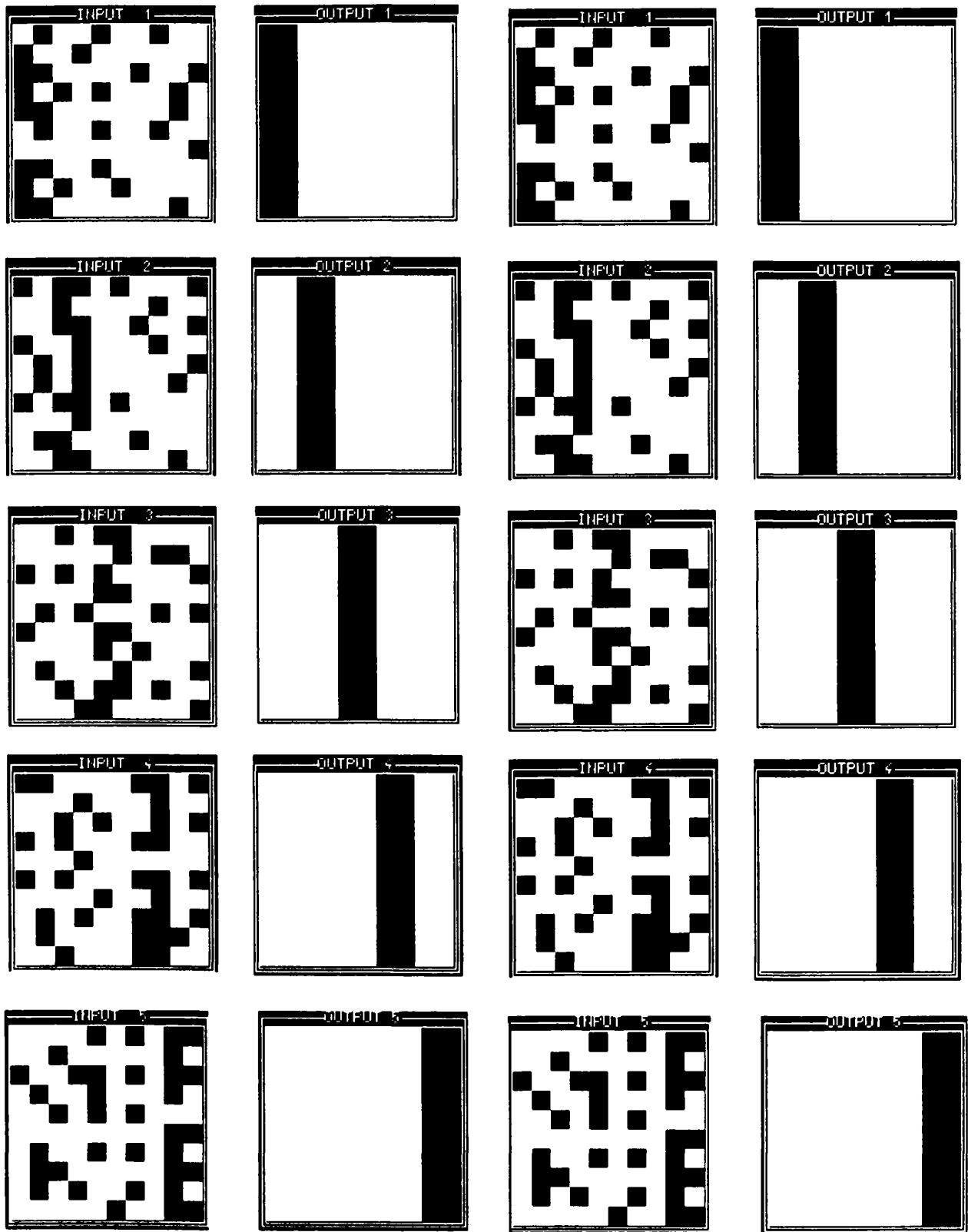


Fig. 6.4 Modelele de intrare, reprezentând cele 5 modele memorate afectate cu o eroare binară de 25% și cele de ieșire după stabilizarea unei RNC de dimensiune 10x10

## Exemplul 2

Cele zece modele din Fig.6.5 au fost memorate cu o MAB de dimensiune 10x10. În alegerea modelelor nu s-a urmărit respectarea restricțiilor de proiectare cerute de algoritm, ci s-au utilizat modele uzuale de testare a memoriilor asociative neuronale, cu o suprapunere cât mai mare, pentru a testa eficacitatea memoriei.

Fig.6.6 prezintă modelele de intrare afectate cu un zgomot gaussian cu medie nulă și dispersie  $\sigma=0.4$ . Figura ilustrează intrarea și ieșirea doar a unuia dintre cele două straturi, pentru cel de-al doilea ele fiind identice. Memoria a regăsit cinci dintre modelele ideale, celelalte fiind afectate cu o eroare de 3-4 pixeli.

Fig. 6.7 prezintă rezultatele procesării pentru o eroare binară a modelelor de intrare de 15%, 7 modele au fost corect apelate din memorie, 2 cu 1 pixel eroare și unul cu 6 pixeli eroare.

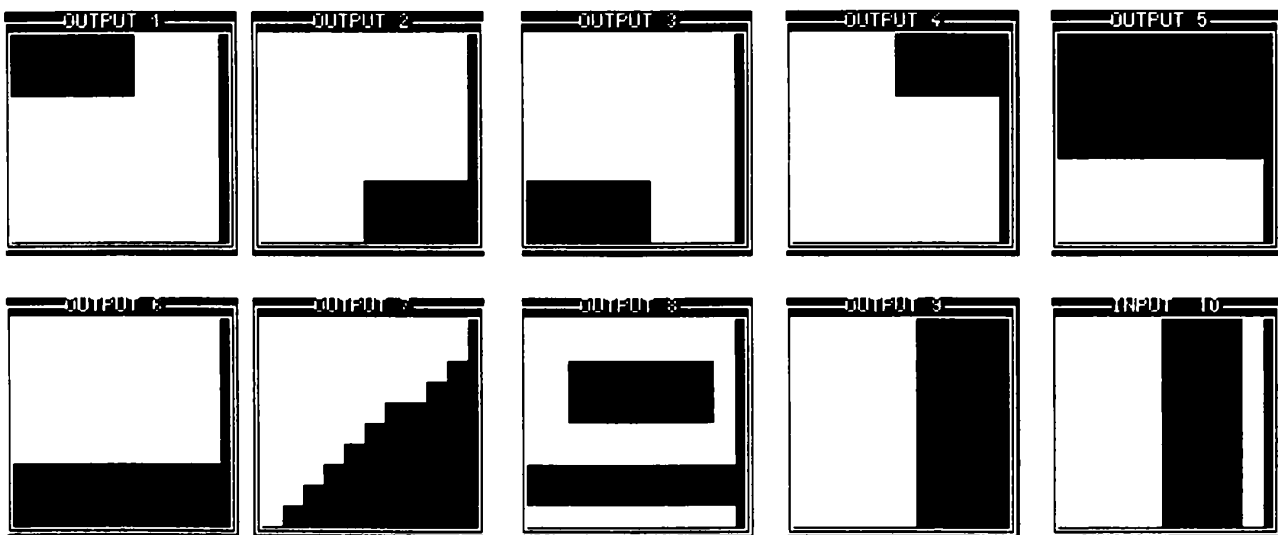


Fig.6.5 Modelele ideale memorate de o MAB de dimensiune 10x10



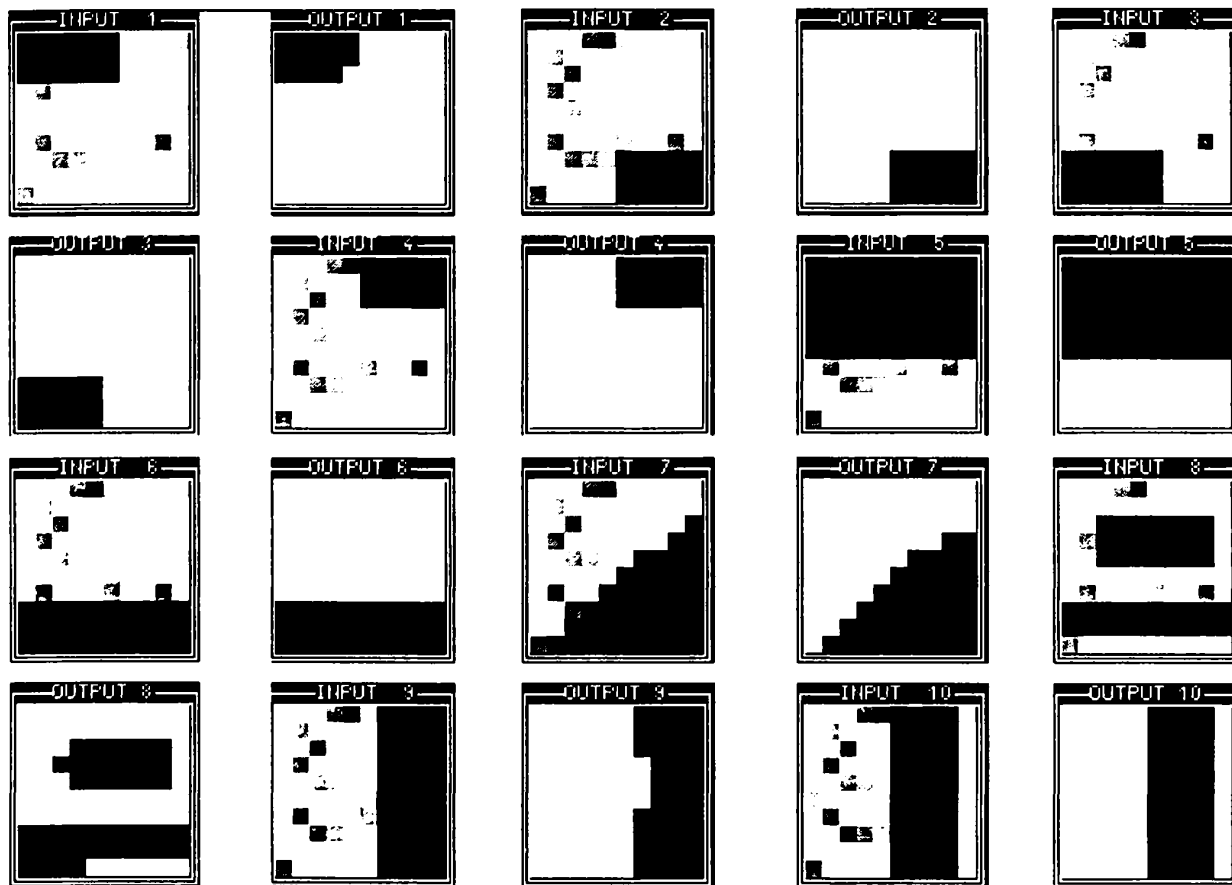


Fig. 6.6 Cele zece modele de intrare afectate de un zgomot gaussian cu medie nula si dispersie  $\sigma=0.4$  și ieșirile corespunzătoare

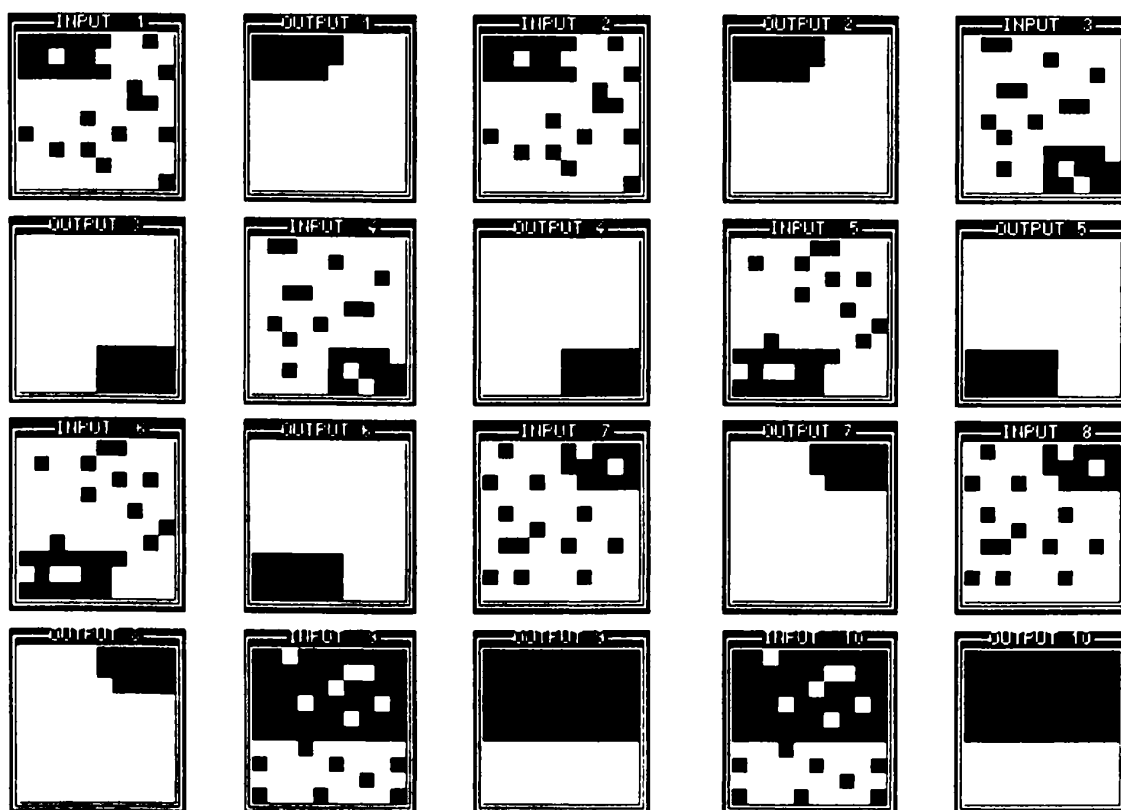


Fig.6.7 Modelele de intrare, reprezentând cele 10 modele memorate afectate de un zgomot binar 15% și cele de ieșire după stabilizarea rețelei de dimensiune 10x10

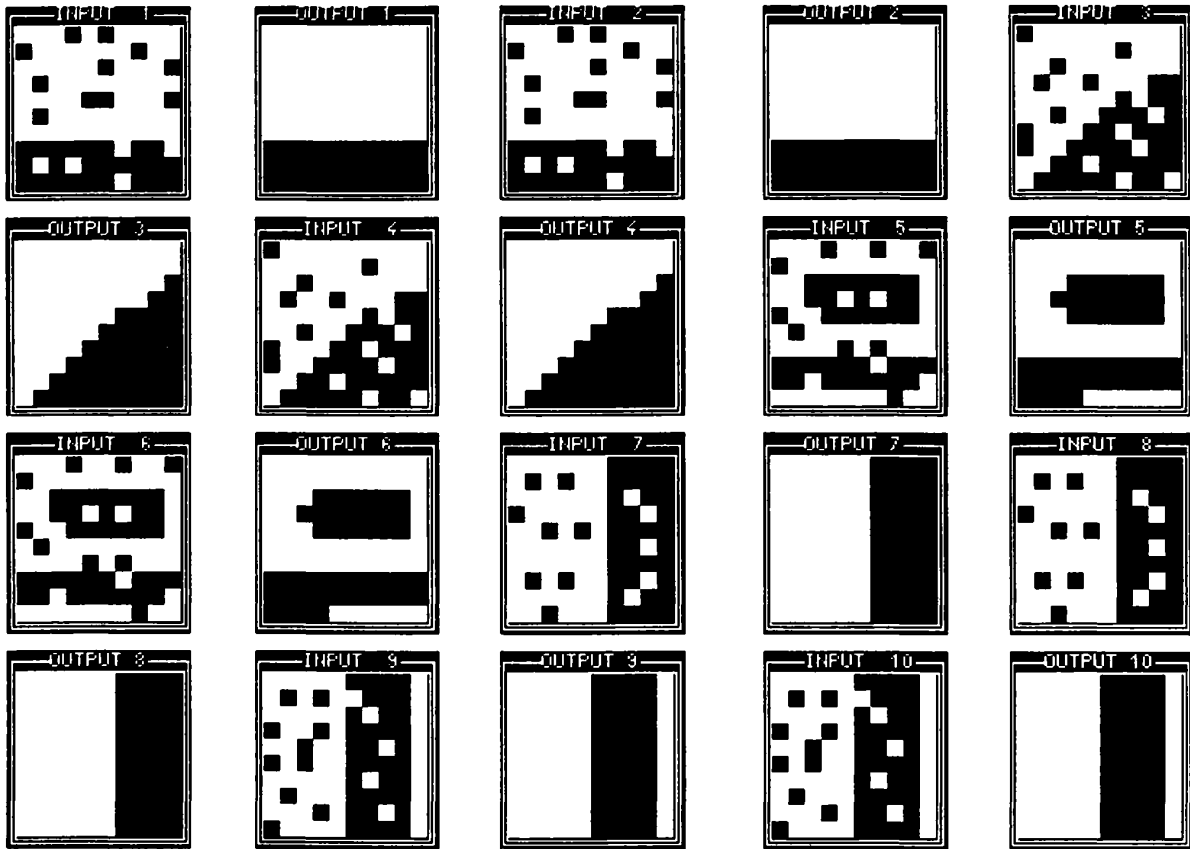


Fig. 6.7 Continuare

### Exemplul 3

Modele similare cu cele din exemplul 2 au fost memorate cu o rețea de dimensiune 12x12, după cum se vede în Fig.6.8.

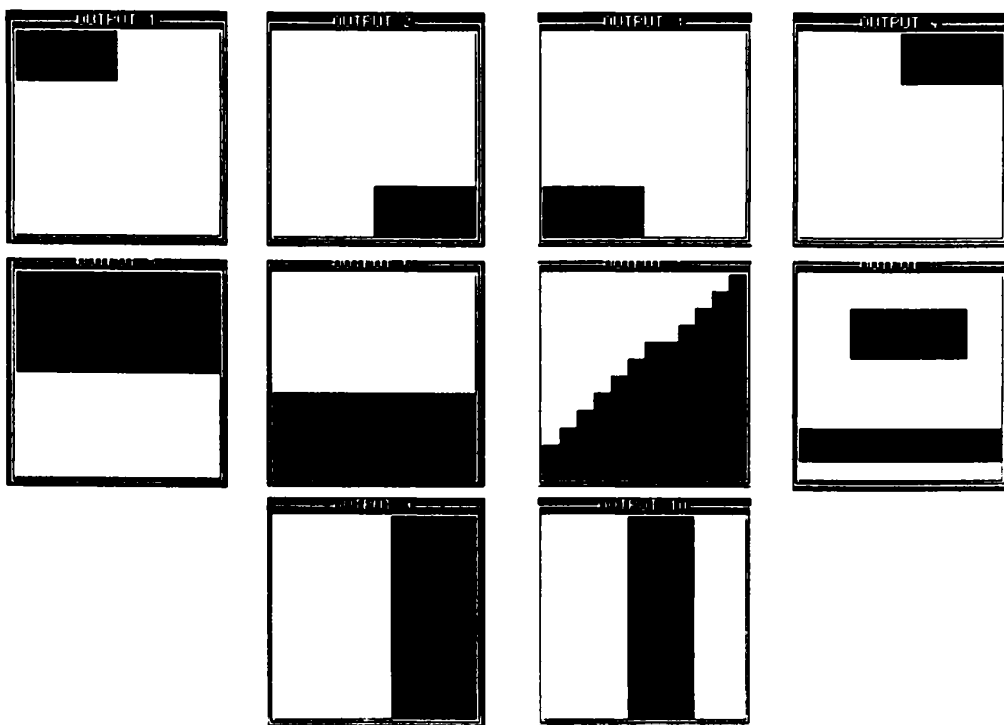


Fig.6.8 Modelele ideale memorate de o MAB de dimensiune 12x12

Fig.6.9 prezintă modelele de intrare afectate cu un zgomot gaussian cu medie nulă și dispersie  $\sigma=0.4$ . Figura ilustrează intrarea și ieșirea doar a unuia dintre cele două straturi, pentru cel de-al doilea ele fiind identice. RNC a apelat 7 modele corect din memorie, două cu o eroare de 1 pixel și unul cu 6 pixeli eroare.

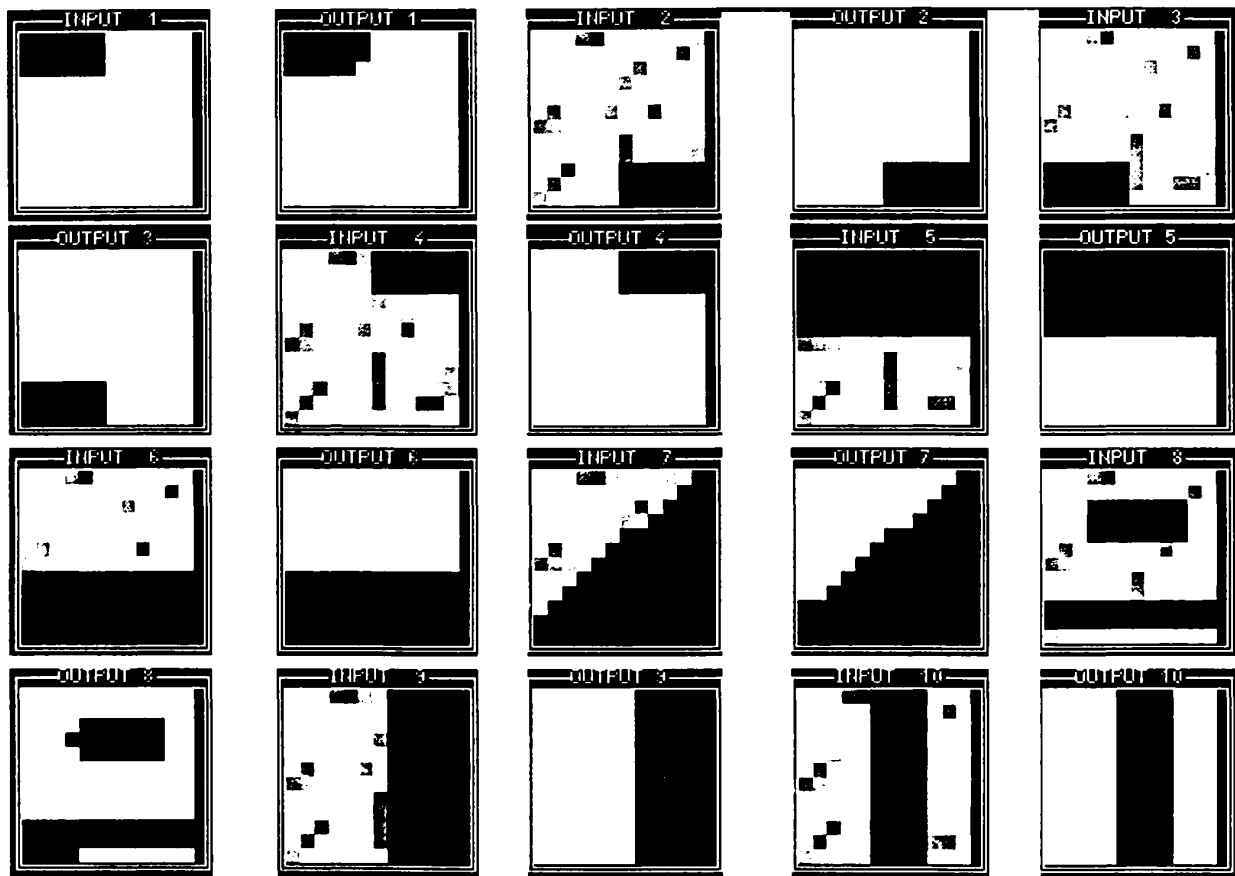


Fig. 6.9. Cele zece modele de intrare afectate de un zgomot gaussian cu medie nula și dispersie  $\sigma=0.4$  aplicate la intrarea unei MAB de dimensiune 12x12 și ieșirile acesteia, corespunzătoare primului strat

Fig.6.10 prezintă modelele de intrare, afectate de o eroare de 15% și modelele de ieșire după stabilizarea rețelei. Opt modele au fost corect apelate din memorie, unul cu o eroare de 1 pixel și unul cu o eroare de 3 pixeli.

Se poate constata că la o aceeași dispersie a zgomotului gaussian, respectiv la un același procentaj de eroare pentru zgomotul binar aplicat modelului de intrare, rețeaua de dimensiune 12x12 a avut performanțe net mai bune decât ale celei de dimensiune 10x10. În concluzie performanțele MAB se îmbunătățesc cu dimensiunea, în aceleași condiții de funcționare.

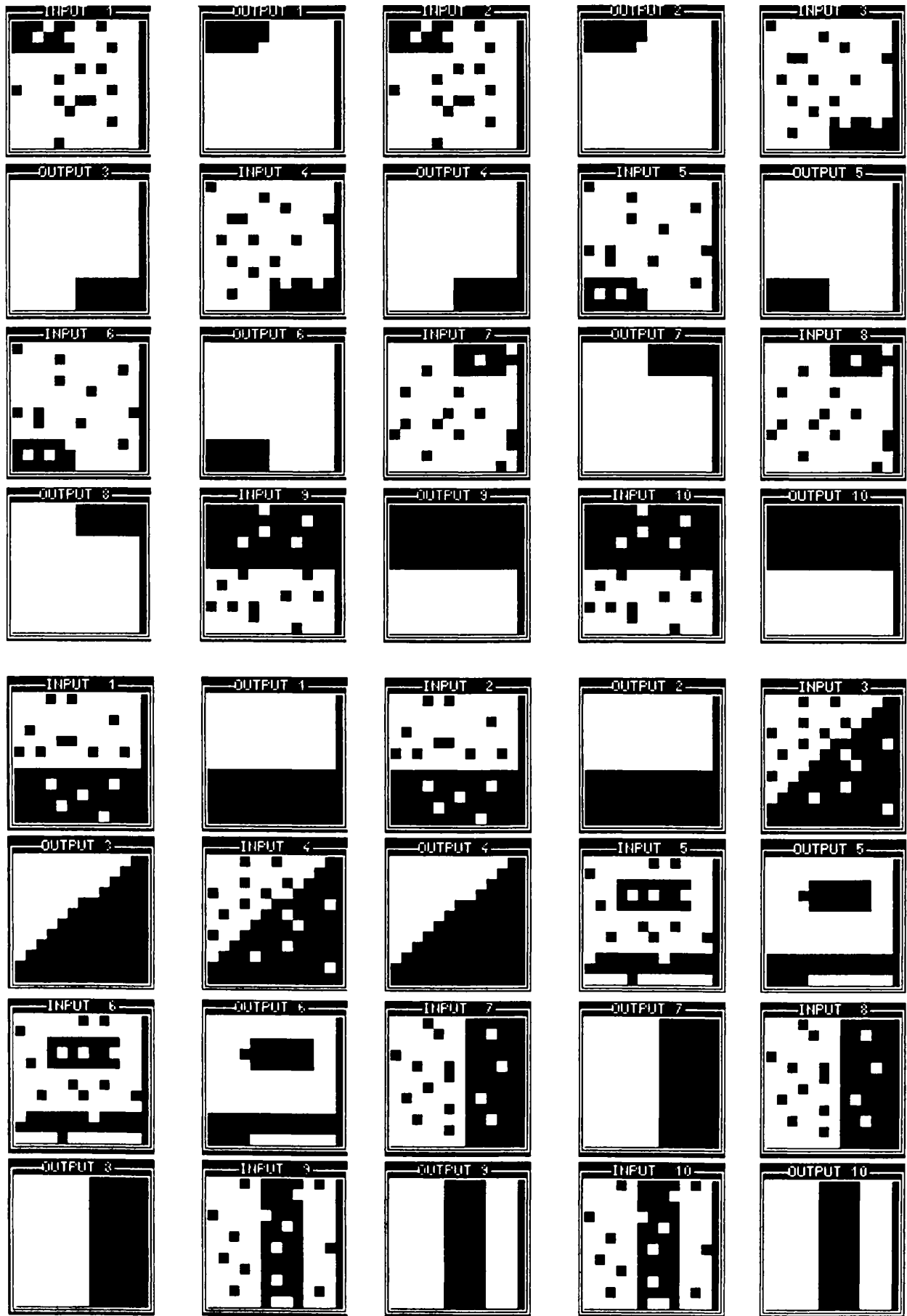


Fig. 6.10 Modelele de intrare, reprezentând modelele memorate, afectate cu o eroare binară de 15% și cele de ieșire după stabilizarea rețelei de dimensiune 12x12

## Exemplul 4

În urma experimentelor am observat că dacă introducem un factor de ponderare a regulii lui Hebb rezultatele obținute au fost sensibil mai bune. Aceasta demonstrează că regula lui Hebb este foarte sensibilă la reglaje și că, în cazul unei aplicații concrete, experiența programatorului are o influență majoră.

Cu o MAB de dimensiune 9x9 au fost memorate ideogramele chinezești din Fig. 6.11.

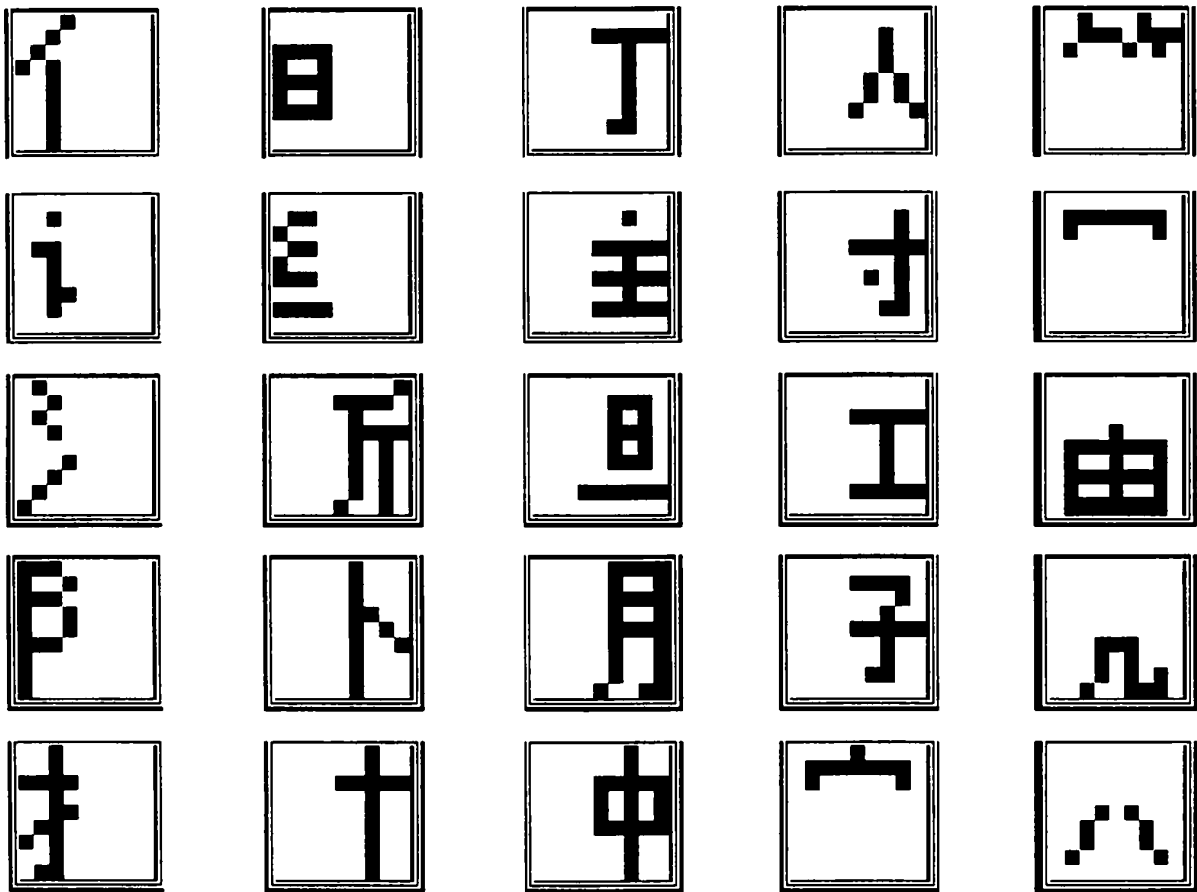


Fig. 6.11 Modelele ideale memorate reprezentând 25 de ideograme chinezești de dimensiune 9x9

Se poate observa că există submodele care impun și o distanță de regăsire  $b=1$ . Capacitatea de memorare ar fi deci  $C=1.3.3=9$ . Cum nu între toate modelele există restricția impusă asupra distanței de regăsire  $b=1$ , se poate constata practic, că în condițiile în care zgomotul este mic, memoria apelează corect toate cele 25 de modele.

Am ponderat regula lui Hebb cu un factor constant, subunitar:

- pentru autoreacție:

$$A_{ij,ij} = a \cdot \sum_{q=1}^P y_{ij}^q \cdot y_{ij}^q$$

- pentru interacțiunea cu celulele vecine:

$$A_{kl,ij} = b \cdot \sum_{q=1}^P y_{kl}^q \cdot y_{ij}^q$$

Tabelul 6.1 prezintă rezultate testelor efectuate cu  $a=0.5$  și  $b=0.1$ , cu o vecinătate de  $r=1$ . Modelele de intrare au fost afectate cu un zgomot gaussian de medie nulă și dispersie  $\sigma$  cuprinsă între 0.2 și 0.7.

În cazul în care zgomotul este binar memoria nu regăsește modelele ideale.

Fig.6.12 prezintă modelele de intrare afectate de un zgomot cu  $\sigma=0.3$ , care au fost corect apelate de memoria asociativă.

