

**Academia
Oamenilor de Știință
din România**



**Academy
of Romanian
Scientists**

Add: Splaiul Independenței nr. 54 sector 5, 050094, București, ROMANIA, Cod Fiscal: 5091859
Tel. 00-4021/314.74.91; Fax. 00-4021/314.75.39, Web-site: www.aos.ro, E-mail: aosromania@yahoo.com



UTILIZAREA INTELIGENȚEI ARTIFICIALE ÎN PREDICȚIA DATELOR DIN SISTEMELE AUTOMATIZATE INDUSTRIALE

RAPORT 1 - 29.06.2018

Autor:

CS III dr. ing. Daniel-Petru **GHENCEA**

Coordonator:

Prof. univ. dr. ing. Miron **ZAPCIU**

CUPRINS

REȚELE NEURONALE ARTIFICIALE	2
1 Introducere.....	2
2 Conceptul de Rețele Neuronale Artificiale	2
2.1 Neuronul artificial static	4
2.2 Funcții de activare a neuronului	6
3 Modele de neuroni artificiali	7
4 Tipuri și algoritmi de instruire	10
5 Rețele neuronale recurente	11
5.1 RNA cu propagare înapoi.....	11
5.2 RNA total recurente de tip Hopfield	12
5.3 RNA parțial recurente de tip Elman	14
6. Regresia liniară simplă și corelația.....	15
BIBLIOGRAFIE	18

REȚELE NEURONALE ARTIFICIALE

1 Introducere

Încercarea de a explica despre cum ar funcționa neuronii din creierul uman a fost realizată de W.McCulloch (neurofiziolog) și W.Pitts (matematician). D.Hebb în cercetările efectuate ulterior a sesizat faptul că doi nervi care se declanșează simultan legătura dintre ei se îmbunătățește [1].

Modelul RNA ADELIN (ADaptive LINear Elements), (B.Widrow și M.Hoff – 1959), a fost realizat pentru a fi capabil să recunoască modele binare iar RNA MADELINE (Multiple ADaptive LINear Elements) a fost prima rețea neuronală aplicată unei probleme practice.

O problema specială, în dezvoltarea RNA, a constituit-o gestionarea erorilor, problemă rezolvată prin modelul de recunoaștere al erorilor care este distribuit în întreaga rețea și creșterea numărului de straturi (peste două straturi), model denumit în prezent RNA cu propagare înapoi (back propagation networks). Modelul de recunoaștere al erorilor este realizat astfel încât nodurile finale de ieșire sunt conectate cu nodurile anterioare având caracteristicile unui sistem extrem de neliniar cu reacție (feedback) [2].

Cu ajutorul RNA pot fi realizate prognoze pentru probleme de clasificare, de căutare, de ordonare, probleme de recunoaștere a formelor obiectelor, înțelegerea vederii și a vorbirii, probleme care tratează date imperfecte, incomplete, vagi (fuzzy) sau probabilistice.

2 Conceptul de Rețelele Neuronale Artificiale

Rețelele neuronale artificiale (RNA) sunt recunoscute ca modele dominante ale Inteligenței Artificiale (IA) [3].

Caracteristicile RNA [3], [4], [5]:

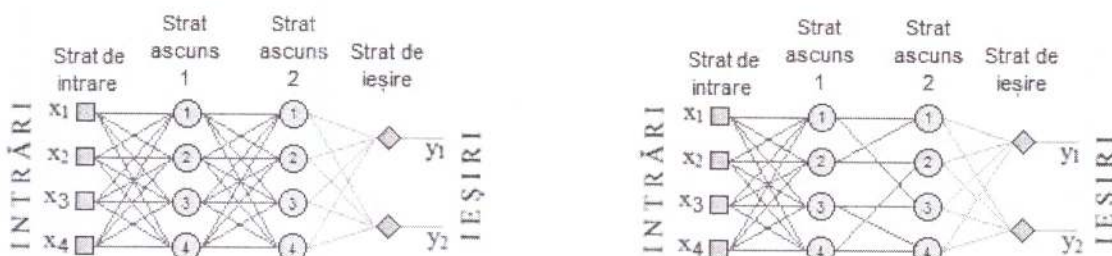
- ✓ *învățarea și adaptarea* sunt cele mai importante proprietăți;
- ✓ *distribuția informației în structura*: informația din rețea este stocată în mod distribuit făcând ca efectul unei anumite intrări asupra ieșirii să depindă de restul ponderilor din rețea.
- ✓ *capacitatea de generalizare* în cazul unor situații neconținute în datele de instruire. Aceasta caracteristică depinde de numărul de ponderi, adică de dimensiunea rețelei. Creșterea dimensiunii rețelei duce la o bună memorare a datelor de instruire, dar scad performanțele asupra datelor de testare fapt ce conduce la pierderea capacității de generalizare de către RNA. Stabilirea numărului optim de straturi ascunse și a numărului de neuroni din fiecare strat ascuns, care este o etapă importantă în proiectarea unei RNA, se realizează alegând valoarea de la care începe să descrească performanța RNA pe setul de testare.
- ✓ *toleranța la erori*: RNA pot fi instruite, chiar dacă datele sunt afectate de erori, diminuându-se performanța ei.

- ✓ rezistența la distrugerea parțială: datorită reprezentării distribuite a informației, RNA poate opera și în cazul distrugerii unei mici părți a ei.
- ✓ viteză în calcul: RNA consuma mult timp pentru instruire, dar odată antrenate și validate datele vor calcula rapid ieșirea rețelei pentru o anumită intrare.
- ✓ sisteme multivariabilă sunt ușor de modelat cu RNA datorită numărului mare de intrări și ieșiri;

Principalele tipuri de RNA sunt: tip perceptron, bazate pe funcții radiale, recurente și cu auto-organizare.

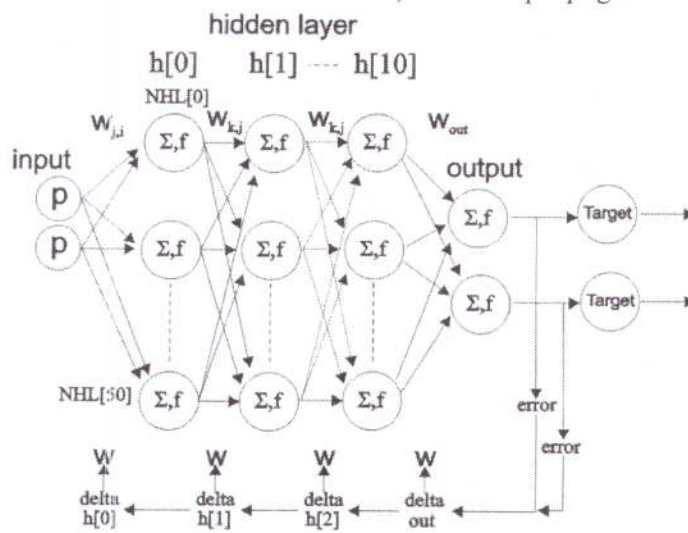
Principalele tipuri de arhitecturi RNA sunt (Figura 1.1):

- RNA feedforward (cu propagare înainte) – total sau parțial conectate;
- RNA feedback (recurente), cu propagare înapoi (backpropagation));



a) RNA cu propagare înainte total conectată

b) RNA cu propagare înainte parțial conectată



c) RNA cu propagare înapoi

Figura 1. 1 Tipuri de arhitecturi RNA

Sursa: [6], [7]

Există o mare diversitate de opinii în ceea ce privește modul de clasificare al algoritmilor și tipurilor de învățare.

Perceptronul multistrat (RNA-MLP, Multilayer Perceptron) reprezintă o generalizare a perceptronului simplu fiind o RNA de tip feedforward (cu propagare înainte a semnalului) compusă din (Figura 1.2):

- ✓ strat de intrare;
- ✓ unul sau mai multe straturi ascunse;
- ✓ strat de ieșire.

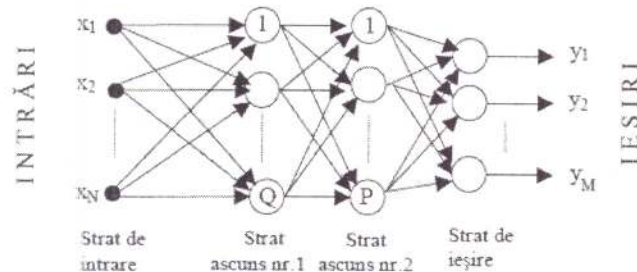


Figura 1. 2 Perceptron cu două straturi

Sursa: [6]

Numărul optim de straturi ascunse și de neuroni/strat ascuns este dificil de precizat apriori [6].

Există anumite estimări, cum ar fi cele ale lui Kolmogorov (1957), care precizează că pentru aproximarea unei funcții de n variabile, ar fi necesari $n \cdot (2 \cdot n + 1)$ neuroni în primul strat ascuns și, în cazul utilizării a două straturi ascunse $(2n+1)$ neuroni. Cercetări mai recente au arătat însă că aceste estimări nu conduc întotdeauna la o soluție optimă [8].

De obicei, un singur strat ascuns e suficient pentru rezolvarea majorității problemelor. În mod excepțional, se pot folosi două, cel mult trei straturi ascunse. De regulă, numărul de neuroni aferenți straturilor de intrare respectiv ieșire este dictat de natura aplicației. Neuronii structurilor ascunse au rolul de a *detecta trăsăturile*, legitățile, regularitățile conținute în tiparele de antrenament.

Un număr prea mare de neuroni ascunși/strat influențează în mod negativ capacitatea de generalizare a RNA. Totodată conduce la sporirea volumului de date care urmează a fi procesat și deci la o durată sporită pentru etapa de instruire. Un număr prea mic de neuroni nu este suficient pentru formarea unei reprezentări interne a datelor adecvată și poate conduce la o eroare medie pătratică mare pe parcursul epocilor de instruire și implicit la o eroare mare corespunzătoare nu numai datelor de test ci și celor de instruire.

Concluzie 1.1: numărul optim de neuroni ascunși se va determina experimental.

2.1 Neuronul artificial static

Neuronul static este cel mai simplu model de neuron artificial în care ieșirea depinde doar de intrare [9]. Sistemul liniar static intrare/ieșire este de forma:

$$y = a \cdot u \quad a = ct. \quad (3.1)$$

Dacă extindem sistemul la intrări și ieșiri multiple putem utiliza notația vectorială:

$$Y = A \cdot U \quad (3.2)$$

unde:

- ∞ Y, U sunt vectorii de ieșire și intrare;
- ∞ A - este matricea de transfer (transpusă);

Pentru sistemul static din figura 3.3. avem relația:

$$y = u_1 \cdot w_1 + u_2 \cdot w_2 + u_3 \cdot w_3 \quad (3.3)$$

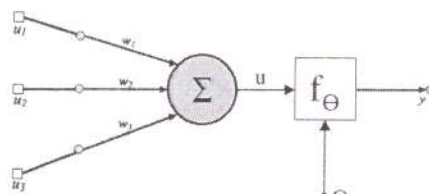


Figura 1. 3 Sistem static liniar

Sursa: [9]

Neuronul artificial prezentat în Figura 1.3 este format din intrările u_1, u_2, u_3 fiecare fiind caracterizată de propria pondere sinaptică w_1, w_2, w_3 . Unificarea intrărilor ponderate se realizează de către sumator iar suma obținută u (numită intrare netă) se aplică unei funcții de activare care are ca rezultat ieșirea neuronului y . **Funcția de activare** este conectată la o sursă de semnal, θ (polarizare) care arată starea inițială a neuronului [10] și simulează timulii exteriori.

$$y = f(u - \theta) \quad (3.4)$$

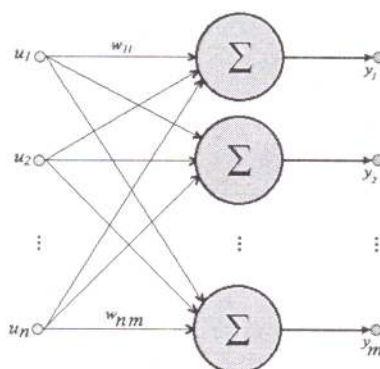


Figura 1. 4 Rețea neuronală artificială liniar asociativă

Sursa: [9]