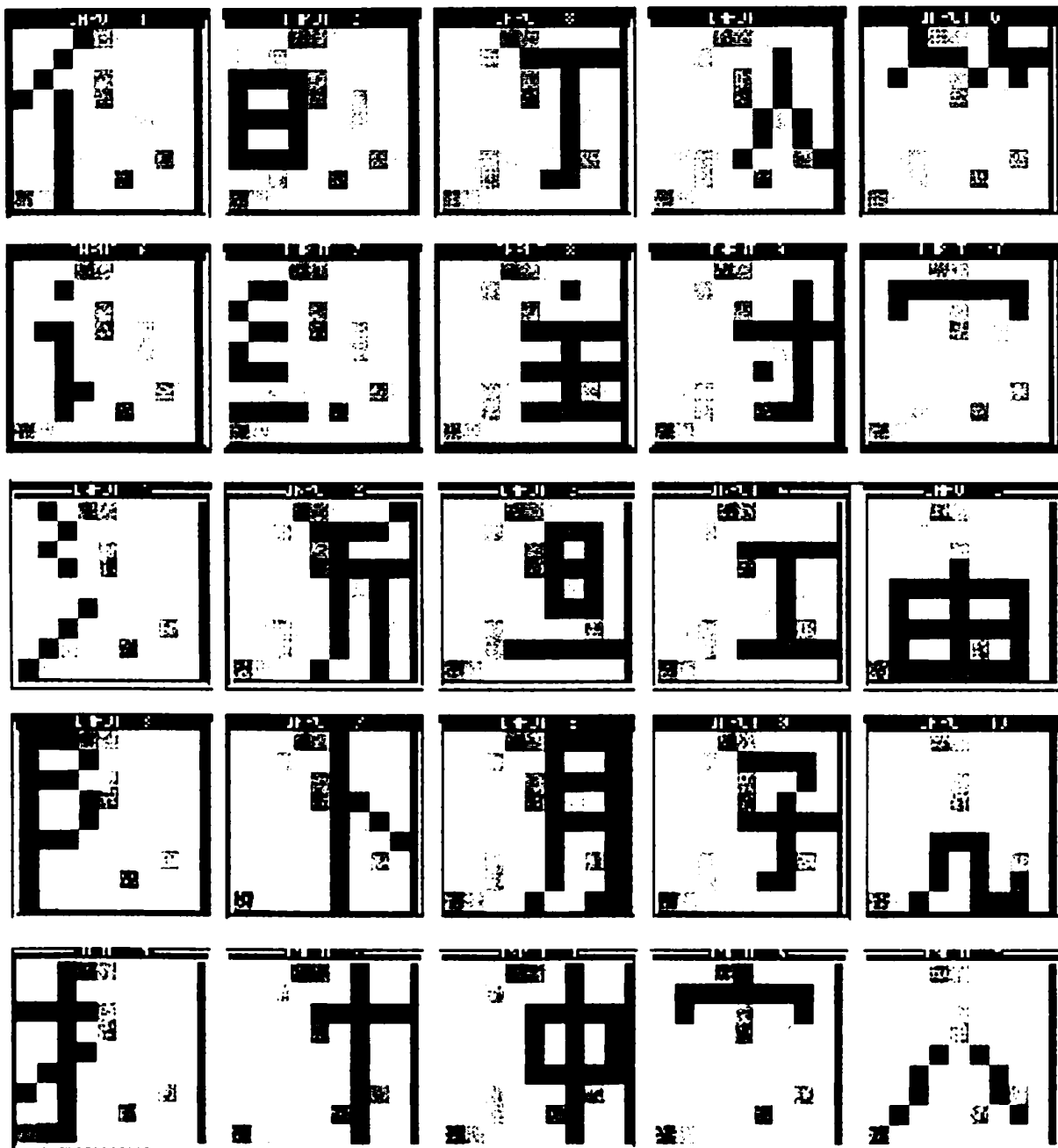


Fig. 6.12 Modele de intrare reprezentând cele 25 de ideograme chinezești afectate cu un zgomot gaussian de medie nulă și dispersie  $\sigma = 0.3$

**Tabelul 6.1**

Zgomot / vecinătate	Vecinătate $r=1$
gaussian $\sigma = 0.3$	25 modele corect apelate din memorie
gaussian $\sigma = 0.4$	18 modele corect apelate din memorie, unul cu 2 pixeli eroare, restul cu 1 pixel eroare
gaussian $\sigma = 0.5$	9 modele corect apelate din memorie, 8 cu o eroare de 1 pixel, restul cu 2-3 pixeli eroare
gaussian $\sigma = 0.6$	5 modele corect apelate din memorie, 14 cu o eroare de 1 pixel, restul cu o eroare de 2-3 pixeli
gaussian $\sigma = 0.7$	2 modele corect apelate, 7 cu 1 pixel eroare, restul cu 2 <sup>1</sup> / <sub>4</sub> pixeli eroare

Memoria a fost testată și pentru o vecinătate  $r=2$ , respectiv  $r=3$ , pentru toate exemplele anterior prezentate. Rezultatele obținute au fost în general mai bune, dar nu semnificativ astfel încât să justifice continuarea studiului și mărirea vecinătății în vederea unei aplicații concrete.

### 6.2.5 Observații

Memoria asociativă bidirecțională introduce o structură nouă cu o performanță bună în regăsirea modelului în condiții de zgomot, mai bună decât a modelului Vandewaale.

Numărul modelelor memorate crește cu dimensiunea rețelei și raza vecinătății. Deși regula lui Hebb este relativ simplă și ușor de utilizat, ea are limitări serioase: nu garantează că fiecare model va fi înmagazinat ca un punct de echilibru stabil, atunci când modelele de intrare nu îndeplinesc condițiile. Întrucât nu se pun condiții asupra traiectoriei, există posibilitatea ca, în cazul în care nu sunt îndeplinite condițiile de distanță dintre modele, memoria să eșueze în minime locale. Capacitatea de memorare este relativ mică. Pentru apelul corect al modelelor din memorie se impun restricții asupra modelelor de memorat .

Regula lui Hebb este foarte sensibilă la reglaje. În cazul unei aplicații concrete poate deveni utilă doar după experimente îndelungate.

Ar trebui îmbunătățită performanța memoriei, prin încorporarea unei informații suplimentare, adăugate regulii lui Hebb, astfel încât să se poată pune problema unei aplicații concrete.

## 6.3 Algoritm rapid de învățare a ponderilor

### 6.3.1 Prezentarea algoritmului și a bazelor sale teoretice

Algoritmul propus pentru obținerea unor matrici de interconexiune variabile în spațiu [95] este rapid, are o bună convergență și este mai puțin sensibil la variația parametrilor.

Fie o rețea neuronală celulară analogică, bidimensională, de dimensiune  $D=M \times N$ .

Fie regiunea de saturație RS, dată prin definiția 12.

Se determină o condiție suficientă ca vectorul de stare  $x$  să reprezinte un punct de echilibru stabil pentru matricile de ponderi date  $A$ ,  $B$  și  $I$ .

#### Teorema 9

Vectorul de stare  $x$  este un punct de echilibru dacă  $x \in RS$  și  $y = \text{sgn}(x)$  îndeplinește condiția  $\text{sgn}(x) = G \cdot y$ . Matricea  $G$  este diagonală, de dimensiune  $D \times D$ , și este obținută din matricea unitate fixând unele elemente de pe diagonală la zero [35]:

$$G_{i,i} = \begin{cases} 1 & \text{pentru } \dot{x} \neq 0 \\ 0 & \text{pentru } \dot{x} = 0 \end{cases} \quad i = 1, \dots, D \quad (6.23)$$

unde  $D = M \times N$  este dimensiunea rețelei

Dacă  $G$  este chiar matricea unitate, echilibrul este stabil.

În regiunea de saturație ecuația de stare a unei celule este:

$$\dot{x} = -U \cdot x + v \quad (6.24)$$

unde  $v$  este un vector constant,  $v = A \cdot y + B \cdot u + I$ ,

Aceasta înseamnă că toate valorile proprii sunt egale cu  $-1$ . Dacă există puncte de echilibru în afara regiunii de saturație RS, ele vor fi asimptotic stabile. Dacă valoarea funcției signum a stării este egală cu valoarea funcției signum a derivatei stării atunci soluția rămâne în interiorul lui S.

Se dorește memorarea a  $P$  modele bipolare. Se pune deci problema determinării matricilor de conexiune  $A$ ,  $B$  și  $I$  atunci când se cunosc ieșirile dorite (modelele de memorat).

Fie ecuația de stare a celulei  $C_{i,j}$  în regim staționar:

$$x_{i,j} = \sum_{kl \in N_r} A_{ij,kl} \cdot y_{kl} + c_{ij} \quad (6.25)$$

unde  $c_{ij}$  este un vector constant dat de relația:

$$c_{ij} = \sum_{kl \in N_r} B_{ij,kl} \cdot u_{kl} + I_{ij} \quad (6.26)$$



Cele două soluții posibile pentru starea staționară a unei celule sunt date de:

$$\text{a. } y_{ij} \geq 1, \text{ implică } x_{ij} > 1, \text{ deci } \sum_{kl \in N_r} A_{ij,kl} \cdot y_{kl} + c_{ij} > 1 \quad (6.27)$$

$$\text{b. } y_{ij} \leq -1, \text{ implică } x_{ij} < 1, \text{ deci } \sum_{kl \in N_r} A_{ij,kl} \cdot y_{kl} + c_{ij} < -1 \quad (6.28)$$

Inegalitățile (6.27), (6.28) se pot sintetiza în relația:

$$\sum_{k,l \in N_r(ij)} A_{ij,kl} \cdot y_{kl} \cdot y_{ij} + c_{ij} \cdot y_{ij} > 1 \quad (6.29)$$

Punctele de echilibru trebuie localizate suficient de departe de frontiera regiunii liniare pentru a avea o stabilitate robustă. Fie E o constantă pozitivă care caracterizează valoarea punctului de echilibru al unei celule.

$$A_{ij}^T \cdot Y_{ij} \cdot y_{ij} + c_{ij} \cdot y_{ij} > E > 1 \quad (6.30)$$

unde:

- $A_{ij}^T$  este transpusa matricii de conexiune de reacție;
- $Y_{ij}$  este submodelul ieșirilor în jurul celulei  $C_{ij}$ ;

Intrarea independentă nu este utilizată în acest caz. Înlocuind deci vectorul constant  $c_{ij}$  se obține:

$$A_{ij}^T \cdot Y_{ij} \cdot y_{ij} + I_{ij} \cdot y_{ij} > E \quad (6.31)$$

Notând produsul  $Y_{ij} \cdot y_{ij} = P_{ij}$  ecuația (6.31) devine:

$$A_{ij}^T \cdot P_{ij} + I_{ij} \cdot y_{ij} > E \quad (6.32)$$

Singura restricție este ca  $E > 1$ . Inegalitatea (6.32) trebuie să fie adevărată pentru fiecare model  $q$ , dintre cele  $P$  de memorat, și toate celulele rețelei:

$$A_{ij}^T \cdot P_{ij}^q + I_{ij} \cdot y_{ij}^q = E_q > E \text{ ptr. } q = 1, 2, \dots, P \quad (6.33)$$

$$i = 1, 2, \dots, M; j = 1, 2, \dots, N$$

Problema care se pune este determinarea matricii de conexiune  $A_{ij}$  pentru toate celulele rețelei astfel încât să fie îndeplinită condiția (6.33). Se poate alege o metodă arbitrară pentru determinarea lui  $A$ , ca de exemplu regula lui Hebb cunoscută,  $Y_{ij} \cdot y_{ij} = P_{ij}$  ponderată cu un coeficient de câștig  $w_{ij}$  pentru fiecare model  $q$ :

$$A_{ij} = \sum_{q=1}^P w_{ij}^q \cdot P_{ij}^q \quad (6.34)$$

Matricea  $A_{ij}$  poate fi procesată iterativ, pornind de la o valoare inițială arbitrară, ca de exemplu zero:

$$A_{ij}[n+1] = A_{ij}[n] + \sum_{n=1}^p \delta \cdot w_{ij}^q \cdot P_{ij}^q \quad (6.35)$$

Coefficientul de ponderare  $\delta \cdot w_{ij}^q$  poate fi o constantă pozitivă de valoare mică, sau o funcție de  $E_q$ . Fie funcția liniară dată de Fig.6.13a., definită în jurul punctului de echilibru, prin relația:

$$\delta \cdot w_{ij}^q = \lambda \cdot (E - E_q) \text{ cu } \lambda > 0 \quad (6.36)$$

unde  $\lambda$  este o constantă pozitivă, de valoare arbitrară;

O condiție mai puțin restrictivă, în care punctul de echilibru ia valori într-un interval  $(E_L, E_R)$ , este prezentată în Fig.6.13 b și este dată de:

$$E_L < E_q < E_R \quad (6.37)$$

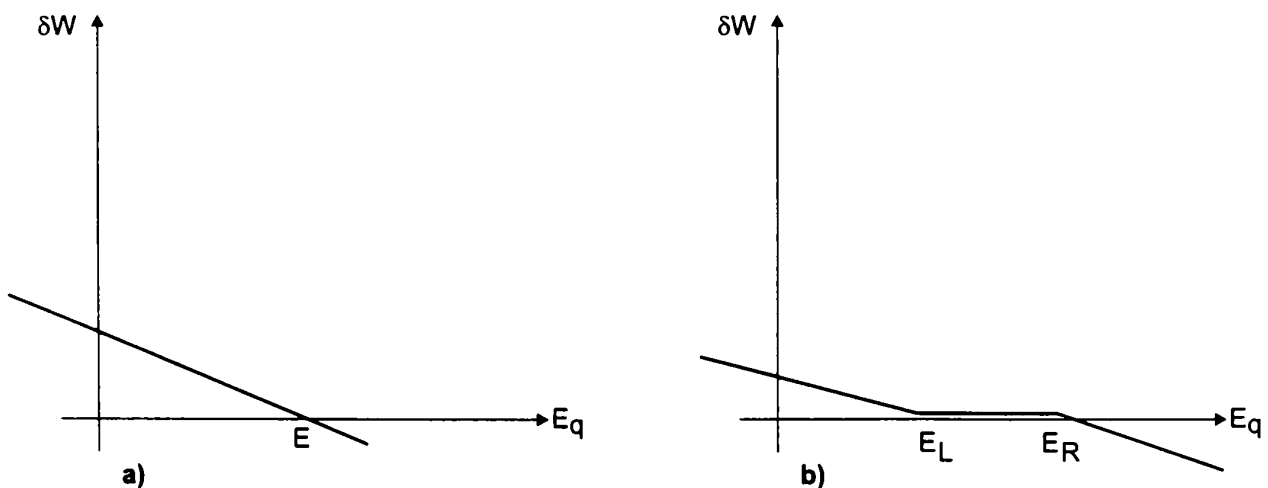


Fig. 6.13 Variația coeficientului de ponderare  $\delta w$  în funcție de  $E$ .

**Procedura de sinteză** este descrisă de următorul algoritm:

1. Pentru toate celulele rețelei  $C_{ij}$   $i=1,2,\dots,M$  și  $j=1,2,\dots,N$  se inițializează matricea de conexiune  $A_{ij}$  și matricea de polarizare  $I_{ij}$  la zero.
2. Pentru fiecare model  $q$ , dintre cele  $P$  de memorat, se procesează submodelul  $P_{ij}^q$ .
3. Se inițializează  $E_q$  la zero.
4. Pentru fiecare model  $q=1,2,\dots,P$ , pentru fiecare celulă  $C_{ij}$  se calculează  $E_q$  cu relația (6.33).
5. Se calculează coeficientului de ponderare  $\delta \cdot w_{ij}$  cu expresia (6.35) sau cu (6.37)

Dacă  $E_q = E$  :  $\delta \cdot w_{ij} = 0$

Dacă  $E_q \neq E$  :  $\delta \cdot w_{ij}^q = \lambda \cdot (E - E_q)$  cu  $\lambda > 0$

6. Se determină  $A_{ij}[n+1]$  cu relația (6.36) și  $I_{ij}[n+1]$  cu relația:

$$I_{ij}[n+1] = I_{ij}[n] + \delta \cdot w_{ij} \cdot y_{ij} \quad (6.38)$$

$A_{ij}[n+1]$  și  $I_{ij}[n+1]$  reprezintă matricile de interconexiune căutate pentru celula  $C_{ij}$ .

### 6.3.2. Rezultatele simulării

S-a experimentat memorarea celor 25 de ideograme chinezești din Fig. 6.11.

Programul pentru determinarea matricilor de interconexiune a fost implementat în Borland C++.

S-au utilizat două tipuri de semnale aleatoare nedorite, suprapuse peste semnalul util și necorelate cu acesta: un zgomot gaussian cu medie nulă și dispersie  $s \in [0.0, 0.7]$  și unul digital, cu un procentaj de inversare a valorii pixelilor între 5 % și 10 %. Modelele afectate de zgomot gaussian cu  $\sigma=0.7$  sunt prezentate în Fig.6.14.

Capabilitatea de corecție a erorii a fost testată ca o funcție de  $[E_L, E_R]$ . Rezultatele experimentelor sunt sintetizate în Tabelul 6.2 pentru  $E \in [1.2, 2]$  și în Tabelul 6.3 pentru  $E \in [3, 10]$ .

**Tabelul 6.2**

Zgomot / vecinătate	Vecinătate $r=1$	Vecinătate $r=2$
gaussian $\sigma=0,2$	25 modele corect apelate din memorie	25 modele corect apelate din memorie
gaussian $\sigma=0,5$	8 modele corect apelate din memorie	17 modele corect apelate din memorie
gaussian $\sigma=0,7$	toate modelele afectate cu o eroare de 1-8 pixeli	11 modele corect apelate din memorie
binar 5% pixeli inverșați	8 modele corect apelate din memorie	8 modele corect apelate din memorie
binar 10% pixeli inverșați	toate modelele afectate cu o eroare de 1-8 pixeli	5 modele corect apelate din memorie

Cu cât  $E$  este mai mare cu atât capabilitatea de corecție este mai bună și tranziția mai rapidă. Datorită metodei de învățare, autoreacția  $A_{ij,ij}$  este întotdeauna mai mare decât ponderile conexiunilor cu celulele învecinate. În cazul unei valori  $E$  mai mari ponderile au valori mai mari, celulele din jur nu pot compensa acțiunea celulei centrale, eroarea nu poate fi micșorată.

Dacă se utilizează un interval mai mare pentru  $E$  rețeaua poate converge spre un alt model.

Pe măsură ce vecinătatea crește, performanțele memoriei asociative neuronale sunt mai bune.

Tabelul 6.3

Zgomot / vecinătate	Vecinătate r=1	Vecinătate r=2	Vecinătate r=3
gaussian $\sigma=0,2$	25 modele corect apelate din memorie	25 modele corect apelate din memorie	25 modele corect apelate din memorie
gaussian $\sigma=0,5$	17 modele corect apelate din memorie, restul cu o eroare de 1 pixel	25 modele corect apelate din memorie	25 modele corect apelate din memorie
gaussian $\sigma=0,7$	toate modelele afectate cu o eroare de 1-8 pixeli	13 modele corect apelate din memorie	25 modele corect apelate din memorie
binar 5% pixeli inversați	15 modele corect apelate din memorie	7 modele corect apelate din memorie	15 modele corect apelate din memorie
binar 10% pixeli inversați	toate modelele afectate cu o eroare de 1-8 pixeli	6 modele corect apelate din memorie	11 modele corect apelate din memorie

## 6.4 Algoritm rapid cu un număr redus de conexiuni

### 6.4.1. Considerații teoretice

Memoria asociativă propusă în acest paragraf este generată de un algoritm rapid de procesare a interconexiunilor, care are performanțe bune în condiții de zgomot și toate avantajele unui număr redus de conexiuni [14]. Metoda propusă determină convergența rețelei înspre puncte fixe, de echilibru stabil.

Considerând matricea de control B zero, condiția de convergență se reduce la [35]:

$$\sum_{k,l \in Nr} A_{ij;kl} \cdot y_{kl} \cdot y_{ij} + I_{ij} \cdot y_{ij} > E > 1 \quad (6.39)$$

unde E este o constantă pozitivă mai mare decât 1 (cu cât E este mai mare cu atât stabilitatea în regiunea de saturație este mai robustă);

Am ales următoarele valori inițiale pentru ponderile interconexiunilor de reacție:

- pentru elementul central al vecinătății:

$$A_{i,j;ij}[0] = a \cdot \sum_{q=1}^P (y_{ij}^q - y_{ij}^q)(y_{ij}^q - y_{ij}^q) \quad (6.40)$$

- și pentru celelalte elemente ale vecinătății:

$$A_{kl;ij}[0] = b \cdot \sum_{q=1}^P (y_{kl}^q - y_{kl}^q)(y_{ij}^q - y_{ij}^q) \quad (6.41)$$

unde:

- a și b sunt constante pozitive;

- $\bar{y}_{ij}^q (\bar{y}_{kl}^q)$  este valoarea medie a pixelului ij (respectiv pixelului kl) pentru modelul q de memorat;

Ponderile de interconexiune de reacție pentru fiecare celulă  $C_{ij}$  se calculează iterativ cu:

$$A_{ij,kl}[n+1] = A_{ij,kl}[n] + \sum_{n=1}^p d_{ij}^q \cdot A_{ij,kl}[0] \quad (6.42)$$

unde:

- P este numărul modelelor de memorat;
- $d_{ij}^q$  este o constantă pozitivă mică, calculată pentru fiecare model q și fiecare celulă  $C_{ij}$  cu formula:

$$d_{ij}^q = \lambda \cdot (E - E_{ij}^q) \quad (6.43)$$

unde :

- $\lambda$  este o constantă mică pozitivă;
- $E_{ij}^q$  este o funcție de energie calculată cu relația:

$$\sum_{kl \in N_r} A_{ij,kl} \cdot y_{kl}^q \cdot y_{ij}^q + I_{ij} \cdot y_{ij}^q = E_{ij}^q \quad (6.44)$$

Curentul de polarizare este determinat cu:

$$I_{ij}[n+1] = I_{ij}[n] + \sum_{n=1}^p d_{ij}^q \cdot y_{ij}^q \quad (6.45)$$

Fie P modele de memorat (bipolare cu valoarea de -1 pentru alb și +1 pentru negru), de dimensiune  $M \times N$ .

**Procedura de memorare** a lor într-o RNC asociativă, de dimensiune  $M \times N$ , este următoarea:

1. Pentru fiecare celulă  $C_{ij}$ ,  $i=1,2, \dots, M$  și  $j=1,2, \dots, N$  interconexiunile de reacție sunt inițializate cu (6.40) și (6.41), I și  $E_{ij}^q$  la zero,  $\lambda$  la o valoare arbitrară pozitivă mică (suficient de mică pentru a asigura o variație gradată a ponderilor de interconexiune, de ex. 0.01) și E la o constantă.
2. Pentru fiecare model q ( $q=1,2,3, \dots, P$ ) se calculează  $E_{ij}^q$  și testează condiția (6.39).

Dacă  $E_{ij}^q < E$  atunci {

Dacă DA  $E_{ij}^q > E$  atunci  $d_{ij}^q = 0$

Dacă NU  $d_{ij}^q$  este determinată cu relația (6.43).

Interconexiunile de reacție sunt determinate cu (6.42) și cele ale curentului de polarizare cu (6.45). }

Se salvează matricile  $A_{ij}$  și  $I_{ij}$  și se trece la o nouă celulă.

### 6.4.2 Rezultatele simulării

Modelele ideale de memorat au fost cele 25 ideograme chinezești de dimensiune 9x9, din Fig.6.11, aceleași raportate și în [60], [95]. Am testat performanțele rețelei aplicând la intrarea ei modele afectate de un zgomot gaussian cu medie nulă și dispersie  $\sigma$  în intervalul [0,0.7] .

În Fig.6.14 sunt prezentate modelele de intrare cu un zgomot gaussian  $\sigma=0.7$ .

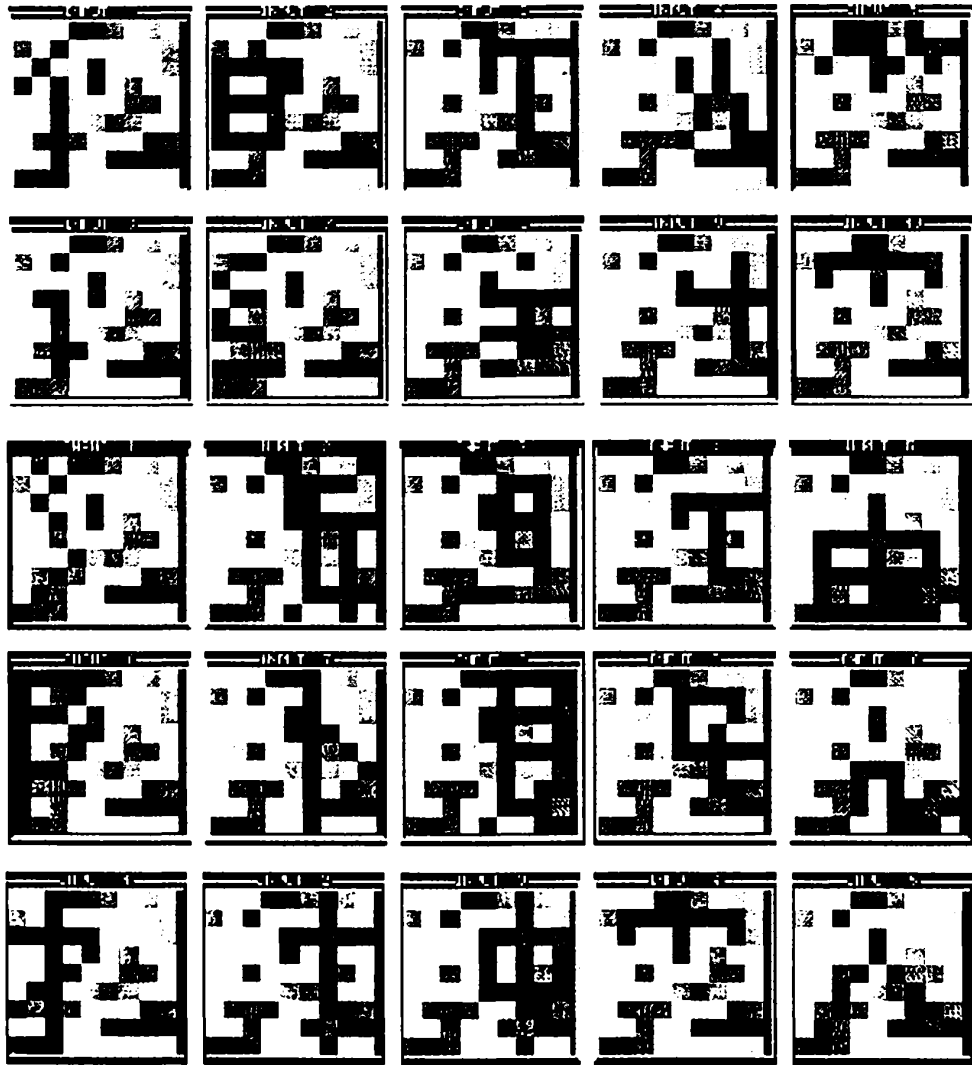
Am utilizat  $a=0.5$  sau mai mare și  $b \in (0.01,0.5)$ . Dacă  $b$  a fost mai mare celula centrală nu a putut compensa efectul celulelor învecinate.

Tabelul 6.4 prezintă câteva dintre rezultatele simulării (pentru  $a=0.5, b=0.1, E=5$  și  $E=10$ ), pentru o vecinătate  $r=1$ , interconexiuni de reacție date de relațiile (6.40) și (6.41), un curent de polarizare nul și pentru două valori diferite ale constantei  $E$  .

**Tabelul 6.4**

Zgomot / valoare E	E = 5	E = 10
gaussian $\sigma=0,3$	25 modele corect apelate din memorie	25 modele corect apelate din memorie
gaussian $\sigma=0,4$	22 modele corect apelate din memorie, restul cu o eroare de 1 pixel	25 modele corect apelate din memorie
gaussian $\sigma=0,5$	14 modelele corect apelate din memorie, 2 modele cu 2 pixeli eroare, restul cu o eroare de 1 pixel	6 modele corect apelate din memorie, 12 cu o eroare de 1 pixel, restul cu o eroare de 2 pixeli
gaussian $\sigma=0,6$	4 modele corect apelate din memorie, 9 cu 1 pixel eroare, restul cu o eroare de 2 pixeli	6 modele corect apelate din memorie, 6 cu o eroare de 1 pixel, restul cu o eroare de 2 pixeli

Datorită inițializării particulare date de relațiile (6.40) și (6.41) a interconexiunilor de reacție, procedura generează o matrice de reacție rară, care constituie un avantaj în comparație cu simpla regulă a lui Hebb, în cazul implementării VLSI.



**Fig. 6.14. Cele 25 de modele memorate afectate de un zgomot gaussian cu medie nulă și dispersie  $\sigma=0.7$**

Tabelul 6.5 prezintă câteva dintre rezultatele simulării pentru cazul în care curentul de polarizare este diferit de zero. Se observă că pentru o aceeași valoare  $E = 5$ , rezultatele au fost mai bune comparativ cu situația în care curentul de polarizare a fost nul.

Tabelul 6.6 prezintă rezultatele simulării pentru o valoare a lui  $E$  mai mare și un curent de polarizare diferit de zero (pentru  $a=0.5$ ,  $b=0.1$ ,  $E=10$ ).

Cu cât  $E$  este mai mare, cu atât capacitatea de corecție este mai bună și tranziția mai rapidă. Valoarea  $E$  caracterizează robustețea rețelei. Cu cât  $E$  este mai mare, stabilizarea rețelei are loc mai departe de punctul de echilibru dorit, în bazinul de atracție al acestuia. O valoare  $E$  foarte mare generează ponderi de valori mari, ceea ce nu este de dorit, datorită limitărilor de implementare.

În toate experimentele rezultatele procesării au fost mai bune pentru vecinătăți mai mari ( $r=2$  și  $r=3$ ).

MA poate apela și suma a două modele memorate (după cum era de așteptat [60]). În Fig.6.15 sunt prezentate câteva combinații posibile afectate cu un zgomot gaussian, cu medie nulă și dispersie  $\sigma=0.7$ . Toate au condus la modelele ideale.

Tabelul 6.5

Zgomot / vecinătate	Vecinătate r=1	Vecinătate r=2	Vecinătate r=3
gaussian $\sigma=0,4$	25 modele corect apelate din memorie	25 modele corect apelate din memorie	25 modele corect apelate din memorie
gaussian $\sigma=0,5$	21 modele corect apelate din memorie, restul cu o eroare de 1 pixel	25 modele corect apelate din memorie	25 modele corect apelate din memorie
gaussian $\sigma=0,6$	11 modele corect apelate din memorie, 12 cu o eroare de 1 pixel și 2 cu o eroare de 2 pixeli	16 modele corect apelate din memorie, restul cu o eroare de 1 pixel	21 modele corect apelate din memorie, 4 cu o eroare de 1 pixel

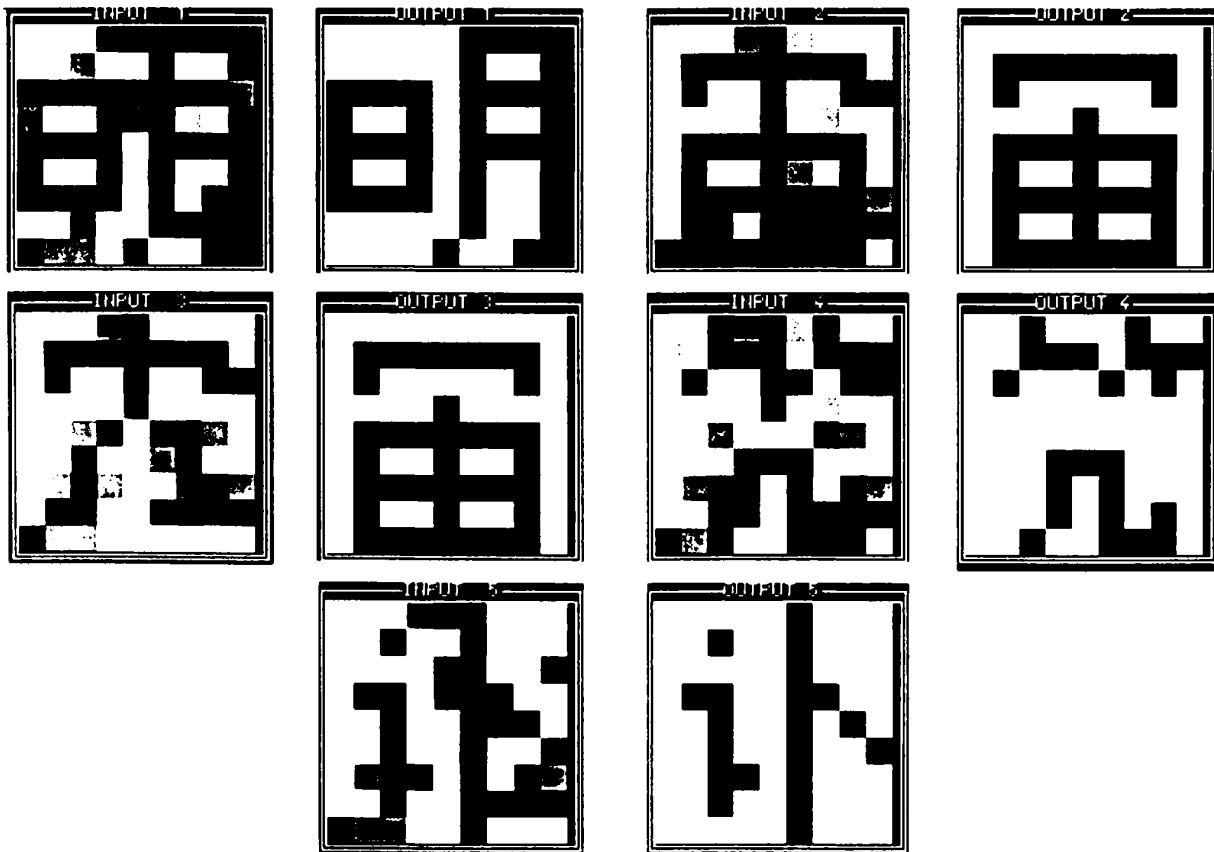


Fig. 6.15 Combinații ale modelelor memorate, afectate de un zgomot gaussian cu medie nulă și dispersie  $\sigma=0.7$



Tabelul 6.6

Zgomot / vecinătate	Vecinătate r=1	Vecinătate r=2	Vecinătate r=3
gaussian $\sigma=0,4$	25 modele corect apelate din memorie	25 modele corect apelate din memorie	25 modele corect apelate din memorie
gaussian $\sigma=0,5$	21 modele corect apelate din memorie, restul cu o eroare de 1 pixel	23 modele corect apelate din memorie, restul cu o eroare de 1 pixel	25 modele corect apelate din memorie
gaussian $\sigma=0,6$	17 modele corect apelate din memorie, restul cu o eroare de 1 pixel	20 modele corect apelate din memorie, restul cu o eroare de 1 pixel	24 modele corect apelate din memorie, 1 cu o eroare de 1 pixel
gaussian $\sigma=0,7$	5 modele corect apelate din memorie, 10 cu o eroare de 1 pixel, 5 cu 2 pixeli eroare, restul cu 3,4 pixeli eroare	12 modele corect apelate din memorie, 10 cu o eroare de 1 pixel, 3 cu 2 pixeli eroare	23 modele corect apelate din memorie, 2 cu o eroare de 1 pixel
binar 0,8% (1 pixel inversat)	2 modele corect apelate din memorie	23 de modele corect apelate din memorie	25 modele corect apelate din memorie
binar 1,6% (2 pixeli inversați)	—	6 modele corect apelate din memorie, 11 cu o eroare de 1 pixel, 8 cu 2 pixeli eroare	24 modele corect apelate din memorie, 1 cu 1 pixel eroare
binar 2,5% (3 pixeli inversați)	—	5 modele corect apelate din memorie, 5 cu 2 pixeli eroare, 15 cu 3 pixeli eroare	14 modele corect apelate din memorie, 15 cu 1 pixel eroare

## 6.4.3 Aplicație. Ghidarea automată a unui vehicul

### 6.4.3.1 Considerații teoretice

O aplicație posibilă a memoriei asociative neuronale celulare este ghidarea unui vehicul automat de-a lungul unei traiectorii marcate pe o podea [15].