În cazul în care avem n ieșiri, deci și n neuroni putem realiza RNA prezentată în Figura 1.4 iar relația dintre ieșiri și intrări este:

$$y_j = \sum_i w_{ij} u_i \quad (3.5)$$

Pentru m ieșiri și n intrări se poate scrie între intrare-ieșire relația următoare:

$$Y = W^T \cdot U \quad (3.6)$$

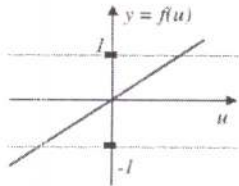
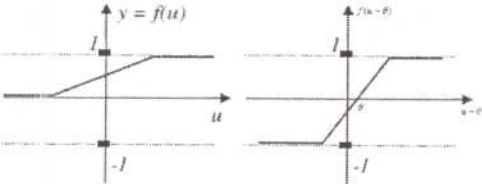
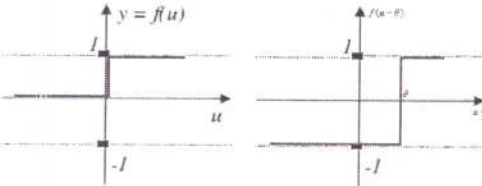
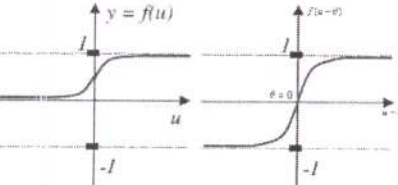
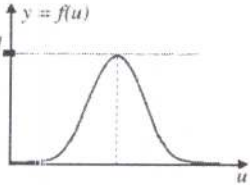
sau sub formă matricială:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \quad W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix} \quad W^T = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix} \quad (3.7)$$

2.2 Funcții de activare a neuronului

Funcția de activare a neuronului are rolul de a limita domeniul de variație al ieșirii la un domeniu prespecificat. Funcțiile de activare pot fi monotone (foarte utilizate) sau nemonotone (care conduc la performanțe foarte bune) [11]. Funcțiile de activare monotone cele mai utilizate sunt prezentate în Tabelul 1.1 [10], [11], [12], [13].

Tabelul 1.1 Funcțiile de activare monotone

Denumire funcție	Reprezentare grafică	Ecuatie
LINIARĂ		$f(u) = k \cdot u$
RAMPĂ (unipolar și bipolar)		$f(u) = \begin{cases} 0, & u \leq 0 \\ u/k, & 0 < u \leq k \\ 1, & u > 0 \end{cases}$
PRAG (unipolar și bipolar)		$f(u) = \begin{cases} 0, & u \leq 0 \\ 1, & u > 0 \end{cases}$
SIGMOIDALĂ (unipolar și bipolar)		$f(u) = \frac{1}{1 + e^{-k \cdot u}}$
GAUSS		$f(u) = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma}} \cdot e^{-\frac{1}{2} \left(\frac{u - \mu}{\sigma} \right)^2}$

Sursa: [10], [11], [12], [13]

Funcții de activare non monotone, care conduc la performanțe foarte bune, în special la memoriile asociative, sunt prezentate în Figura 1.5 [11], [14], [15].

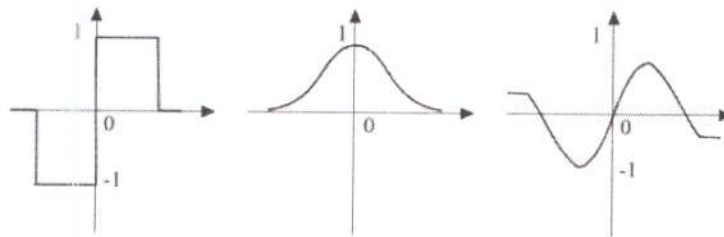


Figura 1. 5 Funcții de activare nonmonotone
Sursa [11], [14], [15]

Sincronizarea impulsurilor este importantă pentru a explica modul în care se realizează calculele la nivel de neuron [16]. Pentru a se obține precizie în sincronizare se utilizează modele cu **neuroni pulsanți** în locul modelelor tradiționale bazate pe rata impulsurilor fapt ce a condus la dezvoltarea simulatoarelor de rețele neuronale artificiale pulsative (RNAP). Acestea oferă utilizatorului posibilitatea să obțină simulări precise ale unui sistem computațional dat și rezultate într-un timp relativ scurt. RNAP utilizează modele bazate pe conductanță sau modele simple de tipul „integrează și activează” (integrate-and-fire – IF) care sunt atractive pentru simularea rețelelor neuronale la scară mare.

Concluzie 1.2: Controlarea unui rețele neuronale artificiale (RNA) de către o rețea neuronală artificială pulsativă (RNAP) conduce la conturarea conceptului de hiper rețea neuronală artificială (HRNA) sau RNA de gradul doi.

3 Modele de neuroni artificiali

A. Neuronul McCulloch – Pitts (MCP) este primul model de neuron artificial cunoscut și sub numele de unitate prag (Threshold Unit). Neuronul primește pe fiecare conexiune de intrare un semnal binar (0 sau 1) și emite la ieșire tot un semnal binar. Conexiunile de intrare pot fi: inhibatoare ($w_j < 0$) și excitatoare ($w_j > 0$) [3], [17]. Modelul neuronului artificial sumează cele n intrări ponderate (ponderi fiind stabilite de către rețea în timpul procesului de învățare prin algoritmi specifici), apoi sumei obținute i se aplică o funcție de activare (polarizare) și trimite rezultatul printr-o neliniaritate la ieșire.

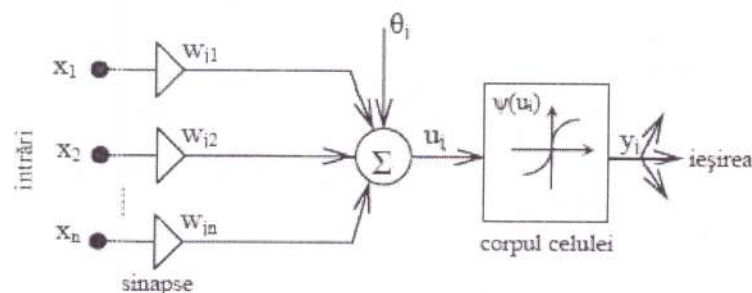


Figura 1. 6 Modelul structural al neuronului McCulloch – Pitts
Sursa: [6]

Neuronul artificial formal este caracterizat de neliniaritatea și offset (θ_j). Modelul utilizează ca funcție de activare funcția prag binară, care poate fi înlocuită de o funcție neliniară mai generală

iar la ieșirea neuronului poate lua valori într-un set discret $\{-1,1\}$ sau $\{0,1\}$ sau poate varia continuu, între două valori limită y_{min} și y_{max} , cu condiția $y_{max} > y_{min}$.

B. **Neuronul dinamic** are un rol esențial în studiul sistemelor neuronale fiind capabile să primească feedback de la alți neuroni, deci activitatea sa este controlată prin compararea performanței sale efective cu performanțe testate [18], [19]. Unitățile neuronale dinamice (DNU), elementele de bază ale rețelelor neuronale dinamice, primesc nu numai intrări externe, ci și semnale de feedback de la ei și alți neuroni. Conexiunile sinaptice dintr-un DNU conțin o conexiune auto-recurentă care reprezintă un semnal de feedback ponderat al conexiunilor sale de stare și de inhibare laterală, care sunt semnale de feedback de stare de la alte DNU din rețea. Prelucrarea informației cu DNU se bazează pe o prelucrare a cunoștințelor anterioare și stochează informații actuale pentru utilizarea ulterioară. Fiecare DNU are propriul său potențial intern sau stare internă care este folosit pentru a descrie caracteristicile dinamice ale rețelei.

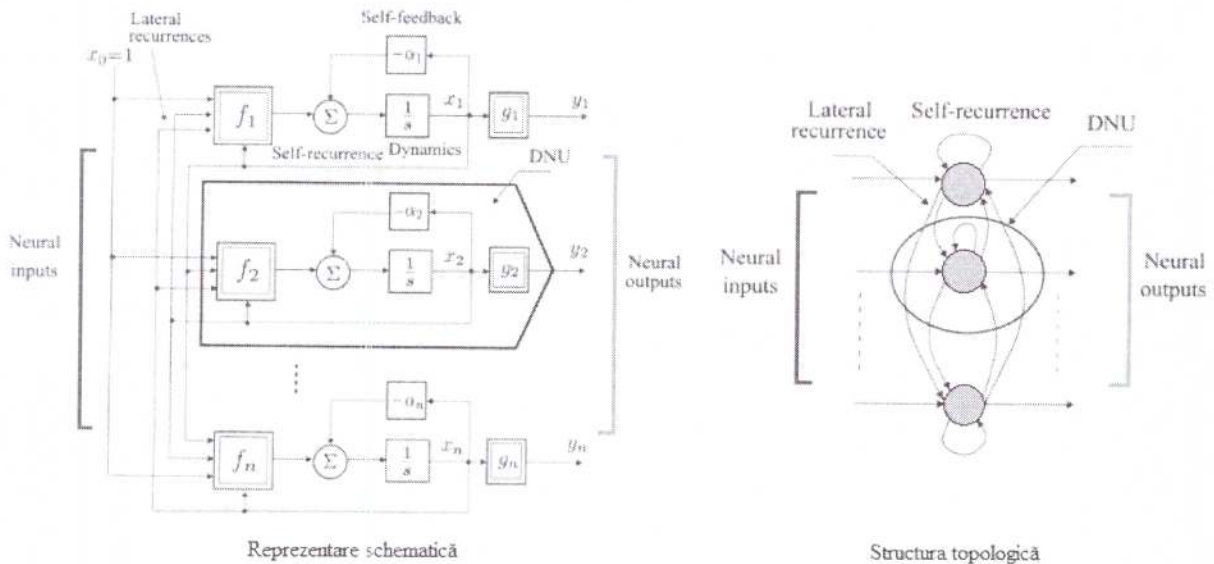


Figura 1. 7 Conectarea unității dinamice neuronale (DNU) în rețeaua neuronală artificială
Sursa: [18].

C. **Neuronul Fukushima** – este caracterizat de faptul că ponderile sinaptice pot lua valori pozitive, negative sau zero. La acest model toate ponderile și toate semnalele de intrare/ieșire sunt nenegative iar intrările și ponderile sinaptice corespunzătoare sunt separate în două grupe: excitatorii a_{ji} și inhibitorii b_{ji} .

Ieșirea neuronului este descrisă de relația:

$$y_j = \Psi \left(\frac{1 + \sum_{i=1}^n a_{ji} \cdot x_i}{1 + \sum_{i=1}^m b_{ji} \cdot v_i} - 1 \right) \quad \text{unde} \quad \Psi(u_j) = \begin{cases} u_j, & u_j \geq 0 \\ 0, & u_j < 0 \end{cases} \quad (3.11)$$

Ponderile sinaptice sunt în general variabile pe parcursul procesului de autoorganizare a RNA.

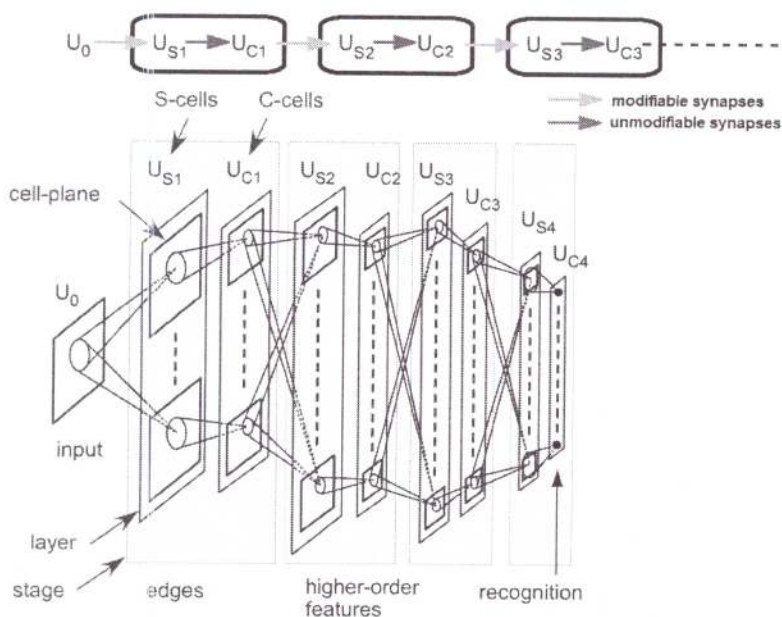


Figura 1. 8 Structura de rețea ierarhică a Neocognitronului
Sursa: [20], [21], [22]

K. Fukushima a elaborat un sistem de calcul neuronal pentru recunoașterea imaginilor, cu aplicabilitate practică în domeniul recunoașterii caracterelor [23]. Rețeaua neuronală construită, are la baza un sistem performant de recunoaștere a formelor, numit **Neocognitron** (Figura 1.8).