Schema bloc a unității de control a vehiculului automat este prezentată în Fig.6.14. Unitatea analogică de control realizată cu RNC constă din două straturi: primul strat este o memorie asociativă care realizează eliminarea zgomotului din marcajele detectate și recunoașterea de model, iar cel de-al doilea strat este un detector de componente conectate care interpretează modelul.

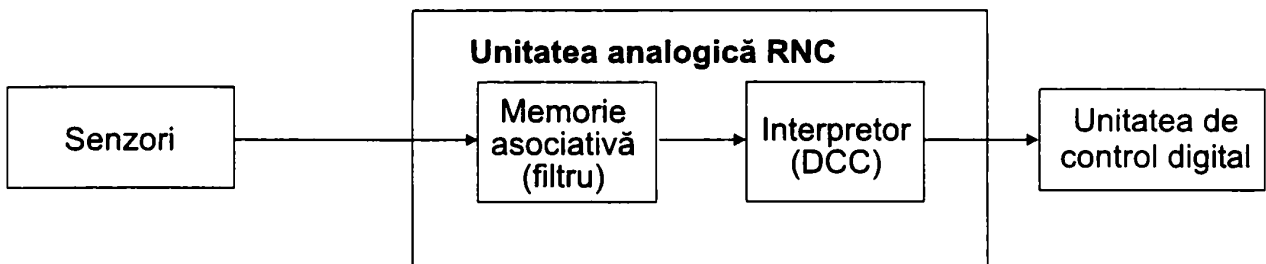


Fig. 6.16 Schema bloc a unității de control a vehiculului automat

Matricea de interconexiuni a memoriei asociative a fost obținută conform algoritmului anterior prezentat (§ 6.4).

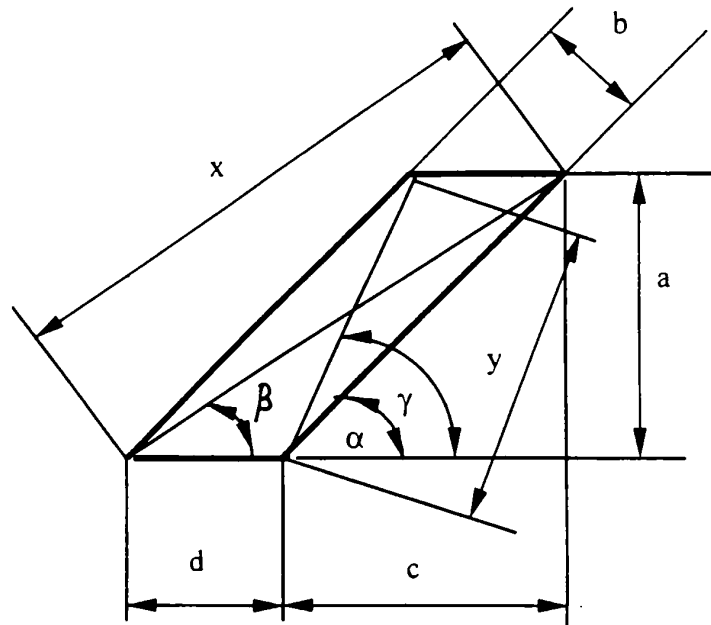
Detectorul de componente conectate (DCC) [113] are următoarele matrici de interconexiune:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{I} = 0 \quad (6.46)$$

Modele proiectate pentru diferitele direcții de deplasare sau acțiuni ce trebuie îndeplinite (ca de exemplu: oprire, ridicare platformă) trebuie să satisfacă următoarele cerințe:

- direcția de deplasare, sau acțiunea vehiculului să poată fi ușor determinată dintr-un model particular;
- modelul să contrasteze cu fondul podelei și să fie robust;

Modelul elementar ales este reprezentat în Fig.6.17. Trei parametri definesc fără ambiguitate modelul. Ei sunt  $\alpha$ ,  $\beta$  și  $\gamma$ . Parametrul  $\alpha$ , poate fi setat în concordanță cu câmpul vizual al detectorului. Proiecția orizontală a modelului este determinată de lungimea diagonalei lungi și a celei scurte, atunci când modelul este rotit în câmpul vizual.



**Fig. 6.17 Modelul elementar de control**

Direcția spre înainte este corespunzătoare situației în care proiecțiile celor două diagonale sunt egale. Vehiculul se deplasează spre dreapta când proiecția orizontală este mai scurtă și spre stânga când aceasta este mai lungă, comparativ cu direcția de rotație 0. Lungimea proiecției este:

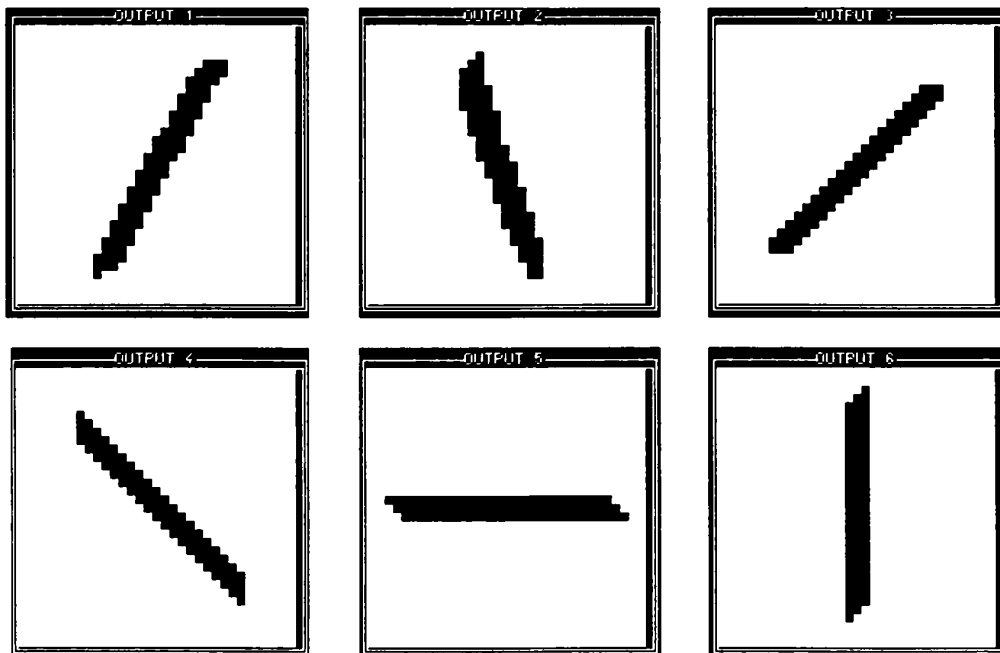
$$h(\varphi) = \max \{x \cdot \sin(\varphi + \beta), y \cdot \sin(\varphi + \gamma)\} \quad (6.47)$$

unde  $\varphi$  este rotația corespunzătoare direcției de urmat și ia valori în intervalul  $[-\alpha, 90^\circ - \beta]$ . Valoarea parametrului  $-\alpha$  poate fi procesată cu ajutorul celorlalți parametri aleși. Se poate demonstra [34] că din modelul elementar și proiecția sa orizontală poate fi determinată inversa funcției  $h(\varphi)$  pentru a se afla orientarea dorită, deci direcția de deplasare.

Corespunzător cu principiile anterior enunțate, cât și din considerații practice [34], parametrii aleși ai modelelor de control sunt următorii:

$$a = 5.25\text{cm} \quad \beta = 41,88^\circ \quad \gamma = 57,11^\circ$$

Modelele de control alese, de dimensiune 33x 33 sunt ilustrate în Fig.6.18. Proiecția orizontală a modelelor reprezintă direcția dorită și ea este determinată cu detectorul de componente conectate.



**Fig. 6.18 Modelele memorate**

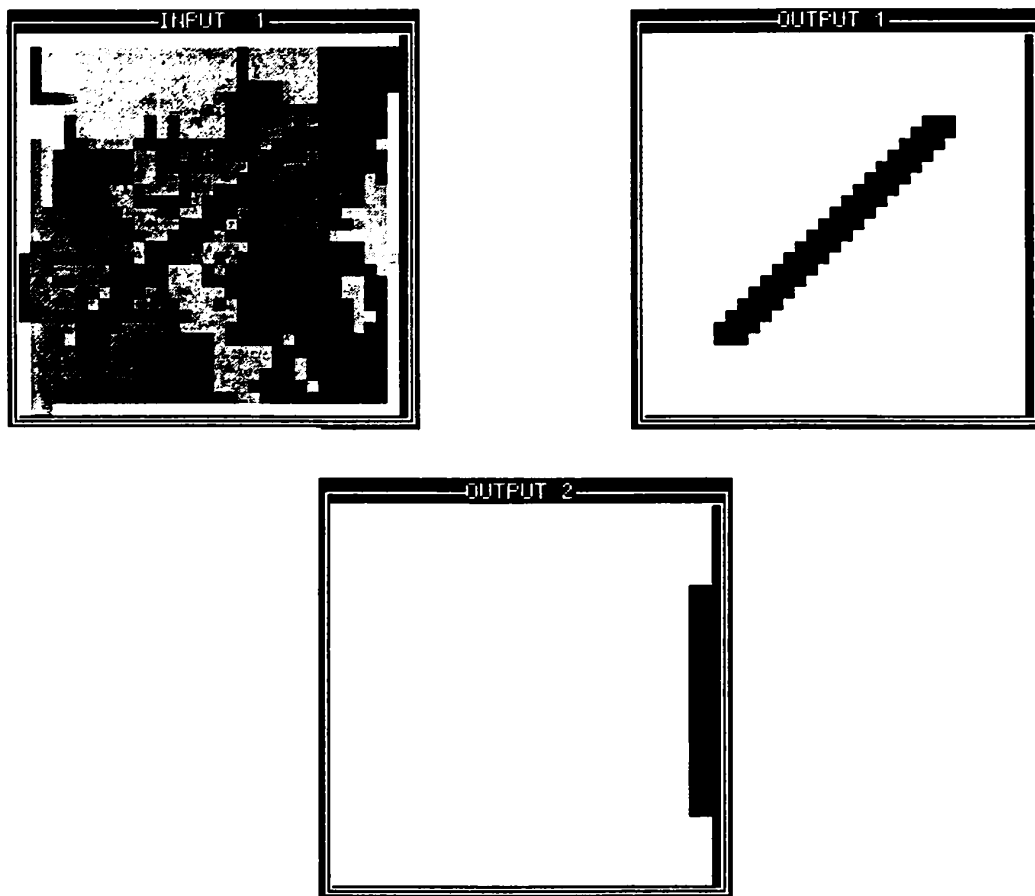
#### 6.4.3.2 Rezultatele simulării

Am testat unitatea de control RNC în condițiile funcționării cu modele de intrare afectate de un zgomot gaussian cu medie nulă și dispersie  $\sigma=0.6$ . Modelele perturbate prezentate în Fig.6.19 au fost aplicate ca stare a primului strat a unității de control RNC, simulată prin software-ul de RNC variante în spațiu [114]. Matricile de interconexiune au fost determinate în Borland C++. După ce rețeaua s-a stabilizat, la ieșirea memoriei asociative a fost obținut unul dintre modelele ideale memorate, iar la ieșirea stratului al doilea proiecția orizontală a acestuia.

Am testat unitatea de control și într-o situație reală. Câteva modele au fost scanate cu o video cameră și digitizate (cu frame-graber-ul). Fig. 6.20 prezintă un astfel de exemplu.



**Fig. 6. 19 Modelele de intrare afectate cu un zgomot gaussian cu medie nulă și dispersie  $\sigma=0,6$**



**Fig. 6.20** Intrarea digitizată, ieșirea primului și a celui de-al doilea strat ale unității de control cu RNC

#### 6.4.3.3 Concluzii

Simulările au demonstrat că modelele de control au fost corect apelate din memorie, eliminându-se zgomotul și generând la ieșire proiecția orizontală. Din modelul elementar și proiecția orizontală poate fi determinată rotația vehiculului. Utilizarea memoriei asociative RNC în unitatea de control a unui autovehicul automat este o soluție viabilă într-o aplicație practică pentru eliminarea zgomotului și recunoașterea de model, procesarea având loc într-o manieră similară celei din creierul uman.

RNC, cu posibilitatea utilizării chipului VLSI, oferă o soluție în timp real, flexibilă la schimbarea traseului și este mai ieftină decât metodele convenționale. Abordarea prin RNC are și avantajul toleranței erorilor, a instalării ușoare și a unui control local.

Datorită potențialului RNC eventuale îmbunătățiri pot fi aduse fără dificultăți. Singura limită este dată de capacitatea de memorare.

#### 6.4.4. Observații

Algoritmul prezentat este simplu și generează o matrice de reacție cu un număr redus de conexiuni diferite de zero. Memoria propusă este eficientă și în cazul unor matrici de control și de reacție nule, și a unei vecinătăți utilizate de  $r=1$ , în cazul în care zgomotul este redus ( $\sigma=0.4$ ). Modelul obținut, având un număr redus de interconexiuni de ordinul întâi poate fi ușor implementat în tehnologie VLSI.

Algoritmul este practic instantaneu pe un calculator PC AT 90 MHz PENTIUM, mult mai rapid decât oricare dintre cei raportați în literatură. El garantează că fiecare model dorit poate fi înmagazinat ca un punct de echilibru stabil. Sistemul poate converge și în modele nedorite, întrucât nu există nici o constrângere asupra traiectoriei.

## 6.5. Concluzii

În fiecare paragraf al acestui capitol s-au făcut observații pe baza studiului teoretic și a rezultatelor experimentale.

Tabelul 6.6 sintetizează, comparativ, performanțele memoriilor asociative ce utilizează regula lui Hebb, pe baza unor criterii general acceptate în domeniu.

Modelul Vandewaale [96] este, cronologic, prima memorie asociativă, cu informația inclusă în matricile de interconexiune de control. Ea are restricții asupra modelelor de memorat, și practic, o capacitate de memorare și capabilitate de corectare a erorilor redusă. Importanța acestei MA este doar istorică.

MAB propusă în [11] introduce, pentru prima oară, informația în matricile de reacție. Comparativ cu modelul Vandewaale performanțele MAB sunt superioare, deși, și aici asupra modelelor de memorat trebuie impuse restricții. În cazul în care modelele memorate (5 modele pentru o RNC de dimensiune  $10 \times 10$ ) respectă condițiile impuse de proiectare, MAB corectează un zgomot gaussian cu medie nulă și dispersie  $\sigma=0.9$  și un zgomot binar de 25 %. MAB asigură o corecție mai bună a zgomotului gaussian în comparație cu algoritmul rapid propus în [95] chiar în condițiile care modelele memorate nu respectă restricțiile de proiectare.

Memoria asociativă proiectată de Szolgay & al., în [95], utilizează un algoritm rapid, aditiv pentru procesarea interconexiunilor, are o foarte bună capabilitate de corecție a erorilor gaussiene cu  $\sigma=0.7$  și medie nulă, dar în condițiile unei vecinătăți cu  $r=3$ , ceea ce o face dificil de implementat VLSI.

Memoria asociativă propusă în [14] este proiectată printr-un algoritm simplu și rapid de procesare a interconexiunilor, care generează o matrice rară de reacție, are performanțe bune în condiții de zgomot și toate avantajele unui număr redus de conexiuni. Comparativ cu celelalte metode ce utilizează regula lui Hebb algoritmul propus asigură cea mai bună capabilitate de corecție a erorilor pentru o vecinătate de  $r=1$ , respectiv eliminarea unui zgomot gaussian cu dispersie  $\sigma=0.4$  pentru cazul memorării a 25 de modele oarecare într-o RNC de dimensiune  $9 \times 9$ . În cazul în care vecinătatea este  $r=3$  performanțele sunt similare algoritmului rapid [95].

În paragraful 6.4.3 este prezentată o aplicație concretă a memoriei propuse în unitatea de control a unui vehicul automat care se deplasează de-a lungul unei traiectorii marcate pe o podea. A fost testată atât în condițiile teoretice ale funcționării cu modele de intrare afectate de un zgomot gaussian cu medie nulă și dispersie  $\sigma=0.6$  cât și într-o situație reală. Memoria asociativă a realizat eliminarea zgomotului și apelul modelului ideal în ambele

Tipul rețelei și al învățării	RNCD modelul Vandewaale [96]	RNCA MAB [11]	CNN algoritm rapid [95]	RNCA algoritm rapid cu un număr reduc de conexiuni [14]
Tipul interconexiunilor ce utilizează regula lui Hebb	interconexiuni de control	interconexiuni de reacție	interconexiuni de reacție	interconexiuni de reacție
Dimensiunea rețelei	12 x 12	1. 10 x 10 2. 9 x 9	9 x 9	9 x 9
Vecinătatea	r = 1	r = 1	1. r = 1 2. r = 3	1. r = 1 2. r = 3
Numărul de modele memorate	3	1. 5 modele care respectă condițiile de proiectare 2. 25 modele care nu respectă condițiile de proiectare	25 modele	25 modele și combinațiile lor liniare
Nr. de interconexiuni	1296	1. 900 2. 729	1. 729 (unele pot fi zero) 2. 3969 (unele pot fi zero)	1. 729 (multe sunt zero) 2. 3969 (multe sunt zero)
Nr. de modele / nr. de interconexiuni	0.0023 (teoretic)	1. 0.005 2. 0.0343	1. mai mare decât 0.0343 2. mai mare decât 0.0063	1. mai mare decât 0.0686 2. mai mare decât 0.0126
Capabilitatea de corectare a erorilor	mică zgomot binar 11%	1. mare, zgomot gaussian: $\sigma=0,9$ zgomot binar 25% 2. mică zgomot gaussian: $\sigma=0,3$	mare zgomot gaussian 1. $\sigma=0,2$ 2. $\sigma=0,7$ zgomot binar: 1 pixel	mare zgomot gaussian 1. $\sigma=0,4$ 2. $\sigma=0,7$ zgomot binar: 1 pixel
Capacitatea de memorare (cu restricții)	reducă (cu restricții)	reducă (cu restricții)	mare (25)	mare (50)
Memorarea aditivă	da	da	da	da
Stări nedorite	da	da	da	da
Complexitatea	simplă	simplă	relativ simplă	simplă
Viteza de învățare	mare	mare	mare	foarte mare

situații. Memoria prin RNC, cu posibilitatea utilizării chipului VLSI, oferă o soluție viabilă, operațională în timp real, similară cu procesarea umană. Metoda este flexibilă la schimbarea traseului și este mai ieftină decât variantele convenționale. Abordarea prin RNC are și avantajul toleranței erorilor, a instalării ușoare și a unui control local. Datorită potențialului RNC eventuale îmbunătățiri pot fi aduse fără dificultăți. Singura limită este dată de capacitatea de memorare.

Toate memoriile ce au la bază regula lui Hebb își îmbunătățesc performanțele cu creșterea vecinătății. Algoritmii propuși nu asigură controlul traiectoriei, deci teoretic este posibilă blocarea în minime locale, nedorite. Practic însă, situațiile în care sistemul a convers spre stări nedorite au fost extrem de rare.

Simplitatea, aditivitatea, și de aici viteza de procesare a metodelor ce utilizează regula lui Hebb, similaritatea cu modul de procesare a neuronilor umani sunt câteva dintre argumentele continuării cercetărilor în această direcție, în ciuda faptului că rezultatele obținute până în prezent, nu sunt spectaculoase.

Un studiu comparativ al MA ce utilizează regula lui Hebb cu MA obținute prin alte metode este prezentat în capitolul 9 (§ 9.6).



# Capitolul 7

## MEMORII ASOCIATIVE REALIZATE PRIN UTILIZAREA METODEI RELAXĂRII

### 7.1 Considerații teoretice

Limitele memoriei asociative ce utilizează regula hebbiană, au condus la studiul altor metode pentru determinarea matricii de conexiune.

Un exemplu este memoria asociativă care utilizează metoda relaxării, elaborată de R.Perfetti și G.Martinelli [65].

Se consideră modelul analogic al RNC introdus de sistemul de ecuații (2.1). Se presupune că vecinătatea unei celule este dată de  $r=1$ , că nu există intrare de control,  $B=0$  și polarizarea este nulă  $I=0$ . Se impune condiția de simetrie, care garantează completa stabilitate [27]:

$$A_{ij,kl} = A_{kl,ij} \quad (7.1)$$

Se știe că dacă  $A_{ij,kl} > 1$ , singurele soluții observabile corespund ieșirilor bipolare +1 sau -1.

S-a demonstrat [96] că o condiție necesară și suficientă pentru existența unor soluții asimptotice stabile, bipolare, este:

$$\begin{aligned} A_{i,j;k,l} y_{i,j} - 1 &\geq 0 \text{ dacă } y_{i,j} = +1 \\ -A_{i,j;k,l} y_{i,j} - 1 &\geq 0 \text{ dacă } y_{i,j} = -1 \end{aligned} \quad (7.2)$$

Această condiție poate fi folosită fie pentru a determina parametrii rețelei, fie pentru a verifica dacă un vector bipolar reprezintă sau nu o soluție a rețelei. Impunând restricțiile (7.2) tuturor celor  $P$  modele de memorat, se poate obține un set de constrângeri liniare, care pot fi rezolvate fie prin metoda relaxării numerice [109], fie prin programarea liniară. Ținând cont de simetria conexiunilor (7.1), legătura dintre două celule se caracterizează printr-o singură pondere.

Presupunând că autoreacția  $A_{ij,j}$  se alege apriori, numărul parametrilor nedeterminați este:

$$L = (N - 1).M + (M - 1).N - 2.(N - 1).(M - 1) \quad (7.3)$$

Pentru a obține cele  $L$  necunoscute trebuie satisfăcute  $P \times M \times N$  constrângeri simultane. Fie  $S$  spațiul soluțiilor. Se introduc următoarele notații:

- $A_0$ , autoreacția comună tuturor celulelor ( $A_0 - 1$ );
- $A = [A_{11,12} \ A_{11,21} \ A_{11,22} \ \dots] \in R^L$ , matricea interconexiunilor necunoscute de reacție;
- $A_{i,j} \in R^{M \times N}$   $i \in \{1,2, \dots, M\}$ ,  $j \in \{1,2, \dots, N\}$ , un subvector care conține necunoscutele relative la celula  $C_{ij}$ :

$$A_{i,j} = [A_{i,j;i-1,j-1} \ A_{i,j;i-1,j} \ A_{i,j;i-1,j+1} \ A_{i,j;i,j+1} \ A_{i,j;i+1,j-1} \ A_{i,j;i+1,j} \ A_{i,j;i+1,j+1}]^T$$

- $Y_{ij}^q \in \{-1,+1\}^{M \times N}$  un vector care conține doar ieșirile relative la celula  $C_{ij}$  pentru modelul  $q$  de memorat, excepție făcând termenul  $y_{i,j}$ :

$$Y_{ij}^q = [y_{i-1,j-1} \ y_{i-1,j} \ y_{i-1,j+1} \ y_{i,j-1} \ y_{i,j+1} \ y_{i+1,j-1} \ y_{i+1,j} \ y_{i+1,j+1}]^T$$

**Algoritmul de învățare** este următorul:

1. Se inițializează arbitrar  $A \in R^L$
2. Pentru  $q=1$  până la  $P$  (numărul modelelor prezentate rețelei),

Pentru fiecare  $i=1 \div M$  și  $j=1 \div N$ ,

Dacă  $(y = 1)$  și  $(A_{i,j}^T Y_{ij}^q + A_0 - 1 < 0)$

$$\text{Atunci: } A_{i,j} = A_{i,j} - 2 \frac{A_{i,j}^T Y_{ij}^q + A_0 - 1}{\|Y_{ij}^q\|^2} \cdot Y_{ij}^q$$

Dacă  $(y_{i,j} = -1)$  și  $(-A_{i,j}^T Y_{ij}^q + A_0 - 1 < 0)$

$$\text{Atunci: } A_{i,j} = A_{i,j} - 2 \frac{A_{i,j}^T Y_{ij}^q - A_0 + 1}{\|Y_{ij}^q\|^2} \cdot Y_{ij}^q$$

3. Dacă  $DA$ ,  $A \in S$  atunci Stop.

Dacă NU se reia procedura din pasul 2.

## 7.2 Rezultatele simulării

În fig.7.1 se prezintă rezultatele simulării [65] unei memorii asociative construită prin metoda relaxării pentru memorarea celor șase imagini ideale  $P^1, P^2, \dots, P^6$ . Algoritmul expus a procesat  $L=110$  valori pentru conexiuni și a convers după 105 iterații. La intrare au fost aplicate modele afectate de zgomot și rețeaua care a utilizat ponderile calculate a generat ca stare finală imaginile dorite.

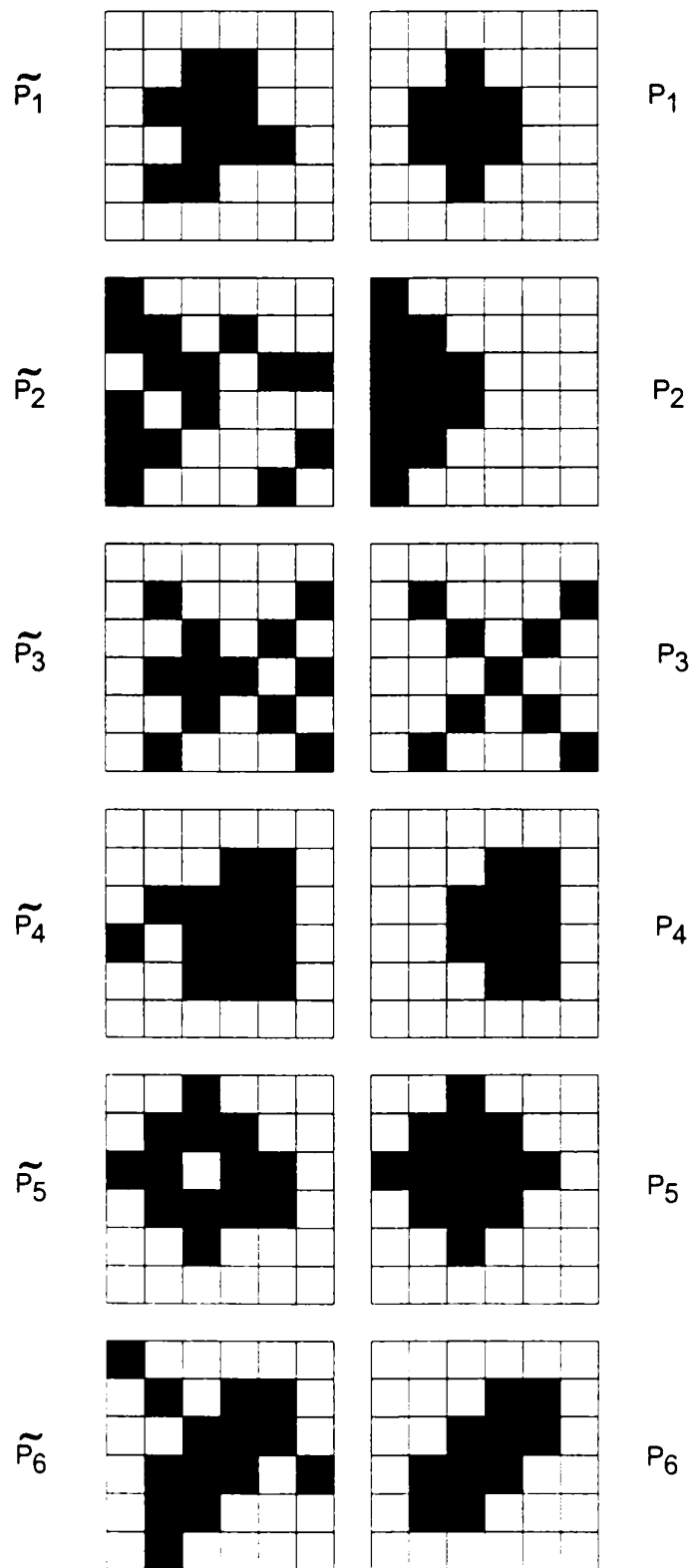


Fig. 7.1 Modelele de intrare afectate de zgomot binar

### Observații

Metoda garantează că soluțiile sistemului de inegalități obținut (dacă există) reprezintă stări stabile ale rețelei. Algoritmul este relativ laborios și el necesită reprojectare pentru încorporarea în memorie a unui nou model.

Calculând raportul dintre numărul de modele și numărul de interconexiuni, obținem valoarea de 0,0545, o valoare bună comparativ cu a unei rețele neuronale complet conectate.

# Capitolul 8

## DETERMINAREA MATRICII DE CONEXIUNE PRIN METODA DESCOMPUNERII ÎN VALORI SINGULARE

Interconexiunile simetrice sunt dificil de implementat în tehnologie VLSI, motiv pentru care ar trebui evitate. Interconexiunile de tip complet (RN fully connected), care pot determina intersecții de legături chiar în cazul în care numărul neuronilor este redus, pun și ele probleme deosebite în implementările fizice concrete. Astfel, s-a impus necesitatea proiectării unor rețele nesimetrice și cu un număr redus de conexiuni. Una dintre metodele cele mai eficiente în proiectarea rețelelor neuronale utilizate ca memorii asociative este metoda descompunerii în valori singulare a modelelor de intrare.

### 8.1 Considerații teoretice

În 1993 D. Liu și A.N. Michel [59] au publicat o procedură de sinteză pentru memorarea unui set de modele dorite, pentru RNCA nesimetrice cu interconexiuni locale predeterminate.

Se consideră RNCA, cu  $n$  celule, descrisă de sistemul de ecuații:

$$\begin{aligned}\dot{\mathbf{x}} &= -\mathbf{C}\cdot\mathbf{x} + \mathbf{A}\cdot\mathbf{f}(\mathbf{x}) + \mathbf{I} \\ \mathbf{y} &= \mathbf{f}(\mathbf{x})\end{aligned}\tag{8.1}$$

unde:

- $\mathbf{x}$  este derivata în funcție de timp a vectorului de stare al celulelor rețelei și are dimensiunea  $n$ ;
- $\mathbf{C}$  este o matrice diagonală cu  $c_i \neq 0 = \text{diag} [c_1, c_2, \dots, c_n]$  pentru  $i \in \{1, 2, \dots, n\}$ ;
- $\mathbf{A}$  este matricea de reacție inversă, de numere reale și dimensiune  $n \times n$ .  
 $\mathbf{A} = [A_{ij}] \in \mathbb{R}^{n \times n}$ ;
- $\mathbf{I}$  este vectorul polarizărilor,  $\mathbf{I} = (I_1, I_2, \dots, I_n)^T$ ;

- $y=f(x) = [f(x_1), f(x_2), \dots, f(x_n)]^T$  este vectorul ieșirilor,  $y \in Y^n$ , iar funcția  $f(\cdot)$  este o funcție sigmoidă dată de relația:

$$f(x_i) = \begin{cases} 1 & \text{dacă } x_i > 1 \\ x_i & \text{dacă } -1 \leq x_i \leq 1 \\ -1 & \text{dacă } x_i \leq -1 \end{cases} \quad (8.2)$$

Se presupune că stările inițiale ale celulelor satisfac condiția  $|x_i(0)| \leq 1$  pentru  $\forall i \in \{1, 2, \dots, n\}$ . Pentru simplitate se poate considera  $c_i = 1$ .