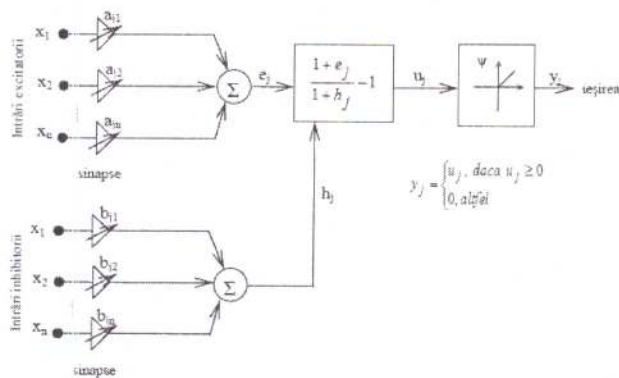


Figura 1. 9 Modelul structural al neuronului Fukushima
Sursa: [3]

Neocognitronul este o RNA cu mai multe straturi ce simulează modul de prelucrare a imaginilor de către cortexul uman. Straturile ascunse succesive de neuroni ale Neocognitronului au rolul de a extrage trăsături definitorii ale imaginii fără a fi influențate de orientare sau distorsiuni. La nivelul stratului de intrare formele sunt unic determinate, o dată cu propagarea informației către stratul de ieșire, activându-se doar anumiți neuroni, care corespund unor trăsături definitorii ale imaginii.

D. Neuronul ADALINE (Adaptive Linear Element) poate fi antrenat pentru a realiza diferite funcții logice, ajustând în mod adecvat ponderile sinaptice. Deoarece rețeaua este de tip liniar, are un număr de aplicații restrâns, dar este avantajată de gama largă de algoritmi de învățare

performanți existenți [11]. ADALINE este capabil să realizeze o gamă redusă de funcții logice - funcții logice liniar separabile (AND, NOT, OR). Diferența dintre neuronul ADALINE și perceptronul standard (McCulloch-Pitts) o reprezintă momentul din faza de învățare când ponderile sunt ajustate în funcție de suma ponderată a intrărilor.

4 Tipuri și algoritmi de instruire

I. **Învățarea de tip supervizat.** Este un algoritm care cunoaște cu exactitate modul de asociere al intrărilor RNA cu ieșirile acestora și realizează un model pentru a genera predicții rezonabile pentru răspunsul la date noi, deci RNA emulează algoritmul [23].

II. **Învățarea prin întărire (reinforcement).** Este un caz particular al învățării supervizate cu diferența că instruirea semnalează faptul că răspunsul generat de RNA este corect sau nu, fără a cunoaște mărimea acestuia. Acest tip de învățare urmărește maximizarea mărimi scalare (indice de performanță sau semnal de întărire) în urma unei acțiuni efectuate de către sistemul supus învățării. Dacă modificările aduse conduc spre o stare mai bună decât cea precedentă, tendința sistemului de a produce acea acțiune particulară este întărită.

III. **Învățarea de tip nesupervizat (cu autoorganizare).** Este caracterizată de absența unui semnal sau supervisor care să aprecieze corectitudinea asociațiilor intrare-ieșire, deci se bazează pe setul de date disponibil [11]. RNA va descoperi singură legăturile conținute în datele de intrare printr-o reprezentare internă adecvată a trăsăturilor vectorului de intrare, deci organizează singură informația, motiv pentru care acest tip de RNA sunt mai puțin complecși și mai imprecizi. Avantajul învățării nesupervizate este viteza de procesare (în timp real) și învățarea din set de date afectate de zgomot.

IV. **Algoritmi de învățare bazați pe corecția erorii.** Scopul algoritmilor bazați pe corecția erorii este de a minimiza **funcția de cost** utilizând ca și criteriu **eroarea pătratică medie** - care urmărește minimizarea valorii medii pătratice pentru suma erorilor pătratice aferente stratului de ieșire al RNA.

V. **Algoritmi de învățare de tip Boltzmann.** La acest tip de RNA neuronii constituie o structură recurentă caracterizată de funcția energie:

$$E = -\frac{1}{2} \cdot \sum_i \sum_j w_{ji} \cdot s_j \cdot s_i \quad (3.18)$$

unde s_i reprezintă starea neuronului i , adică $+1 = neuron_activ$ sau $-1 = neuron_inactiv$.

VI. **Algoritmi de învățare de tip hebbian.** Conform postulatului lui Hebb, modificarea ponderii sinaptice w_{kj} este dependentă de activitatea presinaptică și postsinaptică.

VII. **Algoritmul de învățare de tip competitiv.** Este caracterizat de **competiția** între neuronii de ieșire ai RNA, câștigătorul acestora urmând să fie activat. Spre deosebire de RNA care se bazează pe algoritmi de învățare de tip hebbian și la care există posibilitatea ca mai mulți neuroni să fie activi simultan, la RNA bazate pe algoritmi de învățare de tip competitiv doar un singur neuron este activ la un moment dat.

5 Rețele neuronale recurente

5.1 RNA cu propagare înapoi

RNA cu mai multe straturi sunt capabile să computeze o gamă mai largă de funcții booleene decât RNA cu un singur strat, dar creșterea gradului de complexitate al setului de date conduce la mărirea timpului de a găsi combinația corectă de ponderi [24].

Cum învață RNA

RNA reprezintă un lanț de compoziții de funcții care transformă o intrare într-un vector de ieșire (numit model). Rețeaua reprezintă o implementare specială a unei funcții compuse din spațiul de intrare în spațiul de ieșire, pe care o numim *funcția rețelei*. Problema de învățare constă în găsirea unei combinații optime de ponderi, astfel încât funcția de rețea φ aproximeze o funcție dată f cât mai fidel posibil. Funcția f nu este dată în mod explicit, ci doar implicit prin câteva exemple reprezentate de setul de date de intrare. Considerăm o RNA cu propagare înainte cu n intrări și f ieșiri. Acesta poate conține un număr oarecare de unități ascunse și poate prezenta orice tip de conexiune dorită. Se consideră setul de antrenament $\{(x_1, t_1), \dots, (x_p, t_{p1})\}$ constând din p ordonate de perechi de vectori n - și m -dimensionali, numiți modele de intrare și ieșire. Fie ca funcțiile primitive la fiecare nod al rețelei să fie continue și diferențiate. Ponderile marginilor sunt numere reale alese aleator. Atunci când modelul de intrare x_i din setul de instruire este prezentat RNA, acesta produce o ieșire o_i diferită în general față de țintă t_i . Dorim ca o_i și t_i să fie identici pentru $i = 1, \dots, p$ folosind un algoritm de învățare, prin minimalizarea funcției de eroare a RNA, definită ca:

$$E = \frac{1}{2} \cdot \sum_{i=1}^p \|o_i - t_i\|^2 \quad (3.31)$$

După minimizarea acestei funcții pentru setul de instruire, noi modele de intrare necunoscute sunt prezentate RNA și ne așteptăm ca aceasta să interpoaleze prin faptul că trebuie să recunoască dacă un nou vector de intrare este similar cu modelele învățate și să produce o ieșire similară.

Algoritmul backpropagation este folosit pentru a găsi un minim local al funcției de eroare. Rețeaua este inițializată cu ponderi alese aleatoriu. Gradientul funcției de eroare este calculat și utilizat pentru a corecta ponderile inițiale deci acest gradient recursiv trebuie calculat.

Astfel putem minimiza E folosind un proces iterativ descendent a gradientului, prin calcularea gradientului:

$$\nabla w_i = \left(\frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_i} \right) \quad (3.32)$$

Fiecare pondere este actualizată cu ajutorul incrementului:

$$\Delta w_i = -\gamma \cdot \frac{\partial E}{\partial w_i} \quad i = 1, \dots, l \quad (3.33)$$

unde γ reprezintă constanta de învățare, deci este un parametru de proporționalitate care definește lungimea treptei fiecărei iterații în direcția gradientului negativ.

Etapele algoritmului de backpropagation

În această etapă formulăm algoritmul propagare înapoi complet și demonstrăm prin inducție că funcționează în rețele arbitrare de transmitere cu funcții de activare diferențiate la noduri în cazul în care avem o RNA cu o singură intrare și o singură unitate de ieșire.

Algoritmul de backpropagation

Se consideră o RNA cu o singură intrare reală x și funcția de rețea F . Derivata F este calculată în două etape:

- 1) *propagare înainte (feed-forward)*: intrarea x este introdusă în rețea. Funcțiile primitive la noduri și derivatele lor sunt evaluate la fiecare nod, derivatele fiind stocate.
- 2) *propagare înapoi (backpropagation)*: constanta 1 este introdusă în unitatea de ieșire și rețeaua este instruită înapoi. Se adaugă informațiile de intrare către un nod, iar rezultatul este înmulțit cu valoarea stocată în partea stângă a unității. Rezultatul colectat la unitatea de intrare este derivata funcției de rețea în raport cu x și este transmis în stânga unității.

5.2 RNA total recurente de tip Hopfield

RNA de tip Hopfield (Figura 1.10) reprezintă rețele recurente, simetrice, total conectate și fără autoasocieri [25]. RNA de tip Hopfield a fost realizată astfel încât să nu fie necesară nici o sincronizare, fiecare unitate având un fel de sistem elementar în interacțiune complexă cu restul ansamblului [26].

Simetria conexiunilor se exprimă prin egalitatea:

$$w_{ij} = w_{ji} \quad (3.34)$$

unde i, j reprezintă unități din rețea, iar w matricea ponderilor conexiunilor din rețea.

Lipsa autoasocierii se exprimă prin:

$$w_{ij} = w_{ji} \quad (3.35)$$

O rețea neuronală recurentă se află într-o stare stabilă atunci când neuronii din rețea acționează unii asupra celorlalți, fără a determina schimbarea valorilor de activare ale unităților.

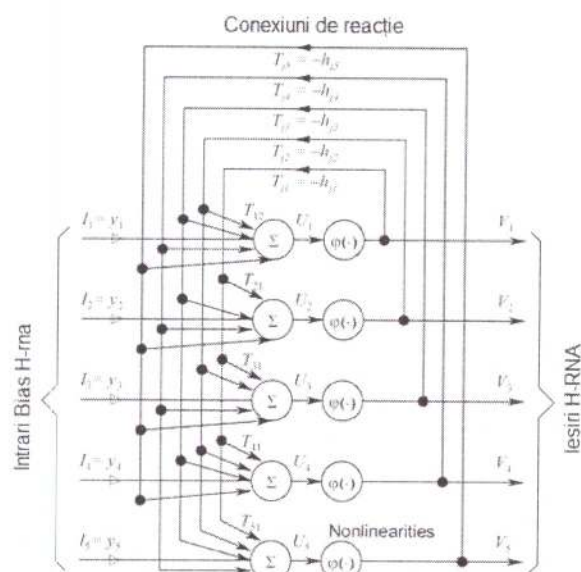


Figura 1. 10 RNA Hopfield cu cinci noduri
Sursa: [27]

Stabilitatea reprezintă proprietatea unei rețele neuronale recurente de a se stabiliza (de a atinge o stare stabilă) indiferent de starea inițială. Au fost definite mai multe **teoreme de stabilitate**: Cohen-Grossberg, Kosko, Abam. Cohen și Grossberg au demonstrat ca rețelele neuronale recurente sunt stabile dacă și numai dacă sunt îndeplinite (3.34) și (3.35).

Acest tip de RNA poate fi asociată cu o **memorie asociativă** sau **memorie adresabilă prin conținut**, a cărei funcție principală este regăsirea tiparelor stocate în memorie, ca răspuns la prezentarea unui tipar incomplet sau contaminat cu zgomot [3].

Algoritmul RNA Hopfield este următorul [28]:

1. Atribuirea de ponderi conexiunilor (legăturilor)

$$w_{ij} = \begin{cases} \sum_{s=0}^{M-1} w_i^s \cdot w_j^s & i \neq j \\ 0 & i = j \quad 0 \leq i, j \leq M-1 \end{cases} \quad (3.41)$$

unde w_{ij} este ponderea conexiunii dintre nodul i și j , iar x_i^s este elementul i aparținând modelului de clasă s , și poate avea valorile: $+1$ sau -1 . Există M modele de la 0 la $M-1$. Pragurile unităților sunt zero.