Fie  $P$  vectori  $n$  dimensional  $\alpha^1, \alpha^2, \dots, \alpha^P$ , modelele pe care dorim să le memorăm.

Pentru  $\forall \alpha \in B^n$ ,  $B^n = \{x \in \mathbb{R}^n : x_i = -1 \text{ sau } x_i = 1, \forall i \in \{1, 2, \dots, n\}\}$  se definește

$D(\alpha) = \{x \in \mathbb{R}^n : x_i \alpha_i > 1, \forall i \in \{1, 2, \dots, n\}\}$ .

### Lema 1

Dacă vectorul  $\beta = A\alpha + I \in D(\alpha)$ , atunci  $\beta$  este un punct de stabilitate asimptotică a sistemului (8.1)

Deoarece  $\alpha_i = \pm 1$ , din relația (8.1) rezultă că  $f(x) = \alpha$  pentru orice  $x \in D(\alpha)$ . Pentru  $x \in D(\alpha)$  ecuația (8.1) poate fi scrisă sub forma:

$$\dot{x} = -C \cdot x + A \cdot \alpha + I \quad (8.3)$$

Se observă că soluția ecuației (8.3) este  $x_c = A \cdot \alpha + I$ , deoarece în stare stabilă valoarea derivatei din (8.3) este nulă (în conformitate cu cele afirmate la stabilitate) și  $x_c = \beta \in D(\alpha)$  reprezintă singurul punct de echilibru. Acest punct de echilibru este evident și asimptotic stabil, deoarece sistemul descris de ecuația (8.3) are toate cele  $n$  valori proprii egale cu  $-1$  (Soluțiile negative reale sunt condiția necesară și suficientă pentru existența stărilor stabile într-o ecuație diferențială).

### Definiția 11

Dacă  $\xi \in \mathbb{R}^n$  este un punct de echilibru asimptotic stabil al sistemului (8.1), atunci  $y_\xi = f(\xi)$  se numește **vector memorie (memory vector)** al sistemului.

Sistemul (8.1) poate atinge un vector memorie  $y_\xi$  dacă există o vecinătate  $N_r$  a lui  $y_\xi$  astfel încât pentru orice  $x(0) \in N_r \cap Y^n$ , vectorul de ieșire  $y(t)$  al sistemului (8.1) tinde asimptotic către  $y_\xi$  când  $t \rightarrow \infty$ .

Fie vecinătatea  $N_r$  a unei celule  $C_{i,j}$ , conform definiției 1, dată de relația:

$$N_r = \{ C_{k,l} : \max \{ |k - i|, |l - j| \} \leq r \} \quad (8.4)$$

Pentru fiecare rază  $r$  se poate defini o matrice index  $S=[S_{ij,kl}] \in R^{n \times n}$  (unde  $n=M \times N$  pentru o rețea bidimensională), care arată apartenență celulei  $C_{kl}$  la vecinătatea  $N_r$  a celulei  $C_{ij}$ .

$$\begin{aligned} S_{ij,kl} &= 1 && \text{dacă } C_{kl} \in N_r(ij) \\ S_{ij,kl} &= 0 && \text{în rest} \end{aligned} \quad (8.5)$$

Fie  $W|S=[h_{ij}] \in R^{n \times n}$  restricția lui  $W=[W_{ij}] \in R^{n \times n}$  pe matricea index  $S$ , ale cărei elemente sunt date de relația:

$$h_{ij} = \begin{cases} 0 & \text{daca } S_{ij} = 0 \\ W_{ij} & \text{daca } S_{ij} = 1 \end{cases} \quad (8.6)$$

### Definiția 12

Vom spune că sistemul (8.1) reprezintă o *rețea neuronală cu o matrice de interconexiune rară* dacă există o restricție  $A=A|S$ . (În literatură o matrice de dimensiune  $n \times n$  se numește rară dacă numărul elementelor sale nenule este  $\ll n^2$ ).

Fiind date cele  $P$  modele pe care dorim să le memorăm  $\alpha^1, \alpha^2, \dots, \alpha^P$  și numerele întregi pozitive  $r, n, M, N, n=M \times N$ , care definesc rețeaua neuronală, problema care se pune este determinarea matricii de reacție inversă  $A$  (care să reprezinte restricția matricii  $S$  date, definite ca în (8.6)) și a matricii  $I$  de polarizare.

**Procedura de sinteză** conține următorii pași:

1. Se alege  $C$  matricea identitate. Se determină matricea  $S$  ca în (8.5)  $S=[S_{ij}] \in R^{n \times n}$ .
2. Se alege un număr real  $k > 1$  și  $P$  vectori  $n$  dimensional  $\beta^1, \beta^2, \dots, \beta^P$  astfel încât  $\beta^i = k \cdot \alpha^i$ .
3. Se procesează matricea  $Y$  de dimensiune  $n \times (m-1)$ ,  $Y=[y^1, y^2, \dots, y^{m-1}] = [\alpha^1 - \alpha^P, \dots, \alpha^{P-1} - \alpha^P]$  și  $Z=[z^1, z^2, \dots, z^{m-1}] = [\beta^1 - \beta^P, \beta^2 - \beta^P, \dots, \beta^{P-1} - \beta^P]$ .  
Notăm cu  $y^i = [y^i_1, y^i_2, \dots, y^i_n]^T$  și cu  $z^i = [z^i_1, z^i_2, \dots, z^i_n]^T$  pentru  $i \in \{1, 2, \dots, m-1\}$ .
4. Se notează rândul  $i$  a matricii index  $S$  cu  $S_i = S_{i1}, S_{i2}, \dots, S_{in}$ . Pentru fiecare  $i \in \{1, 2, \dots, n\}$  se construiesc două mulțimi de numere  $M_i$  și  $N_i$ , astfel încât:  
 $M_i \cup N_i = \{i = 1, 2, \dots, n\}$   
 $M_i \cap N_i = \emptyset$ .  
 $S_{ij} = 1$  dacă  $j \in M_i$  și  $S_{ij} = 0$  dacă  $j \in N_i$ .

În  $M_i$  se găsesc, în ordine crescătoare, indicii elementelor diferite de zero din rândul  $i$  al matricii  $S$ ,  $M_i = (\sigma_i(1), \dots, \sigma_i(m_i))$  unde  $m_i = \sum_{j=1}^n S_{ij}$  și  $\sigma_i(k) < \sigma_i(l)$  dacă  $1 < k < l < m_i$  (se observă că  $m_i$  este numărul elementelor de 1 din rândul  $i$  al matricii  $S$ ). Mulțimea  $N_i$  conține în ordine crescătoare, indicii elementelor egale cu zero din rândul  $i$  al matricii  $S$ .

5. Pentru  $i = 1, 2, \dots, n$  și  $l = 1, 2, \dots, P-1$  fie  $y_i = (y_{\sigma_i(1)}^l, \dots, y_{\sigma_i(p_i)}^l)^T$ . Din fiecare coloană a matricii  $Y$  se extrag acele elemente care sunt determinate de indicii  $\sigma_i$ .

6. Pentru  $i=1, 2, \dots, n$  se procesează matricile  $Y_i$  de dimensiune  $m_i \times (m - 1)$ ,  $Y_i = [y_i^1, \dots, y_i^{P-1}]$  și vectorii  $Z_i$  de dimensiune  $1 \times (P-1)$ ,  $Z_i = [z_i^1, \dots, z_i^{P-1}]$ . Matricile  $Y_i$  sunt formate pe coloane din vectorii determinați la punctul 5, iar  $Z_i$  reprezintă rândul  $i$  al matricii  $Z$  determinată la punctul 3.

7. Pentru  $i = 1, \dots, n$ , se realizează o descompunere în valori singulare matricilor  $Y_i$ , obținându-se:

$$Y_i = [U_{i1} | U_{i2}] \begin{bmatrix} D_i & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{i1}^T \\ V_{i2}^T \end{bmatrix} \quad (8.7)$$

unde:

- $D_i \in \mathbb{R}^{p_i \times p_i}$ , este o matrice diagonală, care conține pe diagonală valorile singulare nenule ale matricii  $Y_i$  și  $p_i$  reprezintă rangul matricii  $Y_i$ ;
- matricile  $U$  și  $V$  rezultate din descompunere sunt matrici ortogonale (a se vedea Anexa2) și dimensiunile lor sunt:

$$\begin{array}{ll} U_{i1}: m_i \times p_i & U_{i2}: m_i \times (m_i - p_i) \\ V_{i1}: (P-1) \times p_i & V_{i2}: (P-1) \times (P-1 - p_i) \end{array}$$

**Observație.** În continuare se consideră că s-a realizat descompunerea în valori singulare pentru matricea  $Y_i$ . În Anexa 2 se prezintă algoritmul de descompunere în valori singulare ce utilizează transformate Householder.

8. Se calculează pentru  $i=1, \dots, n$  vectorii  $G_i$  în modul următor:

$$G_i = [G_{i1}, \dots, G_{im_i}] = Z_i \cdot V_{i1} \cdot D_i^{-1} \cdot U_{i1}^T + W_i \cdot U_{i2}^T,$$

unde  $W_i$  este un vector real oarecare de dimensiune  $1 \times (m_i - p_i)$ .

9. Matricea  $A = [A_{ij}]$  se procesează astfel:

- $A_{ij} = 0$  dacă  $S_{ij} = 0$
- $A_{ij} = G_{ik}$  dacă  $S_{ij} \neq 0$  și dacă  $j = \sigma_i(k)$

Deoarece în fiecare vector  $G_i$  se găsesc  $m_i$  elemente, rezultă că și în linia  $i$  a matricei  $Q$  se găsesc tot  $m_i$  elemente. Matricea  $A$  corespunde cu matricea  $Q$  cu singura diferență că în locul elementelor diferite de zero conține vectorii  $G_i$ .

10. Se determină elementele vectorului de polarizare,  $I=[I_1, \dots, I_n]^T$ , cu relația  $I_i = \beta_i^P - A_i \cdot \alpha^P$ , pentru  $i=1, \dots, n$ , unde  $A_i$  este rândul  $i$  al matricii  $A$ .

Procedura descrisă mai sus asigură ca  $\alpha^1, \alpha^2, \dots, \alpha^P$  să fie vectorii memorie pentru o RNCA descrisă de ecuația (8.1). Stările  $\beta^1, \dots, \beta^P$  corespunzătoare vor deveni puncte de echilibru asimptotic stabil.

Validitatea procedurii se bazează pe următoarele teoreme:

### Teorema 10

Întotdeauna există soluții pentru problema descompunerii în valori singulare, dacă există  $S_{ii} = 1$  pentru  $i=1, 2, 3, \dots, n$

Procedura de sinteză garantează că matricea de interconexiuni de reacție  $A$  este o restricție la matricea  $S$ ,  $A=A|S$

Procedura de sinteză garantează că toți vectorii din  $L_\alpha \cap B^n$  inclusiv  $\alpha^1, \alpha^2, \dots, \alpha^P$  sunt memorați de sistemul (8.1), unde  $L_\alpha = \text{Aspan}(\alpha^1, \alpha^2, \dots, \alpha^P)$ .

Procedura de sinteză expusă poate fi aplicată oricărei mulțimi de modele dorite  $\alpha^1, \alpha^2, \dots, \alpha^P \in B^n$ .

### Teorema 11

Se presupune că  $\alpha^1, \alpha^2, \dots, \alpha^P$  sunt vectorii memorați de sistemul proiectat prin procedura modificată. Fie  $\varphi$  o combinație liniară a vectorilor  $\alpha^1, \alpha^2, \dots, \alpha^P$  și  $\varphi \in B^n$ . Atunci procedura expusă garantează că  $\varphi$  reprezintă un vector memorat de sistemul (8.1).

Demonstrațiile celor două teoreme sunt date în [60].

#### Observații:

1. În procedura descrisă mai sus s-a ales matricea  $C$  matrice identitate, pentru simplificarea calculelor. Dacă se alege  $C = \text{diag}[c_1, c_2, \dots, c_n]$ , cu  $c_i > 0$ , este necesar să schimbăm  $Z_i$  în pasul 6 cu  $Z_i = [c_i z_{i1}, \dots, c_i z_{iP-1}]$  și vectorul  $I_i$  în pasul 10 cu  $I_i = \alpha_i \times \beta_i^P - A_i \times \alpha^P$ .

2. Dacă în matricea index  $S$  există  $q > 1$  rânduri identice, se pot proiecta simultan cele  $q$  rânduri din matricea  $A$ . Pentru aceasta se modifică puțin pașii 5-9. Printr-o alegere adecvată a matricii  $S$  se poate obține o matrice cu puține intersecții ale interconexiunilor sau chiar deloc.



3. Dacă dorim obținerea unei rețele care nu utilizează vectorul de polarizare,  $I = 0$ , procedura de sinteză se modifică astfel:

- În pasul 3 se alege  $Y=[\alpha^1, \alpha^2, \dots, \alpha^P]$  și  $Z=[\beta^1, \dots, \beta^P]$
- În pasul 10 se ia  $I=0$ . În particular, fiecare  $-\alpha^i$  și  $-\beta^i$   $i=1,2,\dots,m$  va fi de asemenea un vector memorie, respectiv un punct de stabilitate asimptotică pentru sistemul considerat.

4. Prin algoritmul anterior expus se memorează și combinațiile liniare ale modelelor înmagazinate.

5. În 1996 s-a pus la punct un simulator soft cu matrici de conexiune variabile în spațiu în virgulă flotantă [112]. Foarte probabil însă că precizia necesară nu va putea fi prea curând abordabilă în implementările analogice.

## 8.2 Rezultatele simulării

Pentru a demonstra aplicabilitatea metodei s-au făcut mai multe experimente.

Algoritmul a fost implementat în MATLAB, atât pentru cazul în care polarizarea a fost zero, cât și pentru cazul în care a fost diferită de zero. Matricile de interconexiuni au fost obținute pe arhitectura unei RN de tip complet conectate, în care s-a ținut cont de localitatea conexiunilor RNC. Acestea au fost modificate, în Borland C, într-o formă adecvată simulatorului soft RNC, variant în spațiu.

Se observă că în pasul 2 al algoritmului valoarea  $k$ , respectiv în pasul 8 vectorul  $W_i$  pot fi alese. Pentru exemplele considerate în această lucrare am ales  $k=2$  și  $W_i = -10 \times O_{m_i} \times U_{i2}$ , unde  $O_{m_i} = [1, 1, \dots, 1]$  și are dimensiunea  $1 \times m_i$  și  $U_{i2}$  este matricea de la pasul 7.

### Exemplul 1

Fie următoarele modele de memorat, de dimensiune  $M=4$ ,  $N=3$  :

$$\alpha^1 = [1, -1, 1, 1, -1, 1, 1, 1, -1, -1, 1]^T$$

$$\alpha^2 = [1, 1, 1, 1, -1, -1, 1, -1, -1, 1, 1]^T$$

$$\alpha^3 = [1, -1, 1, 1, -1, 1, 1, -1, 1, 1, 1]^T$$

$$\alpha^4 = [1, -1, -1, 1, -1, -1, 1, -1, -1, 1, 1]^T$$

Fig.8.1 prezintă cele patru modele ideale.

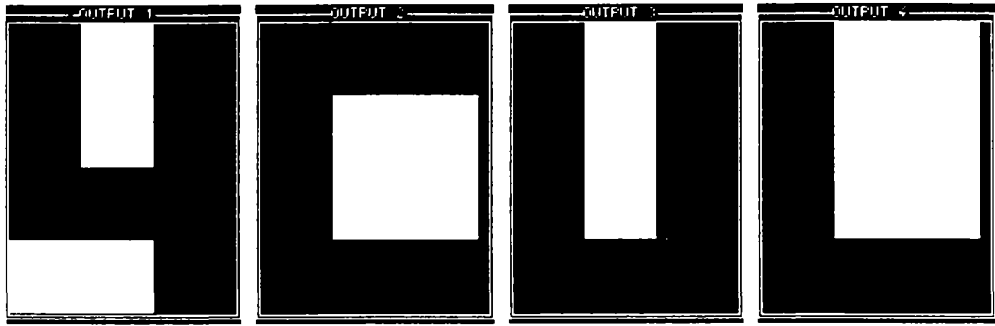


Fig. 8.1 Cele 4 modele memorate de MA de dimensiune 4x3, obținută prin metoda descompunerii în valori singulare

Dacă se alege vecinătatea RNC,  $r=1$ , matricea index S este:

$$S_{ij} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Am ales în procedura de sinteză C matricea identitate, matricea de polarizare I diferită de zero.

Matricea de interconexiuni obținută este aceeași cu cea dată în [59]:

-1.0000000000000000e+001	0.0000000000000000e+000	0.0000000000000000e+000
1.0000000000000000e+001	-1.0000000000000000e+001	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
-0.9999999999999999e+000	-1.9999999999999999e+000	-4.0000000000000000e+000
-1.0000000000000000e+001	-1.0000000000000000e+001	-3.9999999999999999e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000

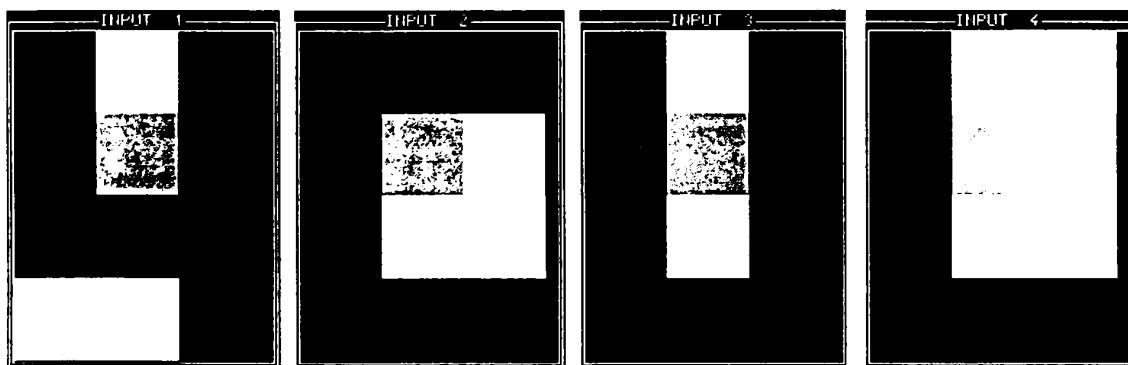
0.0000000000000000e+000	-2.666666666666670e+000	4.666666666666670e+000
0.0000000000000000e+000	-1.0000000000000000e+001	-2.666666666666680e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
-1.0000000000000000e+001	0.0000000000000000e+000	0.0000000000000000e+000
-1.0000000000000000e+001	-1.0000000000000000e+001	0.0000000000000000e+000
-1.0000000000000000e+001	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
-9.999999999999980e+000	-4.000000000000010e+000	3.999999999999990e+000
-1.0000000000000000e+001	-1.0000000000000000e+001	-2.0000000000000000e+000
-1.0000000000000000e+001	9.8224878080175680e-016	-2.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	-4.400000000000000e+000	4.400000000000000e+000
0.0000000000000000e+000	-1.0000000000000000e+001	-1.2000000000000000e+000
0.0000000000000000e+000	1.1575131833431020e-015	-1.2000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
-9.999999999999980e+000	-1.0000000000000000e+001	0.0000000000000000e+000
-1.0000000000000000e+001	-1.333333333333340e+001	0.0000000000000000e+000
-6.666666666666680e+000	-6.666666666666680e+000	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
-1.0000000000000000e+001	-9.999999999999980e+000	6.0006244518627630e-016
-1.0000000000000000e+001	-1.266666666666670e+001	6.0006244518627630e-016
-7.333333333333320e+000	-7.333333333333320e+000	-1.0000000000000000e+001
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	-1.0000000000000000e+001	9.999999999999920e-001
0.0000000000000000e+000	-1.0000000000000000e+001	9.999999999999940e-001
0.0000000000000000e+000	-1.0000000000000000e+001	-1.0000000000000000e+001
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
-9.999999999999980e+000	-1.4000000000000000e+001	0.0000000000000000e+000
-6.000000000000010e+000	-6.000000000000010e+000	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
-1.0000000000000000e+001	-1.4000000000000000e+001	1.8505343623843200e-015
-6.000000000000000e+000	-6.000000000000000e+000	-1.0000000000000000e+001
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	-1.0000000000000000e+001	2.1351110827315980e-016
0.0000000000000000e+000	-1.0000000000000000e+001	-1.0000000000000000e+001

Matricea de polarizare este:

1.2000000000000000e+001	5.9999999999999980e+000	-1.2666666666666670e+00
2.2000000000000000e+001	1.4000000000000000e+001	-1.4400000000000000e+001
1.2000000000000000e+001	2.0000000000000000e+001	-4.8849813083506880e-001
1.0000000000000000e+001	2.0000000000000000e+001	1.2000000000000000e+001

Au fost corectate modelele de intrare afectate de o eroare  $\sigma=0.2$ , conform figurii 8.2

MA obținută prin descompunerea în valori singulare a modelelor de intrare, de această dimensiune, are și o slabă capacitate de corecție a zgomotului binar, corecția obținută fiind de 1 pixel.



**Fig. 8.2 Modelele de intrare, reprezentând cele 4 modele memorate afectate de un zgomot gaussian cu medie nulă și dispersie  $\sigma=0.2$**

## Exemplul 2

Modelele memorate sunt linii verticale și orizontale, într-o RNC de dimensiune 4x4.

Fig. 8.4 prezintă modelele de intrare, cu o eroare gaussiană cu medie nulă și dispersie  $\sigma=1$  și ieșirile MA după stabilizarea rețelei, reprezentând chiar modelele dorite.

Mărind dispersia zgomotului am putut constata o mai bună corecție la zgomot mai mare, pentru liniile de margine, pentru  $r=2$  fiind corecțati inclusiv pixelii eronați din colțuri. Cele 4 modele corectate pentru  $\sigma=1.2$  au fost liniile de pe cele patru laturi ale imaginii.

Acest fenomen a fost mai evident în cazul corectării zgomotului binar. Pentru modelele care implică linii de interior corecția a fost mai slabă decât în cazul liniilor de margine. De exemplu 2 pixeli au fost corecțati în orice situație pentru liniile de margine, pentru cele din interior au fost corectati doar în cazul în care modelul eronat a conținut cel puțin 3 pixeli din linia ce trebuia reconstituită. Am constatat că în cazul în care 3 pixeli de pe o linie au fost corecți modelele ideale au fost corect apelate din memorie. Fig.8.6 prezintă câteva situații în care memoria a corectat un zgomot binar de 18% cu o rază a vecinătății  $r=2$ .

Tabelul 8.2

Zgomot / vecinătate	Vecinătate $r=1$	Vecinătate $r=2$	Vecinătate $r=3$
gaussian $\sigma=1$	8 (toate) modele corect apelate din memorie	8 modele corect apelate din memorie	8 modele corect apelate din memorie
gaussian $\sigma=1,1$	3 modele corect apelate din memorie, restul modelelor afectate cu o eroare de 2-3 pixeli	4 modele corect apelate din memorie	6 modele corect apelate din memorie
gaussian $\sigma=1,2$	2 modele corect apelate din memorie, restul modelelor afectate cu o eroare de 1-6 pixeli	4 modele corect apelate din memorie	6 modele corect apelate din memorie
binar 1 pixel inversat	8 modele corecte	8 modele corect apelate din memorie	8 modele corect apelate din memorie
binar 2 pixeli inversați	4 modele corecte, celelalte 4 diferențiat în funcție de poziția pixelilor eronați	4 modele corecte, celelalte 4 diferențiat în funcție de poziția pixelilor eronați	6 modele corect apelate din memorie

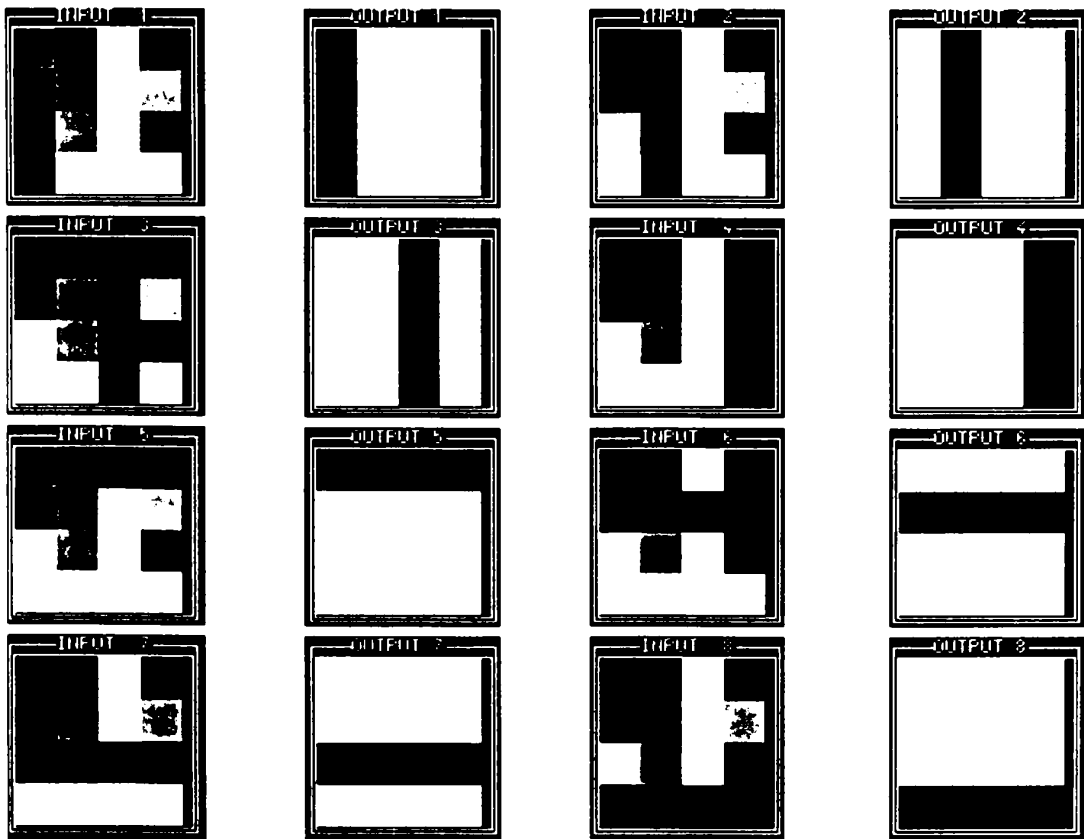


Fig. 8.3 Modele de intrare, reprezentând cele 4 modele memorate, afectate de un zgomot gaussian cu medie nulă și dispersie  $\sigma=1$  și ieșirile obținute după stabilizarea rețelei

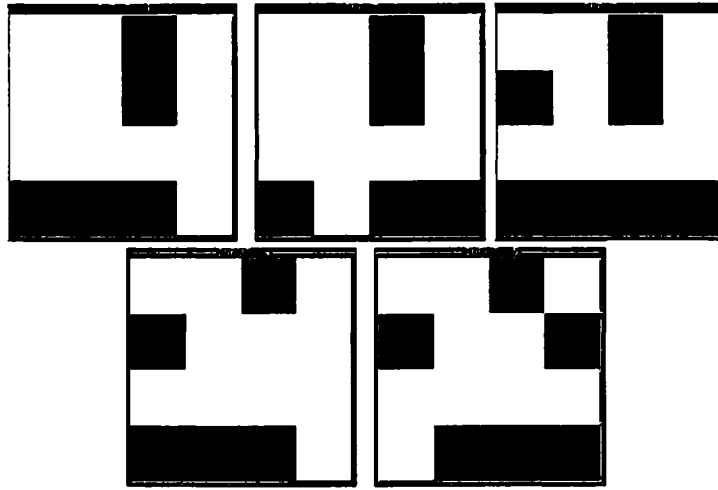


Fig. 8.4 Câteva exemple de modele cu un zgomot gaussian de 3 pixeli apelate corect de MA pentru o vecinătate  $r=2$

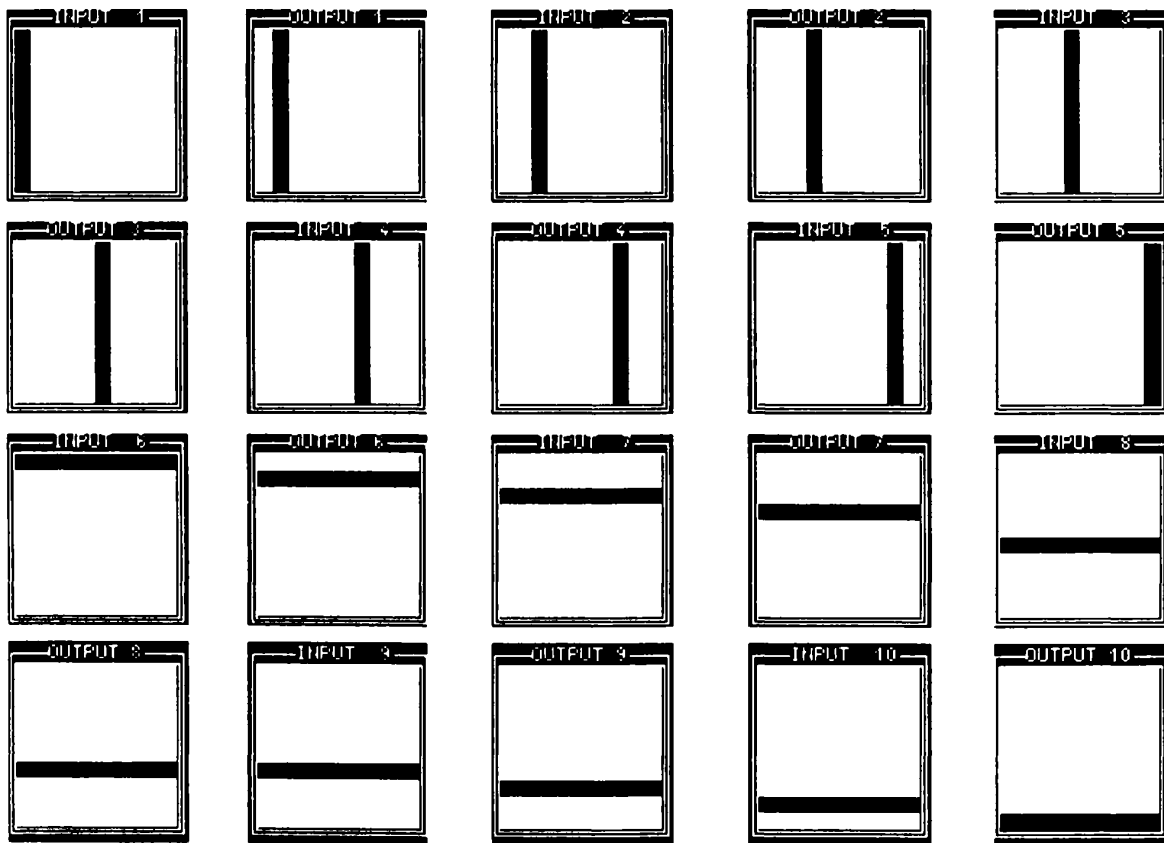


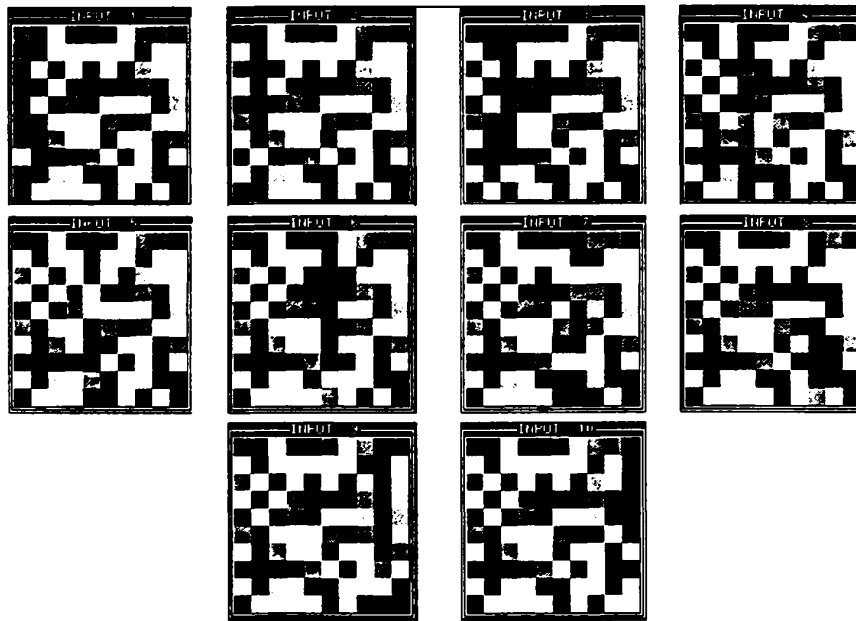
Fig. 8.5 Cele 20 de modele memorate de o MA de dimensiune 10x10, obținută utilizând metoda descompunerii în valori singulare

### Exemplul 3

Cele 20 de modele ideale memorate sunt liniile verticale și orizontale, din Fig.8.5, într-o RNC de dimensiune 10x10.