2. Inițializarea modelului necunoscut

$$\mu_i(0) = x_i \quad 0 \leq i \leq N-1 \quad (3.42)$$

unde $\mu_i(t)$ este ieșirea nodului i la momentul t .

3. Iterarea până la convergență

$$\mu_i(t+1) = f_h \cdot \left(\sum_{j=0}^{N-1} w_{ij} \cdot \mu_j(t) \right) \quad 0 \leq i \leq N-1 \quad (3.43)$$

unde f_h este o funcție puternic limitată și neliniară, similară cu funcția treaptă. Se repetă iterația până când valorile la ieșirile din noduri rămân neschimbate.

5.3 RNA parțial recurente de tip Elman

Rețeaua neuronală Elman (Figura 1.11) este una din cele mai simple, putând fi antrenată cu ajutorul algoritmului standard cu propagarea înapoi a erorii [29].

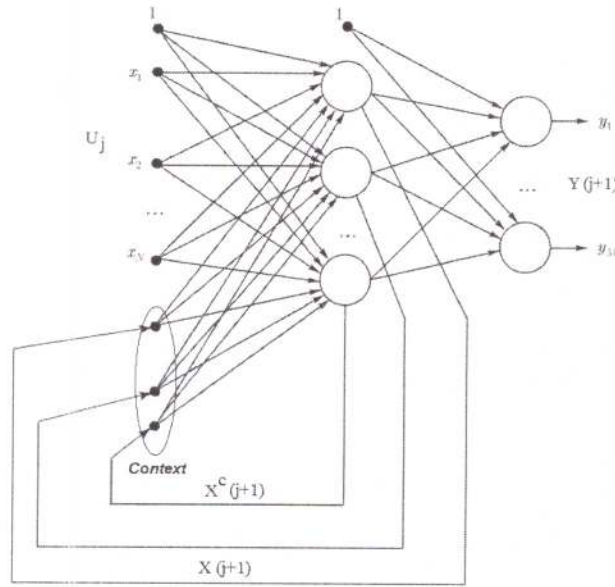


Figura 1. 11 Rețea neuronală de tip Elman
Sursa: [30]

O RNA Elman este constituită din două sau mai multe nivele de unități funcționale în care pe lângă conexiunile specifice rețelelor cu propagare însinte există și conexiuni inverse de la nivelele de unități ascunse către nivelul de intrare [31]. Pentru a ține minte ieșirile unităților ascunse Elman introduce *unități de context* care reprezintă un fel de memorie de scurtă durată (short-term memory) [32]. Unitățile contextuale și conexiunile inverse (starea de la momentul anterior a unităților ascunse ($s(t-1)$) influențează starea curentă ($s(t)$)) permit rețelei să detecteze "structura" unor serii temporale. Unitățile ascunse sunt modelate de funcții activare de tip sigmoidal (funcția implicită este tanh) iar unitățile de ieșire au funcții de activare liniare (purelin).

La o RNA Elman neuroni context sunt alimentați de neuroni de ieșire, nu de neuronii ascunși [30] iar numărul de neuroni context și ascunși trebuie să fie același [19]. Principalul avantaj al RNA Elman este faptul că numărul de neuroni de context nu se definește prin numărul de ieșiri ceea ce face rețeaua mai flexibilă [30].

Prin adăugarea la nivelul neuronilor stratului context, a unei conexiuni de autoreacție, ponderată prin intermediul unei valori fixe, subunitare α se obține RNA Elman extinsă (Figura 1.12) [29], descrisă de ecuația:

$$\frac{\partial x_m(j)}{\partial w_m^{xc}} = X^{cT}(j) = X^T(j-1) + \alpha \cdot \frac{\partial x_m(j-1)}{\partial w_m^{xc}} \quad (3.50)$$

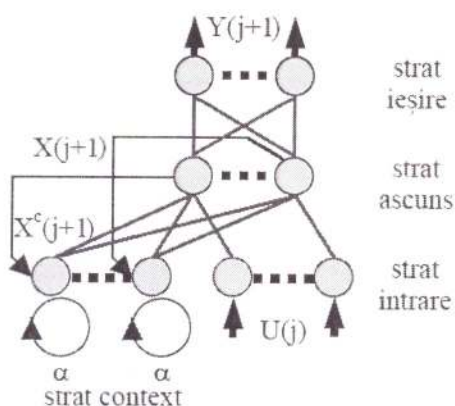


Figura 1. 12 RNA Elman extinsă
Sursa: [29]

Se observă că relația (3.50) are aspectul unui răspuns infinit la impuls (IIR), ceea ce implică memorarea unei cantități semnificativ crescute de informație în RNA Elman extinsă, comparativ cu RNA Elman clasică. RNA Elman sunt aproximatori universali, putând fi teoretic antrenate să aproximeze cu o eroare ϵ orice funcție continuă, liniară sau neliniară, mono sau multivariabilă [33].

Aspectele care merită evidențiate la o RNA Elman sunt [34]: unele probleme își schimbă natura atunci când sunt exprimate ca evenimente temporale; semnalul de eroare, variabil în funcție de timp, poate fi folosit ca un indiciu pentru structura temporală; creșterea dependențelor secvențiale dintr-o sarcină nu conduce la performanțe mai slabe;

6. Regresia liniară simplă și corelația

În regresia liniară simplă, estimăm scoruri pentru o variabilă în funcție de o a doua variabilă [35], între cele două variabile existând o corelație foarte puternică [36]. Variabila pe care o prezicem se numește variabila criterială și face referire la y . Variabila pe care ne bazăm predicțiile se numește variabilă predictor și face referire la x . Când există doar o variabilă predictivă, metoda de predicție se numește regresie simplă. În regresia liniară simplă predicțiile lui y când sunt reprezentate în funcție de x formează o linie dreaptă. O linie de regresie reprezintă o linie care se potrivește cel mai bine datelor (în ceea ce privește distanța maximă cea mai mică de la linie la puncte) prin metoda celor mai mici pătrate.