Tabelul 8.3

Zgomot / vecinătate	Vecinătate $r=1$	Vecinătate $r=2$	Vecinătate $r=3$
gaussian $\sigma=0,8$	20 modele corect apelate din memorie	20 modele corect apelate din memorie	20 modele corect apelate din memorie
gaussian $\sigma=0,9$	3 modele corect apelate din memorie, restul cu o eroare de 2-4 pixeli	20 modele corect apelate din memorie	20 modele corect apelate din memorie
gaussian $\sigma=1$	toate modelele afectate cu o eroare de 1-4 pixeli	20 modele corect apelate din memorie	20 modele corect apelate din memorie
gaussian $\sigma=1,1$	_____	19 modele corect apelate din memorie, unul cu 2 pixeli eroare	20 modele corect apelate din memorie
gaussian $\sigma=1,1$	_____	17 modele corect apelate din memorie, 3 cu 4 pixeli eroare	20 modele corect apelate din memorie
gaussian $\sigma=1,3$	_____	_____	20 modele corect apelate din memorie
gaussian $\sigma=1,4$	_____	_____	19 modele corect apelate din memorie
binar 7% pixeli inversați	20 modele corecte	20 modele corect apelate din memorie	20 modele corect apelate din memorie
binar 10% pixeli inversați	toate modelele afectate cu o eroare de 2-3 pixeli	20 modele corect apelate din memorie	20 modele corect apelate din memorie
binar 15% pixeli inversați	_____	20 modele corect apelate din memorie	20 modele corect apelate din memorie
binar 20% pixeli inversați	_____	trebuie experimentate	trebuie experimentate



**Fig. 8.6 Cele 20 de modele memorate afectate de un zgomot gaussian cu medie nulă și dispersie  $\sigma=1,3$**

Tabelul 8.2 sintetizează rezultatele testelor efectuate atât pentru un zgomot gaussian cu medie nulă și dispersii cât și pentru zgomot gaussian, în condițiile unor diferite vecinătăți.

Se poate constata că performanța a crescut cu dimensiunea vecinătății.

Cea mai bună performanță a fost înregistrată de o MA cu vecinătate  $r=3$ , care a permis apelul corect al modelelor memorate în cazul când ele au fost afectate la intrare cu un zgomot gaussian cu medie nulă și dispersie  $\sigma=1,3$ , prezentate în Fig. 8.6.

Pentru  $r=3$ , MA obținută a avut o foarte bună corecție a zgomotului binar pentru aceste modele. Am observat dependența capacității de corecție de distribuția zgomotului. Performanțele prezentate în tabel sunt în cazul în care fiecare linie din model a avut cel puțin 6 pixeli corecți.

#### **Exemplul 4**

S-a experimentat memorarea celor 25 de ideograme chinezești din Fig. 6.11.

S-au utilizat două tipuri de semnale aleatoare nedorite, suprapuse peste semnalul util și necorelate cu acesta: un zgomot Gaussian cu medie nulă și dispersie  $\sigma \in [0.0, 0.7]$  și unul digital, cu un procentaj de inversare a valorii pixelilor între 5 % și 10 %. Modelele afectate de zgomot gaussian cu  $\sigma=0.7$  sunt prezentate în Fig.6.14.

În [60] se raportează o corecție a zgomotului gaussian cu medie nulă și dispersie  $\sigma=1$ .

Rezultatele mai proaste obținute aici se datorează probabil metodei de iterație folosită de simulatorul CNNM variant în spațiu, metoda lui Euler cu aproximație spre înainte. Există algoritmi mai complecși cu performanțe mai bune, dar în [60] nu se precizează tipul folosit.

S-a testat și rolul parametrilor  $K$  și  $W_i$ .



Tabelul 8.4

Zgomot / vecinătate	Vecinătate r=1	Vecinătate r=2	Vecinătate r=3
gaussian $\sigma=0,4$	25 modele corect apelate din memorie	25 modele corect apelate din memorie	25 modele corect apelate din memorie
gaussian $\sigma=0,5$	12 modele corect apelate din memorie, restul cu o eroare de 1 pixel	25 modele corect apelate din memorie	25 modele corect apelate din memorie
gaussian $\sigma=0,6$	toate modelele afectate cu o eroare de 1-3 pixeli	10 modele corect apelate din memorie, 2 modele cu 2 pixeli eroare, 2 cu 3 pixeli eroare, 2 cu 4 pixeli eroare, restul cu 1 pixel eroare	25 modele corect apelate din memorie
gaussian $\sigma=0,7$	_____	_____	25 modele corect apelate din memorie
gaussian $\sigma=0,8$	_____	_____	18 modele corect apelate din memorie

MA obținută nu elimină zgomotul binar.

S-a ales  $K=1,5; 2; 4$ , obținându-se un număr mediu de iterații necesar stabilizării de 1.5, 8.8 și 7.5. Concluzia este că pe măsură ce  $K$  crește, numărul iterațiilor scade.

Modificând aleator  $W_i$ , am constatat că în unele situații rețeaua a început să oscileze.  $W_i$  s-a ales în funcție de matricea  $U$  obținută în urma descompunerii în valori singulare, deci probabil că e legată de modelele de memorat. Rolul lui  $W_i$  nu se poate preciza.

### 8.3 Concluzii

Avantajul principal al algoritmului este că nu este iterativ (adică o imagine nu trebuie învățată de mai multe ori). Toate imaginile sunt învățate identic, indiferent de compoziția lor.

Dezavantajele sunt următoarele:

- metoda expusă este laborioasă și nu este aditivă. Atunci când se dorește adăugarea unui model nou trebuie re-proiectată matricea de interconexiuni;
- generarea matricii de interconexiuni necesită un timp mai mare de procesare decât alte metode, ca de exemplu cele care utilizează regula lui Hebb;
- Expresia  $[(2 \cdot n_y + 1) \cdot N - n_y \cdot (n_y + 1)] \cdot [(2 \cdot n_x + 1) \cdot M - n_x \cdot (n_x + 1)]$  determină numărul de elemente de valoare 1 din matricea  $S$ , unde  $n_x$  și  $n_y$  reprezintă numărul de celule vecine pe direcția  $x$  respectiv  $y$ . Acest număr este numărul interconexiunilor din rețea. Se poate observa că el depinde de dimensiunea rețelei ( $M, N$ ) și de numărul de vecini într-un mod exponențial. Creșterea vecinătății determină deci creșterea duratei algoritmului.

- La fiecare utilizare trebuie determinat un număr de  $n=M \times N$  matrici  $Y_i$ , a căror dimensiune depinde de  $m_i$ ,  $m_i$  fiind numărul de 1 de pe linia  $i$  a matricei  $S$ , care la rândul său depinde de vecinătate. Dimensiunea matricii  $Y_i$  depinde și de numărul de imagini memorate ca vectori memorie pentru rețeaua neuronală celulară, această dependentă fiind una liniară. Creșterea numărului de imagini memorate de către rețeaua neuronală influențează într-o măsură mai mică viteza algoritmului.
- în această metodă matematică laborioasă precizia are un rol semnificativ.

Se poate constata că performanțele memoriei s-au îmbunătățit cu creșterea dimensiunii rețelei și a vecinătății, depinzând de modelele memorate.

În §9.6 se prezintă un studiu comparativ cu performanțele obținute de MA proiectate prin alte metode.

# Capitolul 9

## MEMORII ASOCIATIVE OBTINUTE PRIN METODA PSEUDOINVERSEI

Deși, la o primă evaluare, metoda pseudoinversei, expusă în Anexa 4, este radical diferită de învățarea hebbiană, se poate demonstra [51] că matricea dată de regula lui Hebb  $Y.X^T$  este un prim termen în dezvoltarea serială a matricii de interconexiune obținută prin formula pseudoinversei  $Y.X^T$ .

### 9.1 Memorie asociativă discretă

Metoda propusă în [18] exploatează proprietățile pseudoinversei.

Fie modelul discret al RNC descris de ecuația:

$$\begin{aligned}x[k+1] &= A.y[k] + I \\ y[k] &= f(x[k])\end{aligned}\tag{9.1}$$

$$f(x) = \frac{1}{2}(|2x+1| - |2x-1|)\tag{9.2}$$

unde:

- $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$
- $y = [y_1, y_2, \dots, y_n]^T \in \mathbb{R}^n$
- $I = [I_1, I_2, \dots, I_n]^T \in \mathbb{R}^n$
- matricea de reacție  $A$  este o matrice rară, care ține cont de localitatea conexiunilor RNC;

Fie  $m$  modele de memorat  $y^1, y^2, \dots, y^m$  și o rețea neuronală celulară cu  $n$  celule. Fiecare vector de memorie  $x$  trebuie să fie un punct asimptotic stabil [60].

## Teorema 12

Fie ecuația de echilibru:

$$x = A \cdot y + I \quad (9.3)$$

Dacă  $y^i = \pm 1$  și  $y^i \cdot x^i > 1$ ,  $i=1,2,\dots,n$ , atunci vectorul  $x$  este un punct asimptotic stabil pentru sistemul (1).

Demonstrația teoremei este dată în [21].

Constrângerea (9.3) poate fi satisfăcută dacă se consideră:

$$x^i = A \cdot y^i + I \quad i = 1, 2, \dots, m \quad (9.4)$$

unde:

- $x^i = [x_1^i, x_2^i, \dots, x_n^i]^T \in \mathbb{R}^n$
- $y^i = [y_1^i, y_2^i, \dots, y_n^i]^T \in \mathbb{R}^n$
- $x_j^i = k \cdot y_j^i$  și  $k > 1$ ;

Ecuația (9.4) poate fi scrisă și sub formă matriceală:

$$X = A \cdot Y + I' \quad (9.5)$$

unde:

- $X = [x^1, x^2, \dots, x^m] \in \mathbb{R}^{n \times m}$
- $Y = [y^1, y^2, \dots, y^m] \in \mathbb{R}^{n \times m}$
- $I' = [I, I, \dots, I] \in \mathbb{R}^{n \times m}$

Se introduc următoarele matrici:

$$T = [Y^T | J] \in \mathbb{R}^{m \times (n+1)}$$

$$I' = [I, I, \dots, I] \in \mathbb{R}^{n \times m}$$

$$J = [1, 1, \dots, 1]^T \in \mathbb{R}^{m \times 1}$$

$$w_j = [A_{j1}, A_{j2}, \dots, A_{jn} | I_j] \in \mathbb{R}^{1 \times (n+1)}$$

$$x_j = [x_j^1, x_j^2, \dots, x_j^m] \in \mathbb{R}^{1 \times m}$$

Ecuația (9.5) poate fi scrisă pentru fiecare celulă  $C_j$  sub forma:

$$T \cdot w_j^T = x_j^T \quad j = 1, 2, \dots, n \quad (9.6)$$

Se introduce o matrice pătratică, matricea index, ca în [60],  $S = [S_{ij}] \in \mathbb{R}^{n \times n}$ , astfel:

$$S_{ij} = \begin{cases} 1 & \text{daca } j \geq i \text{ si } j \text{ apartine vecinatatii celulei } i \\ 0 & \text{in rest} \end{cases} \quad (9.7)$$

Matricea index  $S$  permite obținerea unei matrici  $T$  reduse,  $T_{rj} \in R^{m \times h_j}$  respectiv a vectorului  $w$  redus,  $w_{rj} \in R^{1 \times h_j}$ , obținute prin eliminarea coloanelor al căror indice corespunde elementelor nule din rândul  $j$  al matricii  $S$ , cu :

$$h_j = 1 + \sum_{k=1}^n S_{jk}$$

Pentru prima celulă, ecuația (9.6) devine:

$$T_{r1} \cdot w_{r1}^T = x_1^T \quad \text{ptr. } j = 1, 2, \dots, n \quad (9.8)$$

Ecuația (9.8) are soluția:

$$w_{r1}^T = T_{r1}^+ \cdot x_1^T \quad (9.9)$$

unde  $T_{r1}^+$  este pseudoinversa matricii  $T_{r1}$  (a se vedea Anexa 4).

Expandând în pozițiile adecvate  $w_{rj}$  cu elemente nule se obține vectorul ponderilor interconexiunilor  $w_j$ .

### Observație

Metoda propusă se bazează pe determinarea pseudoinversei matricii modelelor de memorat și păstrează arhitectura RNCD. Transformarea realizată, între intrare și ieșire, este transformarea optimală, în sensul celei mai mici erori și asigură performanțe bune comparativ cu alte proceduri de sinteză.

## 9.2 Memorie asociativă discretă având capacitatea de a învăța și uita modele

Procedura de sinteză propusă pentru RNCD în [19] permite învățarea de modele noi de intrare și uitarea unor modele memorate, fără a fi necesară reprocesarea întregii matrici de interconexiune. Abilitatea de a învăța și uita modele, fără a afecta celelalte puncte de echilibru, este o caracteristică importantă a rețelelor neuronale, demonstrată până acum doar de memoriile asociative ce utilizează regula lui Hebb.

### 9.2.1 Memorie asociativă discretă cu capacitatea de învățare

Fie sistemul descris de ecuațiile (9.1) și (9.2).

Odată proiectată memoria asociativă, prin procedura de sinteză dată în § 9.2.1, se dorește învățarea unui nou model  $y^{m+1}$  corespunzător punctului de echilibru  $x^{m+1}$ .

Într-un mod similar procedurii anterior expuse, se definesc matricile  $X_{m+1}$ ,  $Y_{m+1}$ ,  $I_{m+1}$ ,  $T_{m+1}$  respectiv se introduce vectorul  $a$ :

Matricea index S permite obținerea unei matrici T reduse,  $T_{rj} \in R^{m \times h_j}$  respectiv a vectorului w redus,  $w_{rj} \in R^{1 \times h_j}$ , obținute prin eliminarea coloanelor al căror indice corespunde elementelor nule din rândul j al matricii S, cu :

$$h_j = 1 + \sum_{k=1}^n S_{jk}$$

Pentru prima celulă, ecuația (9.6) devine:

$$T_{r1} \cdot w_{r1}^T = x_1^T \quad \text{ptr. } j = 1, 2, \dots, n \quad (9.8)$$

Ecuația (9.8) are soluția:

$$w_{r1}^T = T_{r1}^+ \cdot x_1^T \quad (9.9)$$

unde  $T_{r1}^+$  este pseudoinversa matricii  $T_{r1}$  (a se vedea Anexa 4).

Expandând în pozițiile adecvate  $w_{rj}$  cu elemente nule se obține vectorul ponderilor interconexiunilor  $w_j$ .

### Observație

Metoda propusă se bazează pe determinarea pseudoinversei matricii modelelor de memorat și păstrează arhitectura RNCD. Transformarea realizată, între intrare și ieșire, este transformarea optimală, în sensul celei mai mici erori și asigură performanțe bune comparativ cu alte proceduri de sinteză.

## 9.2 Memorie asociativă discretă având capacitatea de a învăța și uita modele

Procedura de sinteză propusă pentru RNCD în [19] permite învățarea de modele noi de intrare și uitarea unor modele memorate, fără a fi necesară reprocesarea întregii matrici de interconexiune. Abilitatea de a învăța și uita modele, fără a afecta celelalte puncte de echilibru, este o caracteristică importantă a rețelelor neuronale, demonstrată până acum doar de memoriile asociative ce utilizează regula lui Hebb.

### 9.2.1 Memorie asociativă discretă cu capacitatea de învățare

Fie sistemul descris de ecuațiile (9.1) și (9.2).

Odată proiectată memoria asociativă, prin procedura de sinteză dată în § 9.2.1, se dorește învățarea unui nou model  $y^{m+1}$  corespunzător punctului de echilibru  $x^{m+1}$ .

Într-un mod similar procedurii anterior expuse, se definesc matricile  $X_{m+1}, Y_{m+1}, I_{m+1}, T_{m+1}$  respectiv se introduce vectorul a:

$$\mathbf{X}_{m+1} = [\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m, \mathbf{x}^{m+1}] \in \mathbf{R}^{n \times (m+1)}$$

$$\mathbf{Y}_{m+1} = [y^1, y^2, \dots, y^m, y^{m+1}] \in \mathbf{R}^{n \times (m+1)}$$

$$\mathbf{I}_{m+1} = [\mathbf{I}, \mathbf{I}, \dots, \mathbf{I}] \in \mathbf{R}^{n \times (m+1)}$$

$$\mathbf{J}' = [1, 1, \dots, 1]^T \in \mathbf{R}^{(m+1) \times 1}$$

$$\mathbf{T}_{m+1} = [\mathbf{Y}_{m+1}^T | \mathbf{J}'] \in \mathbf{R}^{(m+1) \times (n+1)}$$

$$\mathbf{a} = [(y^{m+1})^T | 1] \in \mathbf{R}^{1 \times (n+1)}$$

Fiecare vector  $y^i$  corespunde unui punct de echilibru, dacă, și numai dacă, este îndeplinită condiția (9.2) și în plus, pentru  $y^{m+1}$  este satisfăcută relația:

$$\mathbf{x}^{m+1} = \mathbf{A} \cdot \mathbf{y}^{m+1} + \mathbf{I} \quad (9.10)$$

Ecuatia matricială care trebuie soluționată este:

$$\mathbf{T}_{m+1} \cdot \mathbf{w}_{m+1,j}^T = \mathbf{x}_{m+1,j}^T \quad j = 1, 2, \dots, n \quad (9.11)$$

unde:

- $\mathbf{w}_{m+1,j} = [\mathbf{A}_{j1}, \mathbf{A}_{j2}, \dots, \mathbf{A}_{jn} | \mathbf{I}_j] \in \mathbf{R}^{1 \times (n+1)}$  este vectorul necunoscut, al ponderilor
- $\mathbf{x}_{m+1,j} = [\mathbf{x}_j^1, \mathbf{x}_j^2, \dots, \mathbf{x}_j^m, \mathbf{x}_j^{m+1}] \in \mathbf{R}^{1 \times (m+1)}$

Fie matricea redusă a lui  $\mathbf{T}_{m+1,j}, \mathbf{T}_{m+1,r,j} \in \mathbf{R}^{(m+1) \times h_j}$ , respectiv vectorul redus al lui  $\mathbf{w}_{m+1,j}, \mathbf{w}_{m+1,r,j} \in \mathbf{R}^{1 \times h_j}$ , obținute prin eliminarea coloanelor al căror indice corespunde elementelor nule din rândul  $j$  al matricii index  $S$  (introduse prin (9.7)).

Se calculează pseudoinversa matricii  $\mathbf{T}_{m+1,r,j}$ , utilizând informația referitoare la pseudoinversa matricii  $\mathbf{T}_{m,r,j}$ :

$$\mathbf{T}_{m+1,r,j} = [\mathbf{T}_{m,r,j}^T | \mathbf{a}_j^T]^T \quad (9.12)$$

unde  $\mathbf{a}_j$  este un subvector, de dimensiune  $(1 \times h_j)$ , al lui  $\mathbf{a}$ .

Se poate demonstra [19] că:

$$\mathbf{T}_{m+1,r,j}^{-1} = [\mathbf{T}_{m,r,j}^{-1} - \mathbf{b}_j^T \cdot \mathbf{a}_j \cdot \mathbf{T}_{m,r,j}^{-1} | \mathbf{b}_j^T] \quad (9.13)$$

unde  $\mathbf{b}_j$  este un vector unidimensional, de dimensiune  $(1 \times h_j)$ , ales astfel încât, este îndeplinită relația:

$$\mathbf{b}_j^T = \begin{cases} (\mathbf{a}_j - \mathbf{a}_j \cdot \mathbf{T}_{m,r,j}^{-1} \cdot \mathbf{T}_{m,r,j})^{-1} & \text{daca } [\mathbf{I}_b - \mathbf{T}_{m,r,j}^{-1} \cdot (\mathbf{T}_{m,r,j}^{-1})^{-1} \cdot \mathbf{a}_j^T] \neq 0 \\ (\mathbf{T}_{m,r,j}^{-1} \cdot \mathbf{T}_{m,r,j})^{-1} \cdot \mathbf{a}_j^T [1 + \mathbf{a}_j \cdot (\mathbf{T}_{m,r,j}^{-1} \cdot \mathbf{T}_{m,r,j})^{-1} \cdot \mathbf{a}_j^T]^{-1} & \text{in rest} \end{cases} \quad (9.14)$$

unde  $I_h$  este o matrice identitate, de dimensiune  $(h_j \times h_j)$ .

Din ecuația (9.11) se obține:

$$\mathbf{w}_{m+1,r,j}^T = \mathbf{T}_{m+1,r,j}^+ \cdot \mathbf{x}_{m+1,j}^T \quad j = 1, 2, \dots, n \quad (9.15)$$

### Procedura de sinteză :

Pentru fiecare  $j=1, 2, \dots, n$

1. Se procesează vectorul  $\mathbf{a}_j$ ;
2. Se determină vectorul  $\mathbf{b}_j$  conform relației (9.14);
3. Se calculează matricea  $\mathbf{T}_{m+1,r,j}^+$  cu relația (9.12);
4. Se determină vectorii necunoscuți ai ponderilor  $\mathbf{w}_{m+1,r,j}^T$ ;
5. Se completează adecvat cu elementele nule vectorul  $\mathbf{w}_{m+1,r,j}^T$  până când se obține vectorul  $\mathbf{w}_{m+1,j}^T$ ;

## 9.2.2 Memorie asociativă discretă cu capacitatea de uitare

Fie matricile  $\mathbf{X}_m^s$ ,  $\mathbf{Y}_m^s$  și  $\mathbf{T}_m^s$  obținute din matricile  $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $\mathbf{T}$  prin interschimbarea poziției vectorilor  $\mathbf{x}^i$ ,  $\mathbf{y}^i$  cu vectorii  $\mathbf{x}^m$  respectiv  $\mathbf{y}^m$ , adică:

$$\mathbf{X}_m^s = [\mathbf{x}^1, \dots, \mathbf{x}^{i-1}, \mathbf{x}^m, \mathbf{x}^{i+1}, \dots, \mathbf{x}^{m-1}, \mathbf{x}^i] \in \mathbf{R}^{n \times m}$$

$$\mathbf{Y}_m^s = [\mathbf{y}^1, \dots, \mathbf{y}^{i-1}, \mathbf{y}^m, \mathbf{y}^{i+1}, \dots, \mathbf{y}^{m-1}, \mathbf{y}^i] \in \mathbf{R}^{n \times m}$$

$$\mathbf{X}_{m-1}^s = [\mathbf{x}^1, \dots, \mathbf{x}^{i-1}, \mathbf{x}^m, \mathbf{x}^{i+1}, \dots, \mathbf{x}^{m-1}] \in \mathbf{R}^{n \times (m-1)}$$

Prin eliminarea vectorilor  $\mathbf{x}^i$ ,  $\mathbf{y}^i$  din matricile de mai sus se obțin matricile:

$$\mathbf{Y}_{m-1}^s = [\mathbf{y}^1, \dots, \mathbf{y}^{i-1}, \mathbf{y}^m, \mathbf{y}^{i+1}, \dots, \mathbf{y}^{m-1}] \in \mathbf{R}^{n \times (m-1)}$$

$$\mathbf{T}_m^s = [(\mathbf{Y}_m^s)^T | \mathbf{J}] \in \mathbf{R}^{m \times (n+1)}$$

$$\mathbf{T}_{m-1}^s = [(\mathbf{Y}_{m-1}^s)^T | \mathbf{J}] \in \mathbf{R}^{(m-1) \times (n+1)}$$

$$\mathbf{d} = [(\mathbf{y}^i)^T | \mathbf{1}] \in \mathbf{R}^{1 \times (n+1)}$$

Prin eliminarea elementelor cu același indice ca și cel al elementelor nule din rândul  $j$  al matricii index  $S$ , din matricile  $\mathbf{T}_m^s$  și  $\mathbf{T}_{m-1}^s$ , se obțin restricțiile corespunzătoare  $\mathbf{T}_{m,r,j}^s$  și  $\mathbf{T}_{m-1,r,j}^s$ . În mod similar, se obțin vectorii de restricție  $\mathbf{d}_j$  și  $\mathbf{w}_{m-1,r,j}$ .

Se poate demonstra că:

$$\mathbf{T}_{m,r,j}^s = [(\mathbf{T}_{m-1,r,j}^s)^T | \mathbf{d}_j^T]^T \quad (9.16)$$

- unde  $\mathbf{d}_j$  este un subvector, de dimensiune  $(1 \times h_j)$ , al lui  $\mathbf{d}$ .



Matricea  $(\mathbf{T}_{m,r,j}^s)^+$  poate fi exprimată în funcție de ultima coloană  $\mathbf{q}_j$  astfel:

$$(\mathbf{T}_{m,r,j}^s)^+ = [\mathbf{P}_j | \mathbf{q}_j] \quad (9.17)$$

Utilizând proprietățile pseudoinversei se poate scrie relația:

$$(\mathbf{T}_{m,r,j}^s)^+ = [(\mathbf{T}_{m-1,r,j}^s)^+ - \mathbf{c}_j^T \cdot \mathbf{d}_j \cdot (\mathbf{T}_{m-1,r,j}^s)^+ | \mathbf{c}_j^T] \quad (9.18)$$

- unde  $\mathbf{c}_j$  este un vector unidimensional ( $1 \times h_j$ ) ales astfel încât este îndeplinită relația:

$$\mathbf{c}_j^T = \begin{cases} [\mathbf{d}_j - \mathbf{d}_j \cdot (\mathbf{T}_{m-1,r,j}^s)^+ \cdot \mathbf{T}_{m,r,j}^s]^+ & \text{daca } \left\{ \mathbf{I}_h - (\mathbf{T}_{m-1,r,j}^s)^T \cdot [(\mathbf{T}_{m,r,j}^s)^+]^+ \right\} \mathbf{d}_j^T \neq \mathbf{0} \\ [(\mathbf{T}_{m-1,r,j}^s)^T \cdot \mathbf{T}_{m-1,r,j}^s]^{-1} \cdot \mathbf{d}_j^T \cdot \left\{ \mathbf{1} + \mathbf{d}_j \cdot [(\mathbf{T}_{m,r,j}^s)^T \cdot \mathbf{T}_{m-1,r,j}^s]^{-1} \cdot \mathbf{d}_j^T \right\}^{-1} & \text{in rest} \end{cases} \quad (9.19)$$

- unde  $\mathbf{I}_h$  este o matrice identitate, de dimensiune  $(h_j \times h_j)$ .

Comparând relația (9.17) cu (9.18) se obține:

$$\mathbf{P}_j = (\mathbf{T}_{m-1,r,j}^s)^+ - \mathbf{c}_j^T \cdot \mathbf{d}_j \cdot (\mathbf{T}_{m-1,r,j}^s)^+ \quad (9.20)$$

$$\mathbf{q}_j = \mathbf{c}_j^T \quad (9.21)$$

Din ecuațiile (9.20), (9.21) rezultă:

$$(\mathbf{T}_{m-1,r,j}^s)^+ = (\mathbf{I}_h - \mathbf{q}_j \cdot \mathbf{d}_j)^{-1} \cdot \mathbf{P}_j \quad (9.22)$$

Deci, se poate determina soluția sistemului:

$$\mathbf{w}_{m-1,r,j}^T = (\mathbf{T}_{m-1,r,j}^s)^+ \cdot (\mathbf{X}_{m-1,r,j}^s)^T \quad j = 1, 2, \dots, n \quad (9.23)$$

Pseudoinversa  $(\mathbf{T}_{m,r,j}^s)^+$  se poate obține prin simpla interschimbare a coloanei  $i$  cu coloana  $m$ , în matricea pseudoinversă  $\mathbf{T}_{m,r,j}^s$ , ținând cont de următoarea proprietate:

$$(\mathbf{T}_{m,r,j}^s)^+ = (\mathbf{T}_{m,r,j}^s)^s$$

**Procedura de sinteză :**

Pentru  $j=1, 2, \dots, n$

1. Se determină din  $(\mathbf{T}_{m,r,j}^s)^+$  din  $\mathbf{T}_{m,r,j}^s$  ;
2. Se calculează cu relațiile (9.22) și (9.23)  $\mathbf{P}_j$  și  $\mathbf{q}_j$  din  $(\mathbf{T}_{m,r,j}^s)^+$  ;
3. Se procesează  $\mathbf{d}_j$ ;
4. Se determină  $(\mathbf{T}_{m-1,r,j}^s)^+$  cu relația (9.24);
5. Se calculează vectorii necunoscuți ai ponderilor  $\mathbf{w}_{m-1,r,j}^T$  ;
6. Se completează adecvat cu elemente nule vectorul  $\mathbf{w}_{m-1,r,j}^T$  până când se obține vectorul  $\mathbf{w}_{m-1,j}^T$  ;

### Observație

Rețeaua neuronală a fost înzestrată cu capacitatea de învățare și uitare fără a afecta echilibrul existent și menținând localitatea interconexiunilor.

## 9.3 Memorie asociativă complet stabilă

Metoda propusă, bazată tot pe determinarea pseudoinversei, asigură în plus condiția de simetrie a matricii de interconexiune, deci stabilitatea completă a rețelei neuronale.

Se consideră RNC descrisă de următoarea ecuație :

$$\begin{aligned}\dot{x} &= -C \cdot x + A \cdot f(x) + I \\ y &= f(x)\end{aligned}\tag{9.24}$$

S-a demonstrat că dacă matricea de reacție inversă este simetrică, RNC este complet stabilă [21],[27].

Fiind date  $m$  modele de memorat  $y^i$ ,  $i=1,2,\dots,m$  și o rețea neuronală celulară cu  $n$  celule trebuie îndeplinite condițiile:

- fiecare vector de memorie  $x$  trebuie să fie un punct asimptotic stabil;
- rețeaua neuronală trebuie să fie complet stabilă, deci  $A$  simetrică;

Relațiile (9.3)-(9.9) sunt valabile și pentru sistemul continuu (9.24).

Pentru obținerea unei matrici de reacție  $A$  simetrice trebuie îndeplinită în plus condiția  $T_{kj}=T_{jk}$ , astfel încât din ecuația (9.6) se obține:

$$T_{rj} \cdot w_{rj}^T = x_j^T - \sum_{k=1}^{j-1} y_k^T \cdot A_{jk} \quad \text{ptr. } j = 1, 2, \dots, n\tag{9.25}$$

Ecuația (9.25) se poate rezolva determinând  $T_{rj}^+$  pentru  $j=1,2,\dots,n$  și luând în considerare rândul  $k$  al matricii  $Y$ ,  $y_k \in \mathbb{R}^{1 \times m}$ .

Utilizând matricea index  $S$ , sunt procesate doar elementele nenule ale matricii de reacție. În plus, datorită structurii matricii  $S$ , toate coloanele matricii  $T$  și ale vectorului  $j$  sunt eliminate până la  $j-1$  și în timpul evaluării interconexiunilor  $T_{jk}, k=1,2,\dots,j-1$  este îndeplinită și condiția de simetrie. Procedura de sinteză expandează adecvat  $w_{rj}^T$  cu elemente nule și cu interconexiuni simetrice  $A_{jk}, k=1,2,\dots,j-1$ , pentru obținerea lui  $w_j$ .

## 9.4 Memorii asociative de ordin superior ce utilizează metoda pseudoinversei

Rețelele cu conexiuni de ordinul întâi au limitări intrinseci în ceea ce privește capacitatea de memorare. Deși complexitatea proiectării și implementării crește, în cazul conexiunilor de ordin superior, avantajul îmbunătățirii capacității de memorare poate fi esențial în unele aplicații.

Metoda de sinteză expusă [22] permite proiectarea unei MA cu interconexiuni de ordinul întâi și superior, exploatând proprietățile pseudoinversei.

Se introduce următorul model pentru o RNC discretă în timp, cu interconexiuni de ordinul întâi și superior:

$$x_i[k+1] = \sum_{j=1}^n A_{ij} y_j[k] + \sum_{j=1}^{n-1} \sum_{l=j+1}^n H_{ijl} \cdot y_j[k] \cdot y_l[k] + I_i$$

ptr.  $i = 1, 2, \dots, n$  (9.26)

$$y_i[k] = f(x[k]) = \frac{1}{2}(|x_i[k] + 1| - |x_i[k] - 1|)$$

unde:

- $A_{ij}$  este interconexiunea de reacție, de ordinul întâi, de la celula  $C_j$  la celula  $C_i$ , într-o vecinătate finită de rază  $r$ ;
- $H_{ijl}$  este interconexiunea de reacție, de ordinul doi, de la celula  $C_j$  la celula  $C_l$  din vecinătatea predefinită a celulei  $C_i$ ;

Ecuția (9.26) poate fi scrisă în forma vectorială următoare:

$$\begin{aligned} x[k+1] &= -A \cdot y[k] + H \cdot z[k] + I \\ y[k] &= f(x[k]) \end{aligned}$$

(9.27)

unde:

- $z = [y_1 \cdot y_2, y_1 \cdot y_3, \dots, y_1 \cdot y_n, y_2 \cdot y_3, y_2 \cdot y_4, \dots, y_2 \cdot y_n, \dots, y_{n-1} \cdot y_n]^T \in R^d$  este vectorul produselor ieșirilor celulelor RNC, cu  $d = n \cdot (n-1) / 2$ ; În cele ce urmează  $z$  va fi reprezentat ca  $z = [z_1, z_2, \dots, z_d]^T$  cu  $z_1 = z_1(y_1 \cdot y_2) = f(x_1) f(x_2)$ ;
- $A = [A_{ij}] \in R^{n \times n}$  este matricea interconexiunilor de reacție de ordinul întâi;
- $H = [H_{ijl}] \in R^{n \times d}$  este matricea interconexiunilor de reacție de ordinul doi;

Matricile  $A$  și  $H$  sunt matrici rare care țin cont de arhitectura locală a RNC, fiind dependente de raza  $r$  a vecinătății. Pentru o vecinătate  $r=1$  există 9 interconexiuni de ordinul întâi și  $d=9 \cdot 8 / 2 = 36$  de ordinul doi. În general pentru o vecinătate de rază  $r$  care conține  $q$  celule numărul interconexiunilor de reacție de ordinul doi este  $d=q \cdot (q-1) / 2$ .

Pentru analiza stabilității modelului introdus se rescrie ecuația (9.27) sub forma:

$$x[k+1] = A' \cdot f'(x[k]) + I$$

(9.28)

unde:

- $A^* = [A|H] \in \mathbb{R}^{n \times (n+d)}$  este matricea interconexiunilor de ordinul întâi și doi obținută prin extinderea matricii A cu matricea H;

- $f^*(x[k]) = \begin{bmatrix} f(x[k]) \\ z(x[k]) \end{bmatrix} \in \mathbb{R}^{(n+d)}$ ;

Orice punct  $x_0 \in \mathbb{R}^n$  este un punct de echilibru al sistemului dinamic (9.28) dacă:

$$x_0 = A^* \cdot f^*(x[k]) + I \quad (9.29)$$

Analiza stabilității se face în [22].

Procedura de sinteză descrisă în continuare se bazează pe liniarizarea problemei, derivată din alegerea stării corespunzătoare punctului de echilibru dorit,  $x_{0i} = \pm 1$  pentru  $f(x_{0i}) = \pm 1, i=1,2,\dots,n$ .

Se dorește determinarea matricilor de interconexiune **A**, **H**, **I**, ale unei RNC cu n celule, astfel încât, fiind date cele m modele de memorat  $y^i, i=1,2,\dots,m$ , trebuie îndeplinite condițiile:

- fiecare model de memorat  $y^i, i=1,2,\dots,m$  corespunde unui punct de echilibru  $x^i$  al rețelei;
- fiecare punct de echilibru  $x^i, i=1,2,\dots,m$  trebuie să fie un punct asimptotic stabil;

### Teorema 1

Fiecare vector  $y^i$  corespunde unui punct de echilibru, dacă, și numai dacă, este îndeplinită condiția:

$$x^i = A \cdot y^i + H \cdot z^i + I \quad (9.30)$$

unde:

- $x^i = [x_1^i, x_2^i, \dots, x_n^i]^T \in \mathbb{R}^n$
- $y^i = [y_1^i, y_2^i, \dots, y_n^i]^T \in \mathbb{R}^n$
- $z^i = [z_1^i, z_2^i, \dots, z_n^i]^T \in \mathbb{R}^d$

Procedura de sinteză asigură îndeplinirea condiției (1). Constrângerea (2) poate fi satisfăcută [22], impunând ca  $x^i = f(x^i) = y^i$ , adică  $y_j^i = x_j^i = \pm 1$ , pentru  $i=1,2,\dots,m$  și  $j=1,2,\dots,n$ .

Omitând pentru simplitate pasul de iterație, ecuația (9.27) poate fi scrisă compact sub forma:

$$X = A \cdot Y + H \cdot Z + I \quad (9.31)$$

unde:

- $\mathbf{X} = [x^1, x^2, \dots, x^m] \in \mathbf{R}^{n \times m}$
- $\mathbf{Y} = [y^1, y^2, \dots, y^m] \in \mathbf{R}^{n \times m}$
- $\mathbf{Z} = [z^1, z^2, \dots, z^m] \in \mathbf{R}^{d \times m}$
- $\mathbf{I}' = [I, I, \dots, I] \in \mathbf{R}^{n \times m}$

Se introduc următoarele matrici:

$$\mathbf{T} = [\mathbf{Y}^T | \mathbf{Z}^T | \mathbf{J}] \in \mathbf{R}^{m \times (n+d+1)}$$

$$\mathbf{J} = [1, 1, \dots, 1]^T \in \mathbf{R}^{m \times 1}$$

$$\mathbf{w}_j = [A_{j1}, A_{j2}, \dots, A_{jn} | C_{j1}, C_{j2}, \dots, C_{jd} | I_j] \in \mathbf{R}^{1 \times (n+d+1)}$$

Ecuția (9.31) devine:

$$\mathbf{T} \cdot \mathbf{w}_j^T = \mathbf{X}_j^T \quad j = 1, 2, \dots, n \quad (9.32)$$

Ecuția (9.32) trebuie soluționată ținând cont de localitatea conexiunilor RNC.

Se introduc două matrici pătratice,  $\mathbf{S} = [S_{ij}] \in \mathbf{R}^{n \times n}$  și  $\mathbf{S}' = [S'_{iq}] \in \mathbf{R}^{n \times n}$  astfel:

$$S_{ij} = \begin{cases} 1 & \text{daca } j \text{ apartine vecinatatii celulei } i \\ 0 & \text{in rest} \end{cases}$$

$$S'_{iq} = \begin{cases} 1 & \text{daca ambele celule referite de indicele } q \text{ apartin vecinatatii celulei } i \\ 0 & \text{in rest} \end{cases}$$

Matricile  $\mathbf{S}$  și  $\mathbf{S}'$  permit obținerea unei matrici  $\mathbf{T}$  reduse,  $\mathbf{T}_{rj} \in \mathbf{R}^{m \times h_i}$ , obținută prin eliminarea coloanelor:

- al căror indice corespunde elementelor nule din rândul  $j$  al matricii  $\mathbf{S}$ ;
- al căror indice corespunde elementelor nule din rândul  $j$  al matricii  $\mathbf{S}'$ ;

$$\mathbf{h}_j = \mathbf{1} + \sum_{i=1}^n S_{ij} + \sum_{q=1}^d S'_{iq}$$

Într-un mod similar se determină vectorul redus  $\mathbf{w}_{rj} \in \mathbf{R}^{1 \times h_i}$ , prin eliminarea coloanelor corespunzătoare din  $\mathbf{w}_j$ .

Ecuția (9.32) are soluția:

$$\mathbf{w}_{rj}^T = \mathbf{T}_{rj}^+ \cdot \mathbf{X}_j^T \quad (9.33)$$

unde  $\mathbf{T}_{rj}^+$  este pseudoinversa matricii  $\mathbf{T}_{rj}$ .

Expandând în pozițiile adecvate  $\mathbf{w}_{rj}$  cu elemente nule se obține vectorul ponderilor interconexiunilor  $\mathbf{w}_j$ .

### Procedura de sinteză :

Pentru fiecare  $j=1,2,\dots,n$

1. Se alege  $X=Y$ ;
2. Se determină matricea  $S$ ;
3. Se calculează matricea  $S'$ ;
4. Se procesează matricea  $T_{r,j}^+$ ;
5. Se determină vectorii necunoscuți ai ponderilor  $w_{r,j}^T$ ;
6. Se completează adecvat cu elemente nule vectorul  $w_{r,j}^T$  până când se obține vectorul  $w_j^T$ ;

### Observație

Metoda propusă se bazează pe determinarea pseudoinversei matricii modelelor de memorat și păstrează arhitectura RNC. Metoda propusă este flexibilă. Ea este o generalizare a metodei 9.1, care se poate obține dacă se consideră  $S'=0$  și  $H=0$ . Dacă se dorește implementarea unei RNC de ordinul 2 se consideră  $S=0$  și  $A=0$ . Prin încorporarea procedurilor 9.2 și 9.3 se pot aduce îmbunătățiri ulterioare metodei, obținându-se o memorie cu capacitatea de învățare și de uitare. Transformarea realizată, între intrare și ieșire, este transformarea optimală, în sensul celei mai mici erori și asigură performanțe bune comparativ cu alte proceduri de sinteză.

## 9.5 Rezultatele simulării

Pentru a testa performanțele memoriei obținute prin metoda pseudoinversei am făcut mai multe experimente.

Algoritmul a fost implementat în MATLAB. Matricile de interconexiune obținute pe arhitectura unei RN de tip complet conectate, conform procedurilor expuse au fost modificate, în Borland C, într-o formă adecvată simulatorului soft RNC, variant în spațiu.

### Exemplul 1

Au fost memorate utilizând metoda pseudoinversei din §9.1, cu o RNC cu vecinătate  $r=1$ , primele trei modele ideale din Fig.9.1 (literele N,E,T), de dimensiune  $6 \times 6$ , aceleași din [18]. Testele au fost efectuate atât pentru zgomot binar (obținut prin inversarea valorii pixelilor în literele N,E și T din Fig.9.1) cât și pentru zgomot gaussian (primele trei litere din Fig.9.2), cu medie nulă și dispersie  $\sigma \in (0, 1.5)$ .

Tabelul 9.1

Zgomot / vecinătate	Vecinătate r=1
gaussian $\sigma=1,4$	toate cele 3 modele corect apelate din memorie
gaussian $\sigma=1,5$	un model corect apelat din memorie, celelalte două cu 2 pixeli eroare
binar 22% (8 pixeli inversați)	toate cele 3 modele corect apelate din memorie

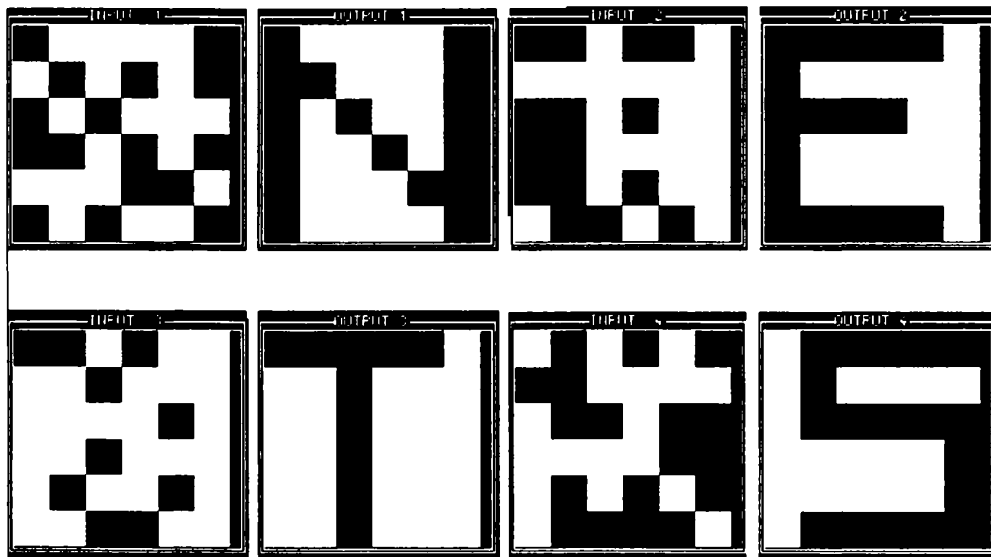


Fig. 9.1 Modele ideale și afectate de zgomot binar (8 pixeli eronați), memorate cu o RNC , proiectată prin metoda pseudoinversei

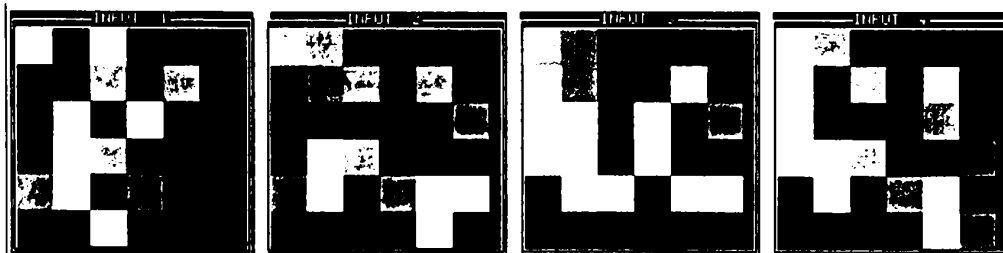


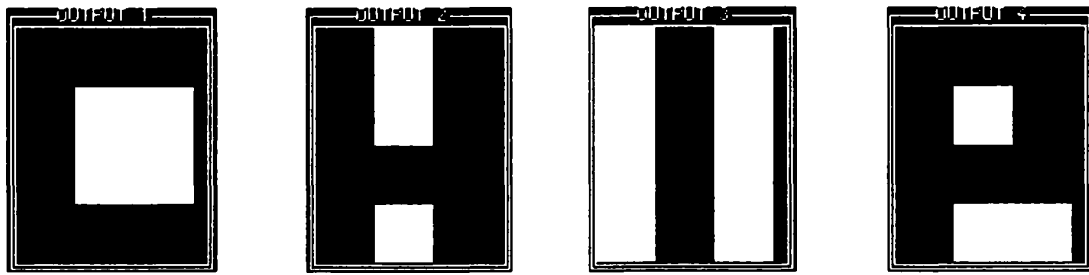
Fig. 9.2 Modele afectate de zgomot gaussian cu  $\sigma = 1,4$ , memorate cu o RNC, proiectată prin metoda pseudoinversei

Cele mai bune rezultate obținute sunt prezentate sintetic în Tabelul 9.1.

Tabelul 9.2

Zgomot / vecinătate	Vecinătate r=1
gaussian $\sigma=1,4$	toate cele 4 modele corect apelate din memorie
gaussian $\sigma=1,5$	2 modele apelate din memorie cu 1 pixel eroare, unul cu 3 pixeli eroare și unul cu 4 pixeli eroare
binar 22% (8 pixeli eronați)	toate cele 4 modele corect apelate din memorie

Reproiectând rețeaua au fost memorate prin aceeași metodă 4 modele. Rezultatele experimentului sunt prezentate sintetic în Tabelul 9.2.



**Fig. 9.3 Modele ideale memorate cu o RNC de dimensiune 4x3, proiectată prin metoda pseudoinversei**

Se poate constata că pe măsură ce numărul modelelor crește performanțele memoriei scad, ceea ce era de așteptat.

### Exemplul 2

Au fost memorate modelele din Fig.9.3, aceleași raportate și în [27] într-o RNC de dimensiune 4x3.

Matricea de interconexiune de reacție este:

0.000000000000e+00	0.000000000000e+00	0.000000000000e+00
0.000000000000e+00	1.333333373070e+00	1.875669758400e-17
0.000000000000e+00	1.333333373070e+00	-1.333333373070e+00
0.000000000000e+00	0.000000000000e+00	0.000000000000e+00
1.110087499354e-16	4.000000000000e+00	1.487210284394e-15
5.686019413657e-16	-6.697621862838e-16	-1.035196234289e-15
0.000000000000e+00	0.000000000000e+00	0.000000000000e+00
1.049290862531e-15	2.000000000000e+00	0.000000000000e+00
-2.000000000000e+00	1.912749472699e-15	0.000000000000e+00
0.000000000000e+00	1.000000000000e+00	5.507747036226e-17
0.000000000000e+00	1.000000000000e+00	-1.000000000000e+00
0.000000000000e+00	1.000000000000e+00	2.838441287567e-16
-7.407407164574e-01	-6.722053469410e-18	-7.407407164574e-01
-7.407407164574e-01	7.407407164574e-01	-1.481481492519e-01
-7.407407164574e-01	2.962962985039e-01	-1.481481492519e-01
4.770489558936e-16	3.333333432674e-01	0.000000000000e+00
-3.333333432674e-01	1.66666626930e+00	0.000000000000e+00
6.666666865349e-01	1.66666626930e+00	0.000000000000e+00
0.000000000000e+00	9.230769276619e-01	-9.230769276619e-01
0.000000000000e+00	9.230769276619e-01	-3.076923191547e-01
0.000000000000e+00	9.230769276619e-01	-3.076923191547e-01
-3.870967626572e-01	3.870967626572e-01	5.161290168762e-01
-3.870967626572e-01	2.451612949371e+00	5.161290168762e-01
-3.870967626572e-01	-5.161290168762e-01	5.815660453212e-16



-4.000000059605e-01	1.200000047684e+00	0.000000000000e+00
4.000000059605e-01	1.200000047684e+00	0.000000000000e+00
-1.200000047684e+00	-3.209238430557e-17	0.000000000000e+00
0.000000000000e+00	1.714285731316e+00	-5.714285969734e-01
0.000000000000e+00	1.714285731316e+00	-5.714285969734e-01
0.000000000000e+00	0.000000000000e+00	0.000000000000e+00
-3.333333432674e-01	-6.666666865349e-01	-1.666666626930e+00
-3.333333432674e-01	1.666666626930e+00	-3.035766082959e-17
0.000000000000e+00	0.000000000000e+00	0.000000000000e+00
-6.661338147751e-16	1.226883178385e-15	0.000000000000e+00
2.393918396848e-16	4.000000000000e+00	0.000000000000e+00
0.000000000000e+00	0.000000000000e+00	0.000000000000e+00

Matricea interconexiunilor de polarizare este:

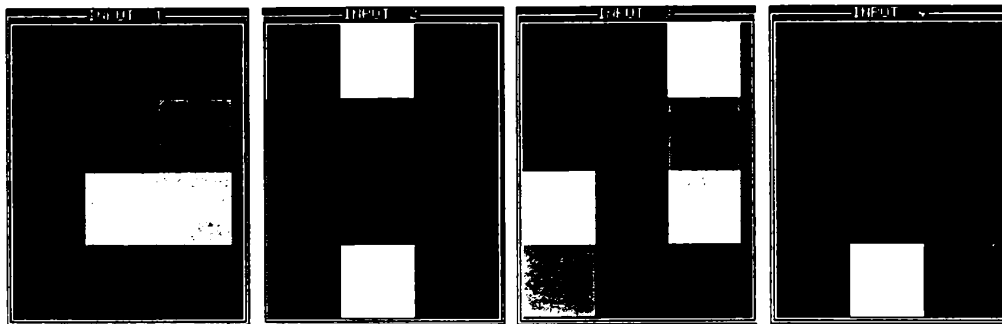
-5.5570782350744530e-016	-5.8069868358323620e-016	-1.7618285302889450e-016
-8.8817841970012520e-016	2.9629629629629600e-001	6.6666666666666610e-001
-3.0769230769230770e-001	-1.5483870967741930e+000	4.0000000000000080e-001
-5.7142857142857130e-001	-6.6666666666666700e-001	-4.2891037943526560e-016

Tabelul 9.3. prezintă rezultatele testelor efectuate. Se poate observa că această RNC de dimensiune 4x3 comparativ cu cea de dimensiune 6x6 este capabilă să corecteze un zgomot gaussian (cu medie nulă) cu o dispersie semnificativ mai mică. ( $\sigma=0,9$  comparativ cu  $\sigma=1,4$ ). Comportarea este evident mai puțin performantă și în cazul corecției zgomotului binar. MA RNC de dimensiune 4x3 ce utilizează metoda pseudoinversei nu este capabilă să corecteze 2 pixeli eronați.

Fig. 9.4 ilustrează modelele memorate peste care s-a suprapus un zgomot gaussian cu medie nulă și dispersie  $\sigma=0,9$ .

**Tabelul 9.3**

Zgomot / vecinătate	Vecinătate $r=1$
gaussian $\sigma=0,8$	toate cele 4 modele corect apelate din memorie
gaussian $\sigma=0,9$	toate cele 4 modele corect apelate din memorie
gaussian $\sigma=1$	toate cele 4 modele au fost apelate din memorie cu o eroare de 1 pixel
binar 2,8% (1 pixel eronat)	3 au fost corect apelate din memorie, unul a fost corect dependent de poziția pixelului eronat
binar 5,6% (2 pixeli eronați)	1 model uneori corect apelat din memorie, dependent de poziția pixelului eronat și trei cu 1 pixel eroare



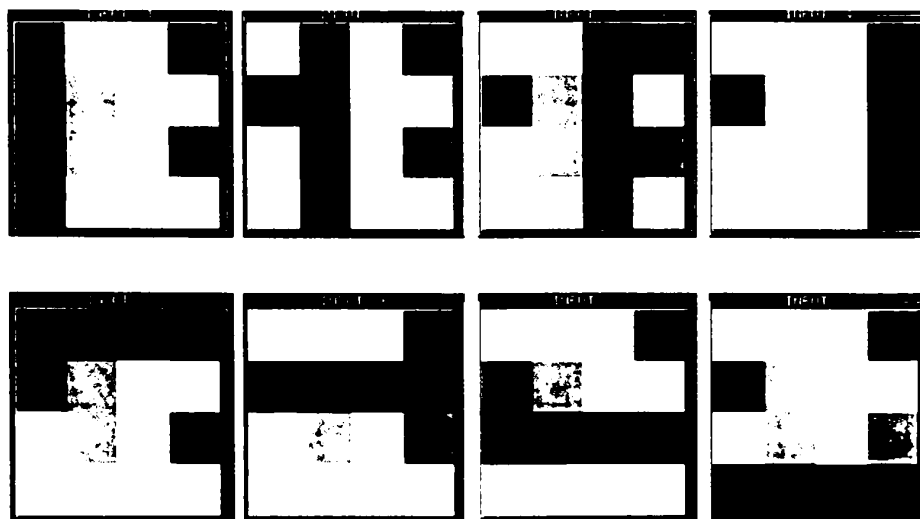
**Fig. 9.4** Modelele memorate afectate cu zgomot gaussian cu medie nulă și disperse  $\sigma = 0,9$

### Exemplul 3

S-a testat modul în care MA obținută prin metoda pseudoinversei apelează din memorie cele opt modele din Fig. 8.3. Fig. 9.5 prezintă modelele memorate afectate de un zgomot gaussian cu medie nulă și dispersia  $\sigma=0,6$ . Acestea aplicate la intrarea în RNC au generat la ieșire modelele ideale.

**Tabelul 9.4**

Zgomot / vecinătate	Vecinătate $r=1$
gaussian $\sigma=0,6$	toate cele 8 modele corect apelate din memorie
gaussian $\sigma=0,7$	2 modele corect apelate din memorie, restul cu 1 pixel eroare
binar 1 pixel eronat	toate cele 8 modele corect apelate din memorie
binar 2 pixeli eronați	toate modelele afectate cu 1 pixel eroare



**Fig. 9.5** Modelele memorate afectate de un zgomot gaussian cu medie nulă și dispersie  $\sigma=0,6$

Deși această MA are dimensiune mai mare decât cea din exemplul 2, capacitatea ei de corecție a zgomotului gaussian este mai slabă. Se poate deduce dependența performanței memoriei de modelele memorate.

Comparativ cu MA obținută prin metoda descompunerii în valori singulare care a corectat un zgomot gaussian cu  $\sigma=1$ , performanța acestei memorii este mai slabă. În ceea ce privește zgomotul binar ambele au corectat doar 1 pixel eroare.

#### Exemplul 4

Au fost testate cele 20 de modele din Fig. 8.5. Rezultatele experimentelor sunt sintetizate în tabelul 9.5.

**Tabelul 9.5**

Zgomot / vecinătate	Vecinătate $r=1$
gaussian $\sigma=0,6$	toate cele 20 modele corect apelate din memorie
gaussian $\sigma=0,7$	18 modele corect apelate din memorie, restul cu 1 pixel eroare
binar 1 pixel eronat	toate cele 20 modele corect apelate din memorie
binar 2 pixeli eronați	toate cele 20 modele corect apelate din memorie
binar 3 pixeli eronați	13 modele corect apelate, 7 cu 1 pixel eroare

Fig. 9.6 prezintă modelele memorate afectate cu un zgomot gaussian cu medie nulă și  $\sigma=0,6$ .

Se observă o îmbunătățire a performanțelor în ceea ce privește eliminarea zgomotului binar față de exemplul 2.

În comparație cu MA obținută prin metoda descompunerii în valori singulare, performanțele obținute prin metoda pseudoinversei sunt mai slabe. În primul caz a fost eliminat un zgomot gaussian cu medie nulă și  $\sigma=1$ , pe când aici doar  $\sigma=0,6$ . Și în cazul zgomotului binar, performanțele descompunerii în valori singulare au fost net superioare, fiind corecțate 7 pixeli inverși față de 2 aici.

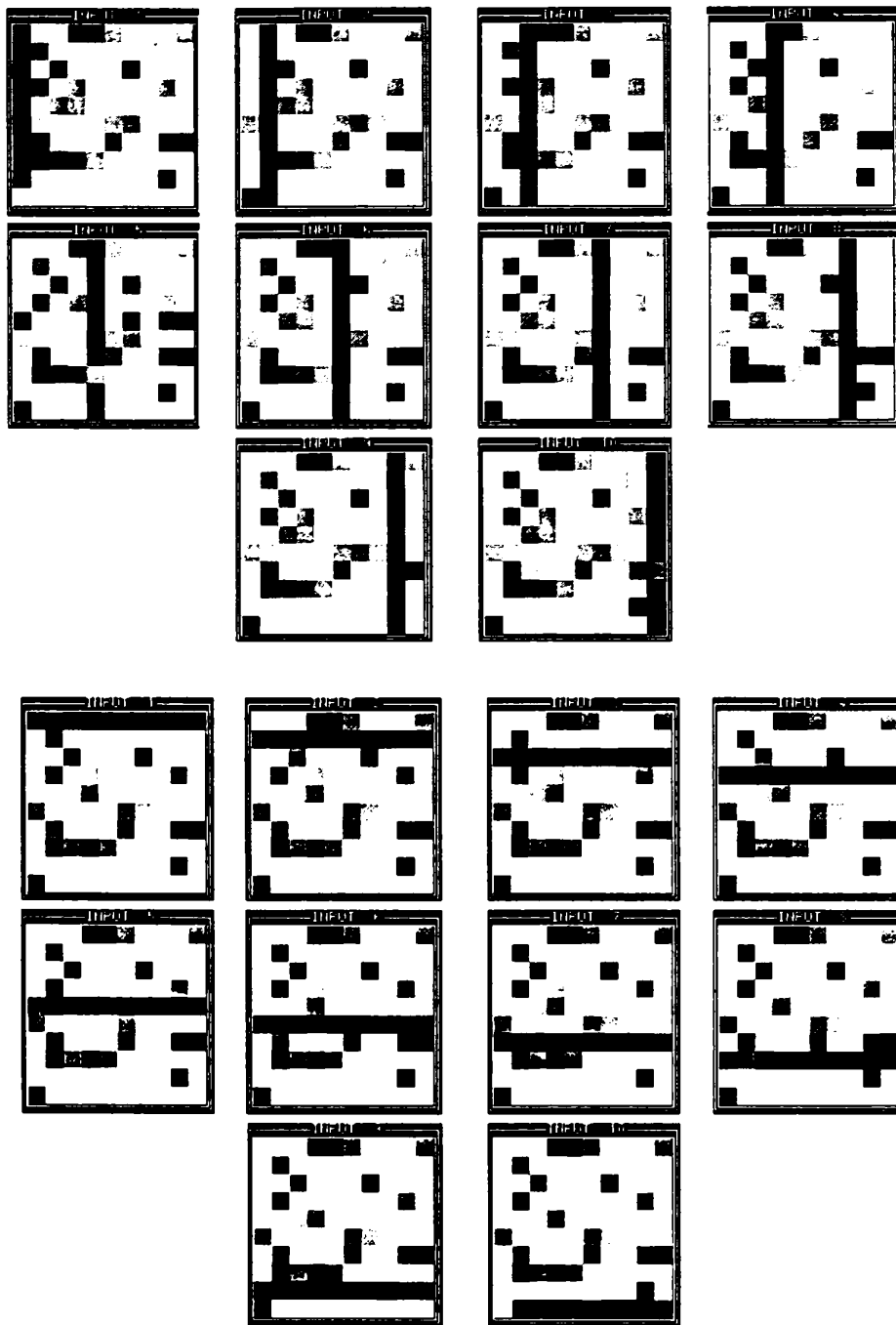


Fig. 9.6 Modelele memorate afectate cu un zgomot gaussian cu medie nulă și dispersie  $\sigma=0,6$

### Exemplul 5

S-a experimentat memorarea celor 25 de ideograme chinezești din Fig. 6.11.

S-au utilizat două tipuri de semnale aleatoare nedorite, suprapuse peste semnalul util și necorelate cu acesta: un zgomot Gaussian cu medie nulă și dispersie  $s \in [0,0, 0,6]$  și unul digital. Modelele afectate de zgomot gaussian cu  $\sigma=0,7$  sunt prezentate în Fig.6.14.

Rezultatele testelor sunt prezentate în Tabelul 9.6. Se poate remarca că odată cu creșterea numărului de modele memorate a scăzut semnificativ capacitatea de corecție. În plus nu a putut fi corectat zgomotul binar.

Tabelul 9.6

Zgomot / vecinătate	Vecinătate r=1
gaussian $\sigma=0,4$	25 modele corect apelate din memorie
gaussian $\sigma=0,5$	15 modele corect apelate din memorie, restul cu o eroare de 1 pixel
gaussian $\sigma=0,6$	toate modelele afectate cu o eroare de 1-5 pixeli
binar 1 pixel	toate modelele afectate cu o eroare de 1 pixel

**Exemplul 6**

Prin metoda 9.3 s-a obținut o memorie complet stabilă, de dimensiune 4x3, cu care au fost testate modelele din Fig.9.3, într-o MA RNC de dimensiune 4x3. Matricea de interconexiune de reacție sunt cele date în continuare, aceleași obținute și în [27].

Matricea de interconexiune de reacție este:

1.33333333333330e+000	1.8756697583999220e-017	0.000000000000000e+000
1.33333333333330e+000	-1.33333333333330e+000	0.000000000000000e+000
0.000000000000000e+000	0.000000000000000e+000	0.000000000000000e+000
0.000000000000000e+000	0.000000000000000e+000	0.000000000000000e+000
1.8756697583999220e-017	4.000000000000010e+000	7.3725747729014300e-018
7.3496625417084290e-018	-7.3767675960205670e-018	-2.8818093744664700e-016
0.000000000000000e+000	0.000000000000000e+000	0.000000000000000e+000
0.000000000000000e+000	0.000000000000000e+000	0.000000000000000e+000
0.000000000000000e+000	7.3725747729014300e-018	2.000000000000000e+000
0.000000000000000e+000	-2.000000000000000e+000	7.7325298941666170e-016
0.000000000000000e+000	0.000000000000000e+000	0.000000000000000e+000
0.000000000000000e+000	0.000000000000000e+000	0.000000000000000e+000
1.33333333333330e+000	7.3496625417084290e-018	0.000000000000000e+000
8.88888888888880e-001	-8.88888888888870e-001	0.000000000000000e+000
8.888888888888910e-001	-4.8463837110102040e-017	0.000000000000000e+000
0.000000000000000e+000	0.000000000000000e+000	0.000000000000000e+000
-1.33333333333330e+000	-7.3767675960205670e-018	-2.000000000000000e+000
-8.88888888888870e-001	-9.2592592592592260e-002	1.8518518518518460e-002
9.2592592592592280e-002	-3.7037037037036900e-002	1.8518518518518450e-002
0.000000000000000e+000	0.000000000000000e+000	0.000000000000000e+000
0.000000000000000e+000	-2.8818093744664700e-016	7.7325298941666170e-016
0.000000000000000e+000	1.8518518518518460e-002	2.0092592592592580e+000
0.000000000000000e+000	-1.8518518518517550e-002	2.0092592592592590e-000
0.000000000000000e+000	0.000000000000000e+000	0.000000000000000e-000
0.000000000000000e+000	0.000000000000000e+000	0.000000000000000e+000
8.888888888888910e-001	9.2592592592592280e-002	0.000000000000000e-000
1.3730158730158730e+000	-4.5767195767195730e-001	0.000000000000000e-000
1.3730158730158730e+000	-4.5767195767195770e-001	0.000000000000000e+000

0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
-4.8463837110102040e-017	-3.7037037037036900e-002	-1.8518518518517550e-002
-4.5767195767195730e-001	2.7396069538926700e+000	6.3945578231292530e-001
-8.3975812547241240e-001	-6.3945578231292500e-001	-6.8456525170734750e-016
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	1.8518518518518450e-002	2.0092592592592590e+000
0.0000000000000000e+000	6.3945578231292530e-001	8.4013605442176880e-001
0.0000000000000000e+000	-8.4013605442176880e-001	3.2898715041572180e-001
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
1.3730158730158730e+000	-8.3975812547241240e-001	0.0000000000000000e+000
1.7872260015117140e+000	-8.3975812547241240e-001	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
-4.5767195767195770e-001	-6.3945578231292500e-001	-8.4013605442176880e-001
-8.3975812547241240e-001	2.1914210128495840e+000	3.2898715041572240e-001
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	0.0000000000000000e+000	0.0000000000000000e+000
0.0000000000000000e+000	-6.8456525170734750e-016	3.2898715041572180e-001
0.0000000000000000e+000	3.2898715041572240e-001	4.0000000000000000e+000

Matricea de polarizare este:

-5.5570782350744530e-016	9.9529759434169300e-017	2.7300210703185000e-016
2.5540189176516870e-016	-3.7037037037036900e-002	-1.8518518518518190e-002
-4.5767195767195750e-001	-1.2603930461073320e+000	3.1046863189720340e-001
-8.3975812547241270e-001	-9.6844293272864720e-001	-2.2204460492503130e-016

Rezultatele experimentelor sunt prezentate în Tabelul 9.7.

Comparativ cu MA din exemplul 2 obținută se poate constata că prin simetrizarea interconexiunilor, pentru a se obține o memorie global stabilă, s-a pierdut din performanța rețelei.

**Tabelul 9.7**

Zgomot / vecinătate	Vecinătate $r=1$
gaussian $\sigma=0,8$	toate cele 4 modele corect apelate din memorie
gaussian $\sigma=0,9$	3 modele au fost corect apelate, 1 cu o eroare de 1 pixel
gaussian $\sigma=1$	2 modele corect apelate din memorie și două cu 2 pixeli eroare
binar 2,8% (1 pixel eronat)	3 au fost corect apelate din memorie, unul a fost corect dependent de poziția pixelului eronat
binar 5,6% (2 pixeli eronați)	2 modele corect apelate din memorie și două cu 1 pixel eroare

## Observații

În acest capitol sunt prezentate câteva variante ale metodei pseudoinversei bazate pe arhitectura specifică RNC.

Prin simetrizarea interconexiunilor în conformitate cu §9.3 este posibilă obținerea unei rețele neuronale complet stabile. Generalizând metoda pseudoinversei se poate obține o RNC cu conexiuni de ordin superior, cu avantajul creșterii capacității de memorare.

Pentru învățarea de noi modele sau uitarea unora într-o memorie obținută prin metoda pseudoinversei, matricile de interconexiune se pot determina pornind de la cele existente fără a fi necesară reproiectarea.

MA obținută prin aceste metode are performanțe mai slabe comparativ cu cea obținută prin metoda descompunerii în valori singulare, în aceleași condiții de proiectare (aceiași precizie, în același mediu de calcul MATLAB) și de funcționare (aceleași modele memorate).

În cele ce urmează se prezintă un studiu comparativ și cu alte metode de proiectare a MA, luând în considerare câțiva parametri generali acceptați pentru caracterizarea performanțelor RN.

## 9.6. Studiul comparativ al performanțelor memoriilor asociative

Din tabel se observă că prin toate metodele implementate au putut fi memorate cele 25 de modele testate. Algoritmul rapid cu număr redus de conexiuni și descompunerea în valori singulare au determinat implicit și memorarea combinațiilor liniare ale modelelor memorate (practic au fost testate sumele modelelor).

Pentru o aceeași vecinătate de  $r=1$ , toate metodele cu excepția algoritmului rapid au demonstrat o capacitate similară de apel al modelelor ideale din memorie în condiții de zgomot gaussian. Și pentru o vecinătate  $r=3$  toate metodele testate au aceeași performanță pentru cele 25 de modele memorate.

În ceea ce privește zgomotul binar, toate memoriile asociative au o slabă capacitate de corectare cu excepția MA obținută prin descompunerea în valori singulare.

Performanța MA RNC depinde de numărul și tipul modelelor memorate (dacă prezintă regularități sau nu) de dimensiunea RNC și a vecinătății, și distribuția zgomotului. Cu cât dimensiunea RNC și a vecinătății crește, MA are o performanță mai bună.

Metodele care utilizează regula lui Hebb au o mai bună capacitate de corecție în zona centrală a imaginii spre deosebire de descompunerea în valori singulare care conduce la o mai bună corecție în zonele marginale. Aceasta se datorează modului specific de calcul al interconexiunilor prin fiecare metodă în parte.

Nu se poate face o apreciere exactă relativ la numărul de interconexiuni generate pentru memorarea aceluiași modele, deoarece de fapt multe dintre ele sunt nule.

Tabelul 9.8

Tipul rețelei și al învățării	RNCA algoritm rapid	RNCA algoritm rapid cu un număr reduc de conexiuni	RNCA descompunere în valori singulare	RNCA metoda pseudoinversei
Dimensiunea rețelei	9 x 9	9 x 9	9 x 9	9 x 9
Vecinătatea	1. r = 1 2. r = 3	1. r = 1 2. r = 3	1. r = 1 2. r = 3	r = 1
Numărul de modele memorate	25 modele	25 modele și combinațiile lor liniare	25 modele și combinațiile lor liniare	25 modele
Nr. de interconexiuni	1. 729 (unele pot fi zero) 2. 3969 (unele pot fi zero)	1. 729 (multe sunt zero) 2. 3969 (multe sunt zero)	1. 729 2. 2601	729
Nr. de modele / nr. de interconexiuni	1. mai mare decât 0.0343 2. mai mare decât 0.0063	1. mai mare decât 0.0686 2. mai mare decât 0.0126	1. mai mare decât 0.0686 2. 0.019	mai mare decât 0.0126
Capabilitatea de corectare a erorilor	mare zgomot gaussian 1. $\sigma=0,2$ 2. $\sigma=0,7$ zgomot binar: 1 pixel	mare zgomot gaussian 1. $\sigma=0,4$ 2. $\sigma=0,7$ zgomot binar: 1 pixel	mare zgomot gaussian 1. $\sigma=0,4$ 2. $\sigma=0,7$ zgomot binar: 3 pixeli	mare zgomot gaussian $\sigma=0,4$ zgomot binar: nu
Capacitatea de memorare (cu restricții)	mare (25)	mare (50)	mare (50)	mare (25)
Memorarea aditivă	da	da	nu	nu
Stări nedorite	da	da	da	da
Complexitatea	relativ simplă	simplă	complexă	complexă
Viteza de învățare	mare	foarte mare	mică	mică



Avantajul major al memoriilor ce utilizează regula lui Hebb este faptul că încorporarea unui nou mode este foarte simplă, aditivă comparativ cu metoda descompunerii în valori singulare, unde este necesară reproiectarea. Metoda pseudoinversei, deși nu necesită reproiectare, se bazează pe o procedură mai complicată.

Nici una dintre metodele expuse nu garantează evitarea modelelor nedorite (care practic sunt foarte rare), nerealizând un control al traiectoriei din starea inițială în cea finală.

Costul computațional și deci timpul de procesare implicat de metodele ce utilizează regula lui Hebb este mai redus decât cel solicitat de celelalte metode.

Simplitatea, aditivitatea, viteza de procesare a metodelor ce utilizează regula lui Hebb, similaritatea cu modul uman de procesare a informației, cu performanțe similare cu cele ale altor metode, sunt argumente ale cercetării în această direcție.

# Capitolul 10

## CONSIDERATII FINALE

De la cercetarea fundamentală teoretică, la realizarea produsului de serie care implementează rezultatele acesteia în aplicații practice, drumul este foarte lung. Prin localitatea conexiunilor, complexitatea comportamentului și viteza de procesare, RNC par să fie o soluție viabilă, implementabilă practic, aplicabilă în timp real într-o arie extrem de largă de domenii.

Este firească problema ridicată de tradiționaliști și anume dacă un sistem clasic, bazat pe un model de calcul pur matematic, nu ar putea fi o soluție la fel de bună, sau chiar perfectă pentru o anumită aplicație. Răspunsul este **DA, DAR cu condiția ca utilizatorul să găsească acel model**. Rețelele neuronale sunt necesare pentru acele fenomene complexe a căror cunoaștere este incertă, incompletă sau principial dificilă.

Rețelele neuronale celulare nu reprezintă un panaceu, ci instrumente utile pentru a pătrunde în domenii aflate încă sub dominația naturii sau imaginației. Ele oferă cadrul adecvat modelării și studiului unor fenomene complexe, mai puțin cunoscute.

### 10.1 Perspective ale evoluției RNC

Rețelele neuronale celulare și aplicațiile lor constituie unul dintre focarele de interes ale cercetării internaționale curente, iar potențialul lor extraordinar va permite noi abordări în cele mai diverse domenii ale activității umane, evident în strânsă legătură cu evoluția tehnologică a implementărilor fizice.

În cadrul general al RN este previzibilă evoluția ascendentă a interesului pentru RNC.

Gama aplicațiilor în care se utilizează rețelele neuronale celulare este extrem de vastă, depășind cu mult preocupările legate de tehnică în general și de electronică în particular. În ultimii ani au fost raportate rezultate încurajatoare privind folosirea acestora în medicină, pentru modele biologice (modele morfogenetice, modelul de

difuzie între membranele celulelor, modelul retinei), în sistemul bancar și altele. Viitorul va duce cu siguranță la diversificarea și înmulțirea aplicațiilor.

Posibilitatea comportării haotice a RNC este o premiză pentru memorarea informației într-o manieră similară creierului uman. Exploatarea acestui potențial ar putea aduce rezultate spectaculoase.

Integrarea algoritmilor genetici în RNC și elaborarea unor algoritmi genetici hibridi ce încorporează și operatori specifici aplicației, deschide noi perspective RNC. Elaborarea unor algoritmi de învățare adaptivi și utilizarea tehnologiilor optice, electromagnetice și cuantice în implementarea pe scară industrială sunt alte posibilități de dezvoltare a RNC.

Mașina universală RNC tinde să revoluționeze lumea supercomputerelor. Foarte probabil că ea va reprezenta pentru inteligența artificială ceea ce reprezintă microprocesorul pentru calculatoarele tradiționale.

## **10.2 Perspective ale evoluției RN în contextul inteligenței artificiale**

Prin aplicațiile în soluționarea unor probleme dificile ca de exemplu estimarea, identificarea, predicția, optimizarea, RN au un impact tot mai mare în domeniul vieții sociale, în luarea deciziilor economice și chiar politice.

Evoluția RN a fost legată încă de la apariție de dezvoltarea așa numitelor “programe de inteligență artificială”. În sprijinul integrării celor două metode de abordare a cunoașterii vin și ultimele descoperiri din neurologie care confirmă existența a două mecanisme de procesare a informației în creierul uman:

- unul rapid, eficient, subconștient;

Rețelele neuronale par a fi cadrul adecvat acestui nivel de prelucrare a informațiilor, de abordare directă a lumii înconjurătoare.

- celălalt lent, serial, conștient;

Prin intermediul acestuia se evaluează gândurile curente, se abordează conștient probleme, se dezvoltă idei noi. Oamenii conștientizează acest proces, putând da explicații asupra raționamentelor.

“Programele de inteligență artificială” au o manieră similară de procesare .

Sisteme hibride care încorporează atât o RN cât și un sistem expert funcționează în prezent cu succes, ca de exemplu în controlul unor procese industriale. Este previzibilă continuarea cercetării în această direcție în vederea îmbunătățirii performanțelor.

În ultimii ani odată cu revigorarea cercetării în domeniul teoriei mulțimilor fuzzy s-a afirmat tot mai mult tendința integrării celor două direcții de cercetare în inteligența

artificială. Teoria mulțimilor fuzzy reprezintă un instrument foarte puternic pentru tratarea ambiguităților de natură nestatistică și pentru modelarea raționamentului nuanțat și aproximativ. Această teorie a furnizat rețelelor neuronale modele care le permit o mai bună abordare a datelor incomplete, ambigue, sau chiar contradictorii. Și invers, calculul neuronal este folosit în proiectarea și optimizarea sistemelor fuzzy. Există aplicații în care se folosesc ambele modele, inițial cu o funcționare independentă, ulterior într-o arhitectură omogenă.

Este previzibilă o evoluție rapidă a cercetării în domeniul de fuziune dintre rețele neuronale și teoria fuzzy, calculul neuro-fuzzy, care oferă un grad mai sporit de robustețe și un coeficient de inteligență mai ridicat.

Integrarea tehnicilor inteligente poate avea consecințe spectaculoase, sinergia lor evoluând spre o formă de inteligență artificială netradițională.

### 10.3 Contribuții personale

În cercetarea efectuată asupra aspectelor teoretice și aplicative ale RNC autorul și-a adus următoarele contribuții:

- Caracterizarea într-o manieră unitară și sistematică a tipurilor de RNC dezvoltate până în prezent, a aspectelor importante în proiectarea lor și a modului de abordare într-o aplicație curentă (Cap.2, Cap.3), a câtorva aplicații posibile ca operatori în probleme complexe de prelucrări de imagini (Cap.4), a caracteristicilor și modalităților de sinteză a memoriilor asociative (Cap.5);
- Determinarea matricii de interconexiune (în MATHCAD) pentru detecția părții statice a imaginilor în mișcare (§ 4.3);
- Implementarea unui program (în MATHCAD) de determinare a matricilor de interconexiune a RNC pentru reconstrucția obiectelor tridimensionale prin interpolare și aproximare, pentru rotirea lor în spațiu (§4.3)
- Introducerea unei noi structuri de RNC funcționând ca memorie asociativă bidirecțională, ce utilizează regula lui Hebb, pentru determinarea interconexiunilor de reacție (§ 6.2), fundamentarea teoretică, determinarea condițiilor impuse asupra modelelor de intrare, a capacității de memorare și experimentarea MA obținute;
- Implementarea în C++ a algoritmului rapid de determinare a matricilor de interconexiune a unei memorii asociative [95], experimentarea memoriei obținute (§ 6.3);
- Dezvoltarea unui nou algoritm ce utilizează o regulă a lui Hebb modificată, bazată pe mediere, pentru determinarea matricilor de interconexiune de reacție și de curent, implementarea în Borland C++ și experimentarea practică a memoriei proiectate (§ 6.4.);

- Aplicarea metodei regulii lui Hebb modificate , din § 6.4 în conducerea automată a unui vehicul (§ 6.4.3);
- Dezvoltarea modelului memoriei asociative ce utilizează descompunerea în valori singulare în MATLAB și C, testarea memoriei obținute în condiții de perturbații ale modelelor de intrare (Cap.8);
- Determinarea interconexiunilor unei memorii asociative prin metoda pseudoinversei în MATLAB și studiul MA obținute (Cap.9);
- Studiul comparativ din punctul de vedere al performanțelor tuturor modelelor de memorii asociative cunoscute până în prezent, cu evidențierea avantajelor și dezavantajelor fiecăruia (§ 6.5), (§ 9.5);

# Anexa 1

## Distanța metrică

### Definiția 1

Distanța metrică  $d_{k,l}(i,v)$  este o funcție cu valori reale care satisface următoarele axiome:

1.  $d_{k,l}(i,v) = d_{k,l}(v,i)$  pentru  $\forall i,v$
2.  $d_{k,l}(i,v) \leq d_{k,l}(i,q) + d_{k,l}(q,v)$  pentru toți  $i,v,q = \{i_1, i_2, \dots, i_n, l\}$ ,  $i_j \in \{1, 2, \dots, N_j\}$  pentru toți  $j \in \{1, 2, \dots, n\}$
3.  $d_{k,l}(i,v) \geq 0$  pentru  $\forall i \neq v$
4.  $d_{k,l}(i,i) = 0$

## Distanța Hamming dintre doi vectori $v_1$ și $v_2$ n dimensionali în funcție de produsul lor scalar

### Definiția 2

Distanța Hamming dintre doi vectori  $v_1$  și  $v_2$  ( $HD(v_1, v_2)$ ) n dimensionali, este dată de numărul biților prin care diferă cei doi vectori.

Produsul scalar al celor doi vectori n dimensionali se poate scrie ca numărul biților (din aceeași poziție) identici în cei doi vectori minus numărul pozițiilor prin care diferă:

$$v_1 \cdot v_2 = (n - HD(v_1, v_2)) - HD(v_1, v_2) \quad (A1.1)$$

## Anexa 2

Fie următoarele matrici de interconexiune:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathbf{A}_0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{a} & \mathbf{a} & \mathbf{a} \\ \mathbf{a} & \mathbf{b} & \mathbf{a} \\ \mathbf{a} & \mathbf{a} & \mathbf{a} \end{bmatrix}, \quad \mathbf{I}, \quad \mathbf{A}^\tau = [\mathbf{0}]$$

$$\mathbf{B}^\tau = \begin{bmatrix} \mathbf{c} & \mathbf{c} & \mathbf{c} \\ \mathbf{c} & \mathbf{d} & \mathbf{c} \\ \mathbf{c} & \mathbf{c} & \mathbf{c} \end{bmatrix} \quad (\text{A2.1})$$

unde:  $\mathbf{A}_0, \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{I}$  sunt constante și  $\mathbf{A}_0 > 1$ .

Ecuția de stare, respectiv cea de ieșire a circuitului, pentru  $C=1\text{m F}$  și  $R=1\text{ k}\Omega$  sunt:

$$\dot{x}_{ij}(t) = -x_{i,j}(t) + \mathbf{A}_0 \cdot y_{i,j}(t) \cdot \mathbf{b} u_{i,j}(t) + \mathbf{a} \cdot \sum_{C_{k,l} \in N_r(i,j)} u_{k,l}(t) + \mathbf{d} \cdot u_{i,j}(t-\tau) +$$

$$+ \mathbf{b} \cdot \sum_{C_{k,l} \in N_r(i,j)} u_{k,l}(t-\tau) + \mathbf{I} \quad \text{ptr. } 1 \leq i \leq M; 1 \leq j \leq N \quad (\text{A2.2})$$

$$y_{i,j}(t) = f(x_{i,j}(t)) = 0.5 (|x_{i,j}(t) + 1| - |x_{i,j}(t) - 1|) \quad (\text{A2.3})$$

Dacă intervalul de timp în care variază intrarea este mult mai mic decât intervalul de timp al tranziției rețelei atunci template-ul dat de relația (A2.1) nu este doar complet stabil (având conexiuni pozitive), ci și variabilele de stare sunt monotone în timp:

$$\text{dacă } \dot{x}(t_0) \leq 0 \text{ atunci } \dot{x}(t) \leq 0 (t > t_0) \quad (\text{A2.4})$$

$$\text{dacă } \dot{x}(t_0) \geq 0 \text{ atunci } \dot{x}(t) \geq 0 (t > t_0)$$

Se scrie ecuația de stare sub forma:

$$\dot{x}_{i,j}(t) = -x_{i,j}(t) + \mathbf{A}_0 y_{i,j}(t) + \mathbf{g}_{i,j} \quad (\text{A2.5})$$

unde:

$$\mathbf{g}_{i,j} = \mathbf{b} u_{i,j}(t) + \mathbf{a} \sum_{C_{k,l} \in N_r(i,j)} u_{k,l}(t) + \mathbf{d} u_{i,j}(t-\tau) + \mathbf{b} \sum_{C_{k,l} \in N_r(i,j)} u_{k,l}(t-\tau) + \mathbf{I} \quad (\text{A2.6})$$

este de fapt independentă de timp pentru că imaginea de intrare nu se modifică semnificativ în timpul tranziției. Starea staționară a celulei  $C_{i,j}$  este determinată de:

$$-x_{i,j} + \mathbf{A}_0 f(x_{i,j}) + \mathbf{g}_{i,j} = 0 \quad \text{unde } x_{i,j} = \lim_{t \rightarrow \infty} x_{i,j}(t) \quad (\text{A2.7})$$

Reprezentând grafic ecuația  $A_0 f(x_{i,j}) = x_{i,j} - g_{i,j}$  constatăm că avem trei cazuri:

a) Dacă  $A_0 - 1 < g_{i,j}$  atunci avem un singur punct de echilibru la  $O_1$  (Fig.A2.1a). Valoarea proprie a circuitului echivalent liniarizat în punctul  $O_1$  este  $\lambda = -1$ , din moment ce punctul de echilibru este stabil. Este evident că în timpul tranziției  $x_{i,j}(t) \leq A_0 + g_{i,j}$ . De aceea din (A2.7) rezultă că:

$$\dot{x}_{i,j}(t) \geq 0, \quad t \geq t_0 \quad (\text{A2.8})$$

Se remarcă că starea celulei va crește ca valoare monoton de la  $x_{i,j}(0)$  până la  $x_{i,j}$ , unde atinge asimptotic punctul de echilibru.

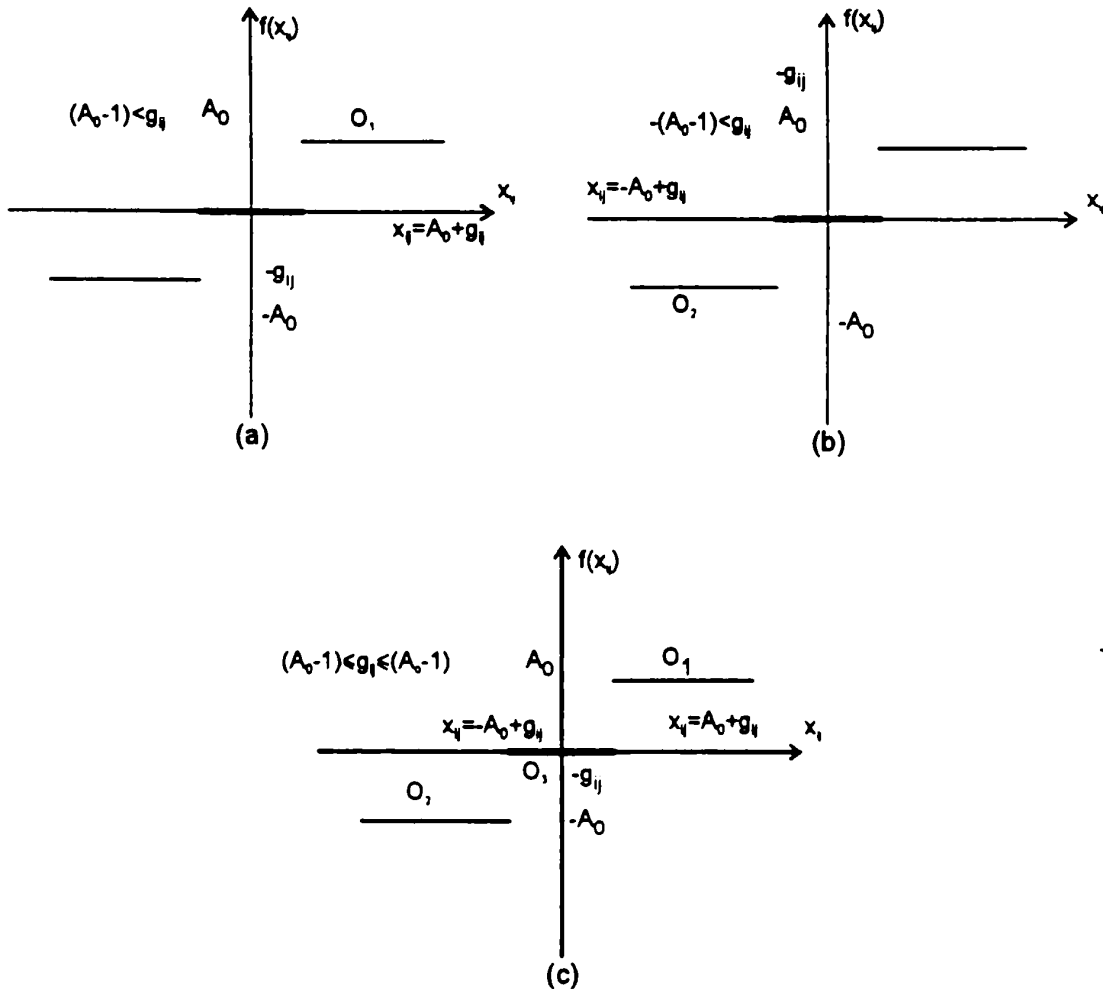


Fig. A2.1 Punctele de funcționare a celulei  $C_{i,j}$

b) Similar, dacă  $-(A_0 - 1) > g_{i,j}$  atunci avem un singur punct de echilibru la  $O_2$  (Fig.A.2.1b). În timpul tranziției avem  $x_{i,j}(t) \geq A_0 + g_{i,j}$ . De aceea din (A2.7) rezultă că:

$$\dot{x}_{i,j}(t) \leq 0, \quad t \geq t_0 \quad (\text{A2.9})$$

Se observă că variabila de stare a celulei  $C_{i,j}$  va descrește monoton de la  $x_{i,j}(0)$  până la  $x_{i,j}$ , unde atinge asimptotic punctul de echilibru  $O_2$ .



c) Dacă  $-(A_0 - 1) \leq g_{i,j} \leq A_0 - 1$ , atunci există trei puncte de echilibru (Fig.a2.1c). Două dintre ele  $O_1$  și  $O_2$  sunt localizate în regiunea orizontală a funcției  $f(x_{i,j})$ . Punctele  $O_1$  și  $O_2$  sunt local stabile, iar  $O_3$  nu este stabil pentru că valoarea proprie a circuitului liniarizat echivalent în acest punct este  $\lambda = 1/(A_0 - 1) > 0$ .

Se poate demonstra că, dacă:

- $\dot{x}_{i,j}(t_0) = (A_0 - 1)x_{i,j}(t_0) + g_{i,j} > 0$ ,  $t \geq t_0$  atunci  $x_{i,j}$  va crește până va atinge punctul  $O_1$ . Asta înseamnă că starea celulei va crește ca valoare monoton de la  $x_{i,j}(0)$  până la  $x_{i,j}$ , unde atinge asimptotic punctul de echilibru  $O_1$ .
- $\dot{x}_{i,j}(t_0) = (A_0 - 1)x_{i,j}(t_0) + g_{i,j} < 0$  atunci  $x_{i,j}$  va descreește până atinge  $O_2$ .

Din demonstrația prezentată rezultă că variabila de stare este strict crescătoare sau strict descrescătoare pentru  $t_0 < t < t_s$ . Circuitul va găsi întotdeauna un punct de echilibru stabil în regiunea orizontală a caracteristicii liniare pe porțiuni  $f(x_{i,j})$ .

1

# Anexa 3

## Algoritm pentru descompunerea în valori singulare

Se prezintă un algoritm de descompunere în valori singulare pentru cazul general al unei matrici cu  $m$ -linii și  $n$ -coloane.

Dacă  $A$  este o matrice reală de tip  $m \times n$ , atunci există matricile ortogonale  $U$  și  $V$  de ordin  $m$ , respectiv  $n$ , astfel încât:

$$\begin{aligned} U^T \cdot A \cdot V &= \begin{bmatrix} \mathbf{D} \\ \mathbf{0} \end{bmatrix} \quad \text{daca } m \geq n, \text{ sau} \\ U^T \cdot A \cdot V &= \begin{bmatrix} \mathbf{D} \\ \mathbf{0} \end{bmatrix}^T \quad \text{daca } m < n. \end{aligned} \tag{A3.1}$$

unde:

- $D = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$  este o matrice diagonală de dimensiune  $r \times r$ , cu  $r = \min\{m, n\}$  și scalarii  $\sigma_1, \sigma_2, \dots, \sigma_r$  sunt valorile singulare ale matricii  $A$ . Descompunerea dată de relația (A3.1) este descompunerea după valorile singulare (DSV) a matricii  $A$
- coloanele lui  $U$  se numesc vectorii singolari la stânga lui  $A$ , iar coloanele lui  $V$  vectorii singolari la dreapta.

Ortogonalitatea se referă la coloanele matricii  $U$  și  $V$ :

$$\sum U_{ik} \cdot U_{ii} = 1 \tag{A3.2}$$

$$\sum U_{ik} \cdot U_{ii} = 1 \tag{A3.3}$$

Determinarea descompunerii în valori singulare se face eficient în două etape:

- reducerea lui  $A$  la forma bidiagonală;
- diagonalizarea formei bidiagonale;

În procesul de calcul se utilizează exclusiv transformări ortogonale [89].

## Reducerea la forma bidiagonală

Pentru simplitate, în continuare, se consideră  $m \leq n$ , iar pentru cazul  $m < n$  se vor menționa doar deosebirile.

Se utilizează transformările ortogonale de tip Householder.

### Definiția 1

Se numește transformare Householder o matrice de forma :

$$U = I_n - 2 \cdot u \cdot u^t, \quad \text{cu } u \in \mathbb{R}^n \text{ și } u^t \cdot u = 1$$

Se poate verifica ușor că  $U$  este simetrică și ortogonală. Matricea  $U$  se numește reflector elementar.

Dacă  $u^t \cdot u = 1$  și  $u \neq 0$ , se poate evita normalizarea și se poate scrie:

$$U = I_n - 2 \cdot \frac{u \cdot u^t}{u^t \cdot u} = I_n - \beta \cdot u \cdot u^t, \quad \text{cu } \beta = \frac{2}{u^t \cdot u} \quad (\text{A3.5})$$

Înmulțind la stânga sau la dreapta o matrice  $A \in \mathbb{R}^{n \times n}$  cu o transformare Householder construită adecvat, se pot anula simultan mai multe elemente dintr-o coloană, respectiv linie a lui  $A$ . Această proprietate, împreună cu ortogonalitatea, justifică largă aplicare a transformărilor Householder.

Dacă se aplică alternativ transformări la stânga și la dreapta lui  $A$  corespunzător alese, matricea  $A$  poate fi adusă la forma bidiagonală. Transformările aplicate la stânga lui  $A$  anulează mai multe elemente de pe o anumită coloană, iar transformările la dreapta lui  $A$  anulează simultan mai multe elemente de pe o linie a lui  $A$ .

Pentru  $m = 5$  și  $n = 4$  transformările se efectuează conform următoarei diagrame:

$$A_1 = A = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix}; \quad A_2 = U_1 \cdot A_1 = \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{bmatrix}; \quad A_3 = A_2 \cdot V_1 = \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{bmatrix}$$

$$A_4 = U_2 \cdot A_3 = \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{bmatrix}; \quad A_5 = A_4 \cdot V_2 = \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{bmatrix}; \quad A_6 = U_3 \cdot A_5 = \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \\ 0 & 0 & 0 & x \end{bmatrix};$$

$$A_7 = U_4 \cdot A_6 = \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Dacă  $m \neq n$ , matricea finală are forma de mai jos (pentru  $m = 4, n = 6$ ):

$$B = \begin{bmatrix} x & x & 0 & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & x & x & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 \end{bmatrix}$$

### Observație:

Prin convenție, în diagramele de mai sus  $x$  indică un element nenul al matricii.

Construcția transformării Householder, pentru un vector  $x \in \mathbb{R}^n$  cu  $x \neq 0$  se face cu următorul algoritm:

1. Se calculează  $s = \max\{|x_i|\}, i=1, \dots, n$ . Dacă  $s = 0$ , atunci STOP (în acest caz toate elementele lui  $x$  sunt nule, deci nu se mai aplică transformarea);

2. a) Se calculează :

$$r = \text{sign}(x_i) \cdot s \cdot \sqrt{\frac{x_1^2}{s^2} + \dots + \frac{x_n^2}{s^2}} \quad (\text{A.3.6})$$

b)  $x_i$  se înlocuiește cu  $x_{i+r}$ ;

c) Se calculează:

$$\beta = \frac{1}{r \cdot x_i} \quad (\text{A.3.7})$$

3) Vectorul  $u$  al transformării este:  $u = [0, 0, \dots, x_{i+r}, x_{i+r+2}, \dots, x_{i+r+n}]$ ;

### Observații

1) Elementele lui  $x$  au fost scalate, pentru a evita apariția depășirilor aritmetice la evaluarea lui  $r$ .

2) În algoritmul de mai sus transformarea  $u$  a fost calculată pentru coloana  $i$ . Componentele lui  $x = (x_1, x_2, \dots, x_i, \dots, x_n)$  vor rămâne neschimbate până la indicele  $i$ ,  $x_i$  se modifică, iar componentele de la  $i+1$  la  $n$  vor deveni nule.

3) Analog se calculează transformarea  $v$  cu deosebirea că  $x = (x_1, \dots, x_m)$ .



Algoritmul de calcul nu formează efectiv matricea  $U$ , problema se reduce la operarea asupra coloanelor lui  $A$ . Astfel, notând  $A=(a_1, \dots, a_m)$ , unde  $a_i$  este coloana  $i$  a lui  $A$ ,  $i=1, \dots, m$ , rezultă că  $U \times A=(U \times a_1, \dots, U \times a_m)$ , deci problema se reduce la calculul produsului  $U \times a$ , cu  $a \in \mathbb{R}^n$ . Putem scrie  $U \times a=(I_n - \beta \times u \times u^t) \times a = a - \beta \times (u^t \times a) \times u$ . Practic se aplică transformarea  $U$ , pe rând, fiecărei coloane în parte. În acest mod se reduce mult durata de calcul la realizarea bidiagonalizării.

Programul realizat în această lucrare implementează următorul algoritm, ce determină rezultatul aplicării lui  $U$  și  $V$  asupra unui vector  $a$ , cu vectorii  $u$ ,  $a \in \mathbb{R}^n$  și  $\beta \in \mathbb{R}$ .

- 1) Se calculează  $t = \beta \times (u^t \times a)$ ; Dacă  $t=0$ , atunci STOP;
- 2) Se înlocuiește vectorul  $a$  cu  $(a - t \times u)$ . Dacă  $a=(a_1, \dots, a_n)$  și  $u=(u_1, \dots, u_n)$  se poate înlocui  $a_i$  cu  $(a_i - t \times u_i)$ ,  $i=1, \dots, n$ .

Matricea bidiagonală finală  $B$  este dată de

$$B=U_k \times \dots \times U_2 \times U_1 \times A \times V_1 \times \dots \times V_l,$$

unde:  $k = \min \{n, m-1\}$  și  $l = \min \{m, n-2\}$ .

Este ușor de văzut că, în general, nu se poate realiza diagonalizarea lui  $A$  utilizând astfel de transformări. (Transformarea  $V_i$  trebuie să nu perturbe zerourile introduse de  $U_i$ ).

### Diagonalizarea formei bidiagonale

Evident în urma transformării ortogonale aplicate, valorile singulare ale lui  $B$  coincid cu valorile singulare ale lui  $A$ .

În cazul în care liniile nule finale ale lui  $B$  pot fi eliminate, deci, dacă  $m(n, \text{reducerea la forma bidiagonală a lui } A \text{ conduce la o matrice bidiagonală de ordin } n$ .

În cazul  $m \geq n$ , matricea bidiagonală rezultată este de tip  $m \times (m+1)$ . Totuși, adăugând o linie nulă se obține o matrice bidiagonală de ordin  $(m+1)$  ale cărei valori singulare sînt cele ale lui  $A$ , plus o valoare singulară zero. Astfel, problema se reduce la determinarea valorilor singulare ale unei matrici bidiagonale pătrate, de ordin  $p$  ( $p = n$  sau  $m+1$ , după cum  $m \geq n$ , respectiv  $m < n$ ), de forma:

$$B = \begin{bmatrix} s_1 & e_1 & & \dots & 0 \\ 0 & s_2 & e_2 & & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & s_{p-1} & e_{p-1} & \\ 0 & \dots & \dots & \dots & s_p & \end{bmatrix} \quad (\text{A.3.8})$$

Diagonalizarea lui  $B$  se poate face prin aplicarea repetată a unor transformări ortogonale (rotații plane), care să anuleze elementele de deasupra diagonalei matricii  $B$  și în același timp să păstreze zerourile de sub diagonală.

## Definiția 2

O rotație plană în planul (i,j) este dată de matricea:

$$G(i,j) = \begin{bmatrix} 1 & & & & & \\ & \dots & & & & \\ & & 1 & & & \\ & & & c & s & \dots \\ & & & -s & c & \dots \\ & & & & & 1 \\ & & & & & \dots \\ & & & & & & 1 \end{bmatrix} \quad (A3.9)$$

Aceasta coincide cu  $I_n$  cu excepția elementelor (i,j),(i,i),(j,i) și (j,j). Dacă se înmulțește la stânga matricea B cu transformarea G(i,j), se obține o matrice care diferă de B doar în liniile i și j. Similar, matricea  $B \times G(i,j)$  diferă de B doar în coloanele i și j.

Prin determinarea unor rotații plane  $V_{23}, \dots, V_{p-1,p}$  și  $U_{12}, \dots, U_{p-1,p}$  astfel încât:

$B' = U_{p-1,p} \dots U_{23} \times U_{12} \times B \times V_{12} \times V_{23} \dots V_{p-1,p}$  este o matrice bidiagonală, se poate efectua o iterație a algoritmului.

$$B \cdot V_{12} = \begin{bmatrix} x & x & 0 & 0 & 0 \\ f & x & x & 0 & 0 \\ 0 & 0 & x & x & 0 \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \Rightarrow B_1 = U_{12}^T \cdot B \cdot V_{12} = \begin{bmatrix} x & x & f & 0 & 0 \\ 0 & x & x & 0 & 0 \\ 0 & 0 & x & x & 0 \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \Rightarrow$$

Operațiile sunt ilustrate în diagramele de mai jos, pentru  $p = 5$ .

$$\Rightarrow B_1 \cdot V_{23} = \begin{bmatrix} x & x & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 \\ 0 & f & x & x & 0 \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \Rightarrow B_2 = U_{23}^T \cdot B_1 \cdot V_{23} = \begin{bmatrix} x & x & 0 & 0 & 0 \\ 0 & x & x & f & 0 \\ 0 & 0 & x & x & 0 \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \dots$$

Astfel sunt anulate alternativ elementele nenule de sub diagonală și de deasupra diagonalei. Fiecare astfel de operație, exceptând-o pe ultima, generează un nou element nenul (f în diagrame), care este apoi la rândul său anulat. Se poate demonstra că elementul  $e_{p,1}$  va tinde rapid spre zero. Alte elemente tind mai lent la zero. Studiile teoretice și practice au arătat că prin scăderea din elementele diagonale ale matricii B, a valorilor proprii a submatricii principale finale de dimensiune  $2 \times 2$  din  $B \times B^T$ , cea mai apropiată de

$s_p^2$ , se mărește viteza de convergență a algoritmului. Pentru a decide dacă un element de deasupra diagonalei  $e_i$  este neglijabil, se poate utiliza relația:

$$|e_i| \leq \text{err} \cdot (|s_{i+1}| + |s_i|) \quad (\text{A3.10})$$

Observație:  $e_i$  este cel mai mic număr pozitiv, reprezentabil pe calculator, astfel încât  $\text{float}(1+\text{err}) - 1$ , unde  $\text{float}(a)$  este valoarea reprezentării în virgulă mobilă a numărului  $a$ . Valoarea se calculează în program. Această eroare  $\text{err}$  este principalul factor ce limitează precizia de descompunere deoarece ea decide dacă o valoare este singulară sau nu.

Când  $e_{p-1}$  este neglijabil,  $s_p$  poate fi acceptat ca valoare singulară aproximativă, iar iterația continuă cu submatricea de ordin  $p - 1$  a lui  $B$ , care conține celelalte valori singulare.

Pentru matrici de dimensiuni mari, este posibil ca în cursul procesului un element  $e_i$ ,  $i < p-1$ , să devină neglijabil. Acest caz nu reprezintă convergența; totuși, valorile singulare ale lui  $B$  sunt chiar cele ale submatricilor de ordin  $l$ , respectiv  $p-l$ , deci problema se poate descompune. Aplicând algoritmul separat pentru fiecare din aceste submatrici, se poate reduce considerabil efortul de calcul pentru obținerea vectorilor singulari.

O tratare specială o necesită valorile singulare nule ale lui  $B$ . O valoare  $s_l$  este neglijabilă dacă:

$$|s_l| \leq \text{err} \cdot (|e_{l-1}| + |e_l|) \quad (\text{A3.11})$$

Când  $s_l$  este neglijabil, problema poate fi redusă prin una din următoarele două căi:

Cazul în care  $l < p$ . Punând  $s_l=0$  și înmulțind la stânga matricea  $B$  curentă cu secvența de rotații  $U_{l,l+1}^T, U_{l,l+2}^T, \dots, U_{l,p}^T$ , elementul nenul din linia  $l$  este deplasat succesiv spre dreapta, rezultând în final o matrice bidiagonală având linia  $l$  nulă. Procesul este ilustrat de diagramele de mai jos ( $p=5, l=2$ ):

$$\begin{aligned} \mathbf{B}_1 &= \begin{bmatrix} x & x & 0 & 0 & 0 \\ 0 & 0 & x & 0 & 0 \\ 0 & 0 & x & x & 0 \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \Rightarrow \mathbf{B}_2 = \begin{bmatrix} x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & f & 0 \\ 0 & 0 & x & x & 0 \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \Rightarrow \\ \Rightarrow \mathbf{B}_3 &= \begin{bmatrix} x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & f \\ 0 & 0 & x & x & 0 \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \Rightarrow \mathbf{B}_4 = \begin{bmatrix} x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x & x & 0 \\ 0 & 0 & 0 & 0 & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \end{aligned}$$

Se observă din diagrame că elementul nenul de pe linia  $l=2$  este deplasat spre dreapta până când se anulează.

Cazul în care  $l = p$ , adică  $s_p=0$  ( $s_p$  este neglijabilă). În acest caz, înmulțind matricea  $B$  curentă cu secvența de rotații  $V_{p-1,p}, V_{p-2,p}, \dots, V_{1,p}$ , elementul nenul din coloana  $p$  este deplasat succesiv în sus, rezultând în final o matrice bidiagonală având coloana  $p$  nulă, deci  $s_p=0$  este acceptată ca valoare singulară nulă, deoarece și  $c_{p-1}=0$ . Procesul este ilustrat de diagramele de mai jos ( $p = 4$ ):

$$\begin{aligned}
 \mathbf{B}_1 &= \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow \mathbf{B}_2 = \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & f \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow \\
 \mathbf{B}_3 &= \begin{bmatrix} x & x & 0 & f \\ 0 & x & x & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow \mathbf{B}_4 = \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

### Calculul vectorilor singurari

Matricile transformărilor utilizate pentru a reduce pe  $A$  la forma bidiagonală pot fi înmulțite între ele pentru a obține matricile ortogonale  $U$  și  $V$  ( $B=U^T \times A \times V$ ). Pentru a calcula vectorii singurari este necesar ca toate rotațiile etichetate cu  $U$  și  $V$  să fie înmulțite la dreapta cu  $U$ , respectiv  $V$ , de îndată ce sunt generate.

Aplicând în mod repetat algoritmul descris mai sus se obține în final descompunerea în valori singulare a matricii  $A$ , având forma

$$\mathbf{A} = \mathbf{U} \cdot \begin{pmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \cdot \mathbf{V}^T \quad (\text{A3.12})$$

unde  $D_1$  este matricea ce conține pe diagonală valorile singulare nenule ale lui  $A$ .



# Anexa 4

## Metoda pseudoinversei

Fie setul de  $m$  vectori de intrare  $x^1, x^2, \dots, x^m$  și setul de  $m$  vectori doriți de ieșire  $y^1, y^2, \dots, y^m$ , de o aceeași dimensiune  $1 \times n$ .

Se pune problema determinării unei transformări liniare sau a unui operator matricial  $W$  care minimizează norma euclidiană a diferenței dintre răspunsul dorit  $y^i$  și ieșirea curentă  $W \cdot x^i$  pentru  $i=1,2,\dots,m$ .

Dacă se definește matricea modelelor de intrare  $X = [x^1, x^2, \dots, x^m] \in \mathbb{R}^{n \times m}$  respectiv a celor de ieșire  $Y = [y^1, y^2, \dots, y^m] \in \mathbb{R}^{n \times m}$ , problema se poate reformula astfel:

Fiind date matricea  $X$ , a modelelor de intrare și matricea  $Y$ , a modelelor de ieșire dorite, să se determine transformarea liniară sau operatorul matricial  $W$ , care minimizează norma euclidiană a unei matrici, definite ca diferența dintre răspunsul dorit  $Y$  și ieșirea curentă  $W \cdot X$ .

Norma matricii erorii de asociere este  $\|Y - W \cdot X\|$ . Conform metodei pseudoinversei soluția  $W$  care minimizează norma matricii de eroare, în sensul celor mai mici pătrate este dată de [50] :

$$W = Y \cdot X^* \tag{A4.1}$$

unde  $X^*$  este pseudoinversa matricii  $X$ .

Ecuția (A4.1) reprezintă **regula de învățare a pseudoinversei** și o memorie astfel proiectată se numește **memorie pseudoinversă (pseudoinverse memory)**.

O condiție suficientă pentru o asociere perfectă este să avem:

$$X^* \cdot X = I \tag{A4.2}$$

unde  $I$  este matricea identitate

Când este satisfăcută condiția (A4.2), răspunsul curent generat la aplicarea matricii  $X$  prin intermediul matricii de interconexiune  $W$ , va fi chiar  $Y$ :

$$W \cdot X = Y \cdot X^* \cdot X = Y \tag{A4.3}$$

Condiția (A4.2) poate fi satisfăcută dacă se utilizează un set de vectori de intrare (coloanele matricii  $X$ ) care sunt liniar independenți. Această condiție, la rândul ei implică inegalitatea  $n \geq m$ , adică dimensiunea spațiului modelelor de intrare să fie mai mare decât numărul asocierilor.

În aceste condiții matricea pseudoinversă se definește prin:

$$X^+ = (X^T \cdot X)^{-1} \cdot X^T \quad (\text{A4.4})$$

unde  $X^T$  este transpusa matricii  $X$ .

Se poate observa că, (A4.2), condiția pentru asocierea perfectă este îndeplinită imediat.

Proprietățile matricii pseudoinverse sunt:

$$\begin{aligned} X \cdot X^+ \cdot X &= X \\ X^+ \cdot X \cdot X^+ &= X^+ \\ X \cdot X^+ &= (X \cdot X^+)^T \\ X^+ \cdot X &= (X^+ \cdot X)^T \end{aligned} \quad (\text{A 4.5})$$

# Bibliografie

- [1] M. Adachi, K. Aihara, M. Kotani "An analysis of Associative Memory Dynamics with Chaotic Neural Networks" International Symposium on Nonlinear Theory and its Applications, NOLTA '93, Proceedings, pag.1169-1172, 1993
- [2] M. Balsi "Recurrent Back-Propagation for Cellular Neural Networks" ECCTD'93- Circuit Theory and Design, Proceedings, pag.677-682, 1993
- [3] M. Biehl, J. K. Anlauf, W. Kinzel "Perceptron Learning by Constrained Optimization", Neurodynamics,1990
- [4] Corina Botoca, "Rețele neuronale", Sesiunea de comunicări tehnico-științifice ale tinerilor absolvenți, Timișoara, nov., pag.266-273 ,1989
- [5] Corina Botoca, "O prezentare generală a rețelelor neuronale", referat de doctorat, Timișoara, 1992
- [6] Corina Botoca About Some New Aspects of Neural Networks. Part I", Sesiunea de comunicări internaționale SCS '93, Iași, pag.324-328, 1993
- [7] Corina Botoca, "About Some New Aspects of Neural Networks.Part II", Sesiunea de comunicări internaționale SCS '93, Iași, pag.329-332, 1993
- [8] Corina Botoca, " Rețele neuronale celulare. Memorii asociative", referat de doctorat, Timișoara, 1994
- [9] Corina Botoca, O.Sas "Nonlinear Equalizer for the Reconstruction of Binary Signals", Proceedings of the Symposium on Electronics and Telecommunications, Timișoara, pag.227-232, 1994
- [10] Corina Botoca, P. Szasz, "A Study of Modified Hebb Rule Neural Networks for Data Compression, Cellular Neural Networks", Buletinul Științific și, Tehnic, UTT, Electrotehnică Electronică șiTelecomunicații, Tom 40 (54), pag.143-152, 1995
- [11] Corina Botoca, "An Associative Memory Using a Two-Layer Cellular Neural Network, Part I", Symposium on Signal, Circuits & Systems, oct.1995, Proceedings , pag.121-124
- [12] Corina Botoca, "An Associative Memory Using a Two-Layer Cellular Neural Network Part II", C.Botoca, Symposium on Signal, Circuits & Systems, oct., Proceedings, pag.125-128, 1995

- [13] Corina Botoca, "Utilizarea rețelelor neuronale celulare ptr. detecția mișcării, A XXVI -a Sesiune de Comunicări Științifice cu Participare Internațională, Academia Tehnică Militară București, pag.100-105, 1995
- [14] Corina Botoca "An Improved Associative Memory using a Fast Algorithm on Cellular Neural Network", NEUREL '97, Belgrad, Proceedings , pag 90-93, 1997
- [15] Corina Botoca, "Using Associative Memories on Cellular Neural Networks Control an Automated Guided Vehicle", acceptată spre publicare, Oradea, 1998
- [16] T. Boros, K. Lotz, A. Radványi, T. Roska, "Some useful, new, nonlinear and delay-type templates", DNS-1-1991
- [17] V. Căprariu, "Ghid de utilizare. Turbo C 2.0", microInformatica SRL, Cluj Napoca, 1991
- [18] M. Brucoli, L. Carnimeo, G. Grassi, "A New Synthesis Procedure For Associative Memories Using Discrete Time CNN ", Symposium On Nonlinear Theory And Its Applications, NOLTA '94, Kagoshima, 1994
- [19] M. Brucoli, L. Carnimeo, G. Grassi "Discrete Time Cellular Neural Networks with Learning and Forgetting Capabilities", IEEE Transactions on Circuits and Systems I, vol 42, iulie, pag.396-398, 1995
- [20] M. Brucoli, L. Carnimeo, G. Grassi "A Design Method for Associative Memory Using DTCNN with a Reduced Number of Cell Interconnections", MELECON '96, Bari, Italia Proceedings, vol II, pag.623-626, 1996
- [21] M. Brucoli, L. Carnimeo, G. Grassi, "Completely Stable Cellular Neural Networks For Associative Memories", ECCTD'97, Budapesta, Proceedings, pag.678-683, 1997
- [22] M. Brucoli, L. Carnimeo, G. Grassi, "Associative Memories Using Discrete Time Second Order Neural Networks With Local Interconnections ", IEEE Transactions on Circuits and Systems I, vol 44, nr.2, Febr, pag.153-158, 1997
- [23] C. M. Bucur, C. A. Popeea, Gh. Simion, *Matematici speciale. Calcul numeric*, Editura didactică și pedagogică, București, 1983
- [24] Ciocoiu I., "Contribuții la studiul aplicării rețelelor neuronale în studiul prelucrării semnalelor", teză de doctorat, Universitatea Tehnică "Gh.Asachi", 1996
- [25] P. P. Civalleri, M. Gilli "Some Stability Properties of CNN's with Delay"IEEE International Workshop on Cellular Neural Networks and Their Applications, pag.94-100, 1992
- [26] P. P. Civalleri, M. Gilli "On the Relation Between Equilibrium Points and Lymit Cycles in one Dimension Cellular Neural Networks" ECCTD'97, Budapesta, Proceedings. pag.157-161, 1997
- [27] L. O. Chua , L. Yang, "Cellular Neural Networks. Theory" IEEE Transactions on Circuits and Systems, vol 36, nr.10. octombrie, pag.1257-1272, 1988

- [28] L. O. Chua , L. Yang "Cellular Neural Networks. Applications" Transactions on Circuits and Systems, vol 36, nr.10, octombrie, pag 1273-1290, 1988
- [29] L. O. Chua and T. Roska "Cellular Neural Networks with Nonlinear and Delay Type Template Elements IEEE International Workshop on Cellular Neural Networks and Their Applications, Proceedings, Budapest, pag.12-25, 1990
- [30] L. O. Chua , T. Roska, P. Venianter, Á. Zarándy, "Some Novel Capabilities of CNN:Game of life and Examples of Multipath Algorithms", IEEE International Workshop on Cellular Neural Networks and Their Applications, Proceedings, Munich, 1992, pag.276-281
- [31] L. O. Chua, T. Roska "The CNN Paradigm" IEEE Transactions on Circuits and Systems I, vol.40, pag.147-156, 1993
- [32] L. O. Chua, T. Roska, T. Kosek, "Cellular Neural Networks-A Tutorial on Programmable Nonlinear Dynamics in Space", Budapest, ", MTA-SZTAKI, Budapesta, DNS-3-1993
- [33] J. M. Cruz, L. O. Chua, T. Roska, "A Fast, Efficient, Complex And Efficient Test Implementation Of The CNN Universal Machine", The Third IEEE International Workshop on Cellular Neural Networks and Their Applications, Roma, Proceedings, 1994
- [34] Gy. Eröss, T. Boros, Á. Kiss, A. Radványi, T. Roska, J. Bito, J. Vass, "Optical Tracking System for Automatic Guided Vehicles Using Cellular Neural Networks", MTA-SZTAKI, Budapesta, DNS-15-1992
- [35] I. Fajfar, F. Bratkovic "Finding All Stable Equilibria of CNN, International Journal of Circuits Theory and Applications, vol.22, pag.145-155, 1994
- [36] I. Fajfar, "On Monotonic Cellular Neural Networks with Nonlinear Template Elements", ECCTD'97, Budapesta, Proceedings, 1997
- [37] G. J. Gibson, S. Siu, C. F. N. Cowan, "The application of nonlinear structures to the reconstruction of binary signals", IEEE Transactions on Signal Processing, no.8, 1991
- [38] M. Ghinea , V. Fireșteanu, " MATLAB Calcul numeric. Grafică-aplicații", București, Edit.Teora, 1995
- [39] L. Goraș, L. O. Chua, D. Leenaerts, "Turing Patterns in CNN's.Part I.Once Over Lightly", IEEE Transactions on Circuits and Systems I, Fundamental Theory and its Applications, vol 42, no.10, pag. 587-611., Oct, 1995
- [40] L. Goraș, L. O. Chua, "Turing Patterns in CNN\_s.Part II.Equations and Behaviors", IEEE Transactions on Circuits and Systems I, Fundamental Theory and its Applications, vol 42, no.10, pag.612-626, Oct, 1995
- [41] L. Goraș, L. O.Chua, L. Pivka, "Turing Patterns in CNN\_s.Part III.Computer Simulations Results", IEEE Transactions on Circuits and Systems I, Fundamental Theory and its Applications, vol 42, no.10, pag.627- 649, Oct, 1995

- [42] S. Grossberg, "Nonlinear neural networks. Principles , Mechanisms and Architectures", Neural Networks, no.1, pag.17-62, 1988
- [43] C. Güzelis "Supervised Learning of The Steady State Outputs in Generalised Cellular Neural Networks" , IEEE International Workshop on Cellular Neural Networks and Their Applications, Proceedings, pag.74-79 , 1992
- [44] C. Güzelis , L. O. Chua "Stability Analysis of Generalized Cellular Neural Networks" International Journal of Circuit Theory and Applications, vol.21, pag.1-33, 1993
- [45] C. Güzelis , S. Karamahmut, "Recurrent Perceptron Learning Algorithm for Completely Stable Cellular Neural Networks", The Third IEEE International Workshop on Cellular Neural Networks and Their Applications, Roma, Proceedings, pag.177-182, 1994
- [46] D. Hammerstrom "Working with Neural Networks", IEEE Spectrum, iulie, , pag.46-53,1993
- [47] H. Harrer "Discrete Time Cellular Neural Networks" PhD thesis, München, iulie, 1992
- [48] H. Harrer , J. A. Nossek, "DiscreteTime Cellular Neural Networks" , International Journal of Circuit Theory and Applications, vol.20, pag.453-467, 1992
- [49] M. H. Hassoun, "Associative Memories", Mcmillan Publishing Co., Englewood Cliffs ,1993
- [50] S. Haykin "Neural Networks" Mcmillan Publishing Co., Englewood Cliffs, 1994
- [51] Hecht-Nielsen "Neurocomputing" Addison-Wesley Publishing Company, 1990
- [52] J. L. Huertas , A.Rodriques-Vasques "VLSI-Implementations of CNN" IEEE International Workshop on Cellular Neural Networks and Their Applications, Proceedings, pag.141-150, 1992
- [53] D. R. Hush, B. Horne, "Progress in Supervised Neural Networks", IEEE Signal Processing Magazine, nr.1, pag.8-39, 1993
- [54] Z. Kiss. P. Szolgay , "Real Time Adaptive Control of Robot Manipulator Using Cellular Neural Network", ECCTD'97, Budapesta, Proceedings, pag.628-633, 1997
- [55] T. Kohonen "Self-organisation and Associative Memory", Springer Verlag, Berlin Heidelberg, 1989
- [56]T. Kosek, T. Roska, L. O. Chua, "Genetic Algorithm for CNN Template Learning" IEEE Transactions on Circuits and Systems II, vol.40, pag.392-402, 1993
- [57] B. Kosko "Neural Networks and Fuzzy Systems", Prentice Hall, Englewood Cliffs, NJ, 1992
- [58] B. Kosko "Neural Networks in Signal Processing", Prentice Hall, Englewood Cliffs, NJ, 1992
- [59] Lanchlan. L. Andrew, "Improving the robustness of winner take all celular neural networks", IEEE Transactions on Circuits and Systems II, vol 43, no.4, pag.295-307. April, pag.329-334, 1996

- [60] D. Liu, A. N. Michel "Cellular Neural Networks for Associative Memories" IEEE Transactions on Circuits and Systems II, vol , feb., pag.119-121, 1993
- [61] D. Liu and A. N. Michel "Sparsely Interconnected Neural Networks for Associative Memories with Applications to Cellular Neural Networks" IEEE Transactions on Circuits and Systems II, vol 41, no.4, pag.295-307, April, 1994
- [62] D. Liu, "Cloning Template Design of Cellular Neural Networks for Associative Memories", IEEE Transactions on Circuits and Systems I, Fundamental Theory and its Applications, vol 44, no.7, pag.646-650, July, 1997
- [63] T. Lehmann, E. Brun "Analogue VLSI Implementation of Back-Propagation Learning In Artificial Neural Networks ", ECCTD '93, - Circuit Theory and Design, Proceedings, pag. 491- 493, 1993
- [64] H. Magnussen. J. Nossek, "Global Learning Algorithms for Discrete-Time Cellular Neural Networks" The Third IEEE International Workshop on Cellular Neural Networks and Their Applications, Roma, Proceedings, pag.165-170, 1994
- [65] H. Magnussen, G. Papoutsis, J. Nossek, "Continuation-Based Learning Algorithm for Discrete-Time Cellular Neural Networks", The Third IEEE International Workshop on Cellular Neural Networks and Their Applications, Roma, Proceedings, pag.171-176, 1994
- [66] G. Martinelli, R. Perfetti "Associative Memory Design Using Space Varying Neural Networks"IEEE International Workshop on Cellular Neural Networks and Their Applications, Proceedings, pag.117-122, 1992
- [67] G. Martinelli, R. Perfetti "Estimating The Storage Capacity Of Space Varying Cellular Neural Networks", Electronic Letters ,1994
- [68] Mladenov, Leenaerts, Uhlmann "First Order Estimation of The Basin of Attraction of Stable Equilibrium Points in Cellular Neural Networks", ECCTD '97, Proceedings,pag.684-688, 1997
- [69] M. Milic, P. Kostic, "Solving Systems of Linear Equations Using Cellular Neural Networks", ECCTD '93, -Circuit Theory and Design, Proceedings, pag.671-676,1993
- [70] J. Miller, T. Roska, T. Szirányi, K. Crouse, L. Chua, L. Nemes, "Deblurring of images by Cellular Neural Networks with Applications to Microscopy", The Third IEEE International Workshop on Cellular Neural Networks and Their Applications, Roma, Proceedings,1994
- [71] T. Matsumoto, T. Yokohama, H. Suzuki, R. Furukawa, L. Chua, "Several Image Processing Examples by CNN" , IEEE International Workshop on Cellular Neural Networks and Their Applications, Budapest, Proceedings, pag.100-112. 1990
- [72] H. Mizutani "A New Learning Method for Multilayered Cellular Neural Networks". The Third IEEE International Workshop on Cellular Neural Networks and Their Applications, Roma, Proceedings, pag.195-198 , 1994

- [73] P. Nachbar, A. J. Schuler, T. Füssl, J. A. Nossek, L. O. Chua, "Robustness of Attractor Networks and an Improved Corner Detector", IEEE International Workshop on Cellular Neural Networks and Their Applications, München, Proceedings, pag.55-61, 1992
- [74] I. Nafoarniță, "Studii si cercetari privind tehnicile de inteligență artificială aplicabile in domeniul de interes militar, ICSITA Bucuresti , raport de cercetare , 1986
- [75] I. Nafoarniță, "Studii si cercetări privind tehnicile de inteligență artificiala si a rețelelor neuronale in domeniul de interes militar, ICSITA Bucuresti , raport de cercetare fundamentala, nr.40, 1989
- [76] L. Nemes, G. Tóth, T. Roska, A. Radványi, "Analogic CNN Algorithms for 3D Interpolation, Aproximation and Object Rotation using Controlled Switched Templates" , MTA-SZTAKI, DNS-1994
- [77] J. A. Nossek, L. Chua "Design and Learning with Cellular Neural Networks" IEEE International Workshop on Cellular Neural Networks and Their Applications, Rome, Proceedings, pag. 137-146, 1994
- [78] R. Perfetti, "Cellular Neural Network for Fast Adaptive Equalization" , 1992
- [79] M. Ritchie, B.W.Kernigham, "The C Programming Language", New Jersey, 1972
- [80] T. Roska, T. Boros, P. Thiran and L. Chua "Detecting Simple Motion Using Cellular Neural Networks" IEEE International Workshop on Cellular Neural Networks and Their Applications, Proceedings, pag.127-138, 1990
- [81] T. Roska "Programmability and Applications of the CNN Universal Machine" IEEE International Workshop on Cellular Neural Networks and Their Applications, München, , Proceedings, pag.181-190 ,1992
- [82] T. Roska "Programmable Cellular Neural Networks .A State- of- the Art" Internal Report, MTA-SZTAKI, 1992
- [83] T. Roska, L. O. Chua "The CNN Universal Machine. An Analogic Array Computer, " IEEE Transactions on Circuits and Systems II, vol.40, pag.163-173, 1993
- [84] T. Roska, L. O. Chua, A. Zarandy, "Language, Compiler and Operating System for the CNN Supercomputer, UCB/ERL M93/34 Berkeley,1993
- [85] T. Roska , G.Bartfay, P.Szolgay, T.Sziranyyi, A.Radvanyi, .T.Kosek, Zs.Ugray, A.Zarandy " A Hardware Accelerator Board for Cellular Neural Neworks:CNN-HAC", International Journal Circuit Theory and Applications, vol.20, pag.589-599, 1992
- [86] T. Roska and J. Vandewalle "Cellular Neural Networks" John Wiley & Sons Ltd., 1993
- [87] F. R. Savaci and J. Vandawalle "On the Stability Analysis of CNN" IEEE International Workshop on Cellular Neural Networks and Their Applications. Proceedings, pag.240- 245, 1992



- [88] A. J. Schuler, P. Nachbar, J. A. Nossek, "State-Based Backpropagation-Through-Time for CNNs", ECCTD'93-Circuit Theory and Design, Proceedings, pag.33-37, 1993
- [89] V. Sima, "Metode noi în matematica aplicată", Editura științifică, București, 1992
- [90] G. Seiler, A. Schuler, J. Nossek, "Design of Robust Cellular Neural Networks", Internal Report: TUN LNS-TR 91-13, sept., 1991
- [91] B. Siemiatkowska "Cellular Neural Network for Mobile Robot Navigation", Third IEEE International Workshop on Cellular Neural Networks and Their Applications, Roma, Proceedings, pag.285-290, 1994
- [92] B. E. Shi, S. Wendsche, T. Roska, L. O. Chua "Random Variations in CNN Templates: Theoretical Models and Empirical Studies", Third IEEE International Workshop on Cellular Neural Networks and Their Applications, Roma, Proceedings, pag.27-32, 1994
- [93] T. Szirányi, M. Csapodi, "Texture Classification by CNN Using Genetic Learning" MTA-SZTAKI, Budapesta, DNS-8-1993
- [94] P. Szolgay, I. Kispál, T. Kosek, "An Experimental System for Optical Detection of Layout Errors of Printed Circuit Boards Using Learned CNN Templates", IEEE International Workshop on Cellular Neural Networks and Their Applications, Munchen, Proceedings, pag.203-209, 1992
- [95] P. Szolgay, K. László, I. Szatmári "A New Learning Algorithm to Implement Associative Memory on CNN, MTA-SZTAKI, Budapesta, DNS-8-1995
- [96] H. Takahashi, E. Tomita, "Estimation Of Learning Curves In Learning Neural Networks From Noisy Samples", International Symposium on Nonlinear Theory and its Applications", NOLTA '93, Proceedings, pag.47-50, 1993
- [97] S. Tan, J. Hao and J. Vandewalle "Cellular Neural Networks as a Model of Associative Memories", IEEE International Workshop on Cellular Neural Networks and Their Applications, Proceedings, pag.26-35, 1990
- [98] M. Tanaka, K. Crouse, T. Roska, "Template Synthesis of Cellular Neural Networks for Information Coding and Decoding", IEEE International Workshop on Cellular Neural Networks and Their Applications, München, Proceedings, pag.29-35, 1992
- [99] P. Thiran "Two Causes of Instability of Cellular Neural Networks", IEEE International Workshop on Cellular Neural Networks and Their Applications, München, Proceedings, pag.100-105, 1992
- [100] P. Thiran, M. Hasler, "Information Processing Using Stable and Unstable Oscillations:a Tutorial" IEEE International Workshop on Cellular Neural Networks and Their Applications, Roma, Proceedings, pag.127-136, 1994
- [101] W. Utshick, J. A. Nossek, "Computational Learning Theory Applied to Discrete-Time Cellular Neural Networks", IEEE Third International Workshop on Cellular Neural Networks and Their Applications, Roma, Proceedings, pag.159-164, 1994

- [102] P. Venianter, A. Radványi, T. Roska, "ACL (an Analogical CNN Language", Version2.0, MTA-SZTAKI Budapesta, DNS-3-1994
- [103] P. Venianter, P. Szolgay, K. Crounse, T. Roska, L. Chua, "Analogic Combinatorics and Cellular Automata Key Algorithms and Layout Design"
- [104] P. Venianter, F. Werblin, T. Roska, L. Chua, "Analogic CNN Algorithms for Some Image Compression and Restoration Tasks, UCB-ERL Memo, Berkeley, 1994
- [105] J. Vasques-Rodriques, S. Espejo, "On the Design of Programmable Cellular Neural Networks, ISCAS'94, Londra, pag292-298, 1994
- [106] G. Yen, N. Michel "A Learning and Forgetting Algorithm in Associative Memories: The Eigenstructure Method", " IEEE Transactions on Circuits and Systems II, vol.39, aprilie, pag.212-224,1992
- [107] Á. Zarándy, F. Werblin, T. Roska, L.Chua, "Novel Types of Analogic CNN Algorithms for Recognizing Bank-notes", Berkeley Memo, 1994
- [108] Á. Zarándy, T. Roska, Gy. Lisyka, J. Hegyesi, L.Kék, Cs.Rekeczky, "Design of Analogic Algorithms for Mammogram Analysis"
- [109] S. I. Zăhan, "Aplicații ale Teoriei Mulțimilor Fuzzy în Telecomunicații și Diagnoză medicală", Teză de doctorat, Universitatea Tehnică Cluj Napoca, 1997
- [110] F. Zou, S. Schwarz and J. A. Nossek "Cellular Neural Network Design Using a Learning Algorithm", IEEE International Workshop on Cellular Neural Networks and Their Applications, Proceedings, pag.73-81, 1990
- [111] I. Zurada " Introduction to Artificial Neural Systems. Learning and Applications", West Publishing Co., Wien, 1992
- [112] \*\*\* "CNN Workstation V.5.1 User's Manual", MTA-SZTAKI, Budapest, 1992
- [113] \*\*\* Analogic CNN Program Library, V 6.1, MTA-SZTAKI Budapest, DNS-5-1994
- [114] \*\*\* "CNNM , Multi-Layer Cellular Neural Network Simulator , V 5.3x, User's Manual" MTA-SZTAKI, Budapest, 1994
- [115] \*\*\* "Application of Spacevariant Templates in the CNNM Simulator", User's Manual"MTA- SZTAKI, Budapest, 1994
- [116] \*\*\* "Student Edition of Mathlab V4, User's Guide, Prentice Hall, Englowood Cliffs, NJ, 1995.