Termenul de regresie a fost introdus de matematicianul Galton [37]. Metoda constă în determinarea unei funcții liniare:

$$\mu_y = E(Y) = \beta_0 + \beta_1 \cdot x \quad (3.51)$$

care să aproximeze media răspunsurilor μ_y (variabila dependentă) prin valorile x (variabila independentă sau predictor), unde x și μ_y sunt variabile de tip continuu.

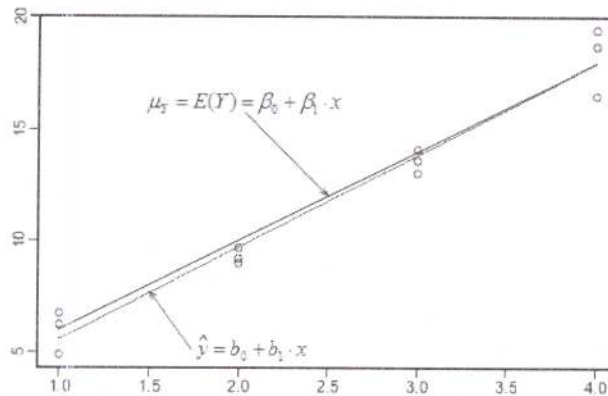


Figura 1.13 Comparație între linia de regresie utilizând metoda celor mai mici pătrate și linia de regresie a populației
Sursa: [38]

Prin extinderea metodei de calcul pentru parametrii dreptei de regresie pentru i respondenți și ținând seama de erorile care apar, relația 3.51 devine:

$$y_i = E(Y_i) + \varepsilon_i = \beta_0 + \beta_1 \cdot x_i + \varepsilon_i \quad (3.52)$$

unde: β_0 – interceptul y , β_1 – panta liniei, ε_i – eroare;

Coeficientul de corelație r este direct legat de coeficientul de determinare r^2 prin relația:

$$r = \pm\sqrt{r^2} \quad (3.54)$$

Coeficientul de determinare fiind la pătrat, r^2 , este întotdeauna un număr pozitiv și variază între 0 și 1. Cu cât valoarea coeficientului de determinare este mai apropiată de valoarea maximă 1, cu atât variația variabilei de răspuns pot fi explicate prin variabilele explicative, diferența putând fi atribuită unor variabile necunoscute sau variabile inerente [37], [38], [39].

Legătura între coeficientul de corelație r și panta eșantionului b_1 este dată de relația:

$$r = \frac{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \cdot b_1 \quad (3.55)$$

Teoria analizei regresiei indică faptul că cel mai sigur loc pentru obținerea interpolării este mijlocul intervalului valorilor lui x . Este mai puțin sigură la marginile intervalului deoarece prin extrapolare rezultatele devin din ce în ce mai nesigure pe măsură ce continuă să se depărteze de domeniul valorilor x .

Modelul **regresie liniară Theil-Sen** propune calculul medianeii pantelor tuturor liniilor prin perechi de puncte de eșantioane bidimensionale. În comparație cu estimatorul celor mai mici pătrate, estimatorul Theil-Sen (TS) este robust împotriva valorilor extreme. Are un punct de descompunere de aproximativ 29,3% în cazul unei regresii liniare simple, ceea ce înseamnă că poate tolera date arbitrare de până la 29,3% în cazul bidimensional [40], [42].

Estimatorul Theil-Sen, $\hat{\theta}_1$, este dat de relația [41], [42]:

$$\hat{\theta}_1 = \text{med}_{1 \leq i < j \leq n} \frac{y_j - y_i}{x_j - x_i} \quad (3.56)$$

Prin repetarea estimării mediane se poate ajunge la un punct de descompunere de aproximativ 50%, metodă aplicată de Siegel în 1982.

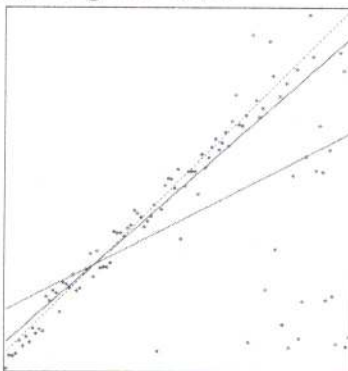


Figura 1. 14 Estimatorul Theil-Sen
Sursa [43]

În Figura 1.14 este prezentat estimatorul TS al unui set de puncte de eșantionare cu valori exagerate (linia neagră), comparativ cu linia estimatorului celor mai mici pătrate (non-robustă) pentru același set (linia albastru). Linia verde punctată reprezintă adevărul de la care au fost generate probele.

Concluzia 3: Dacă setul de date este distribuit în mod normal și sunt de înaltă calitate, estimatorul regresie TS este competitiv cu regresia celor mai mici pătrate, deci generează rezultate superioare evidențiate prin stabilitatea estimărilor și evaluări de bună calitate [44]. Problema ineficienței se elimină folosind metode de eșantionare aleatorie sau metode deterministe [45].

BIBLIOGRAFIE

- [1] <https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html>
- [2] D. Michie, D.J. Spiegelhalter, C.C. Taylor, *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, New York, 1994, ISBN 013106360X, 9780131063600, <https://www1.maths.leeds.ac.uk/~charles/statlog/whole.pdf>
- [3] Swati Aggarwal, Ranjit Biswas, A.Q.Ansari, *Neutrosophic Modeling and Control*, International Conference on Computer and Communication Technology, 2010, DOI: 10.1109/ICCCT.2010.5640435, https://www.researchgate.net/publication/231521008_Neutrosophic_Modeling_and_Control
- [4] Ioan Dzițac, *Inteligență Artificială*, Editura Universității "Aurel Vlaicu", Arad, 2008, ISBN 978-973-752-292-4 004.42, http://www.uav.ro/files/exacte/cursuri/Inteligenta_artificiala_Dzitac.pdf
- [5] Bălan C. George, http://www.mrm.ugal.ro/balan_site/e-books/miassm-pdf/cap.1.pdf
- [6] Ramin Shamshiri, Wan Ishak Wan Ismail, *Design and Simulation of Control Systems for a Field Survey Mobile Robot Platform*, Research Journal of Applied Sciences, Engineering and Technology 6(13): 2307-2315, 2013, ISSN: 2040-7459; e-ISSN: 2040-7467, https://www.researchgate.net/publication/259932485_Design_and_Simulation_of_Control_Systems_for_a_Field_Survey_Mobile_Robot_Platform
- [7] Thiang St, Handry Khoswanto, Rendy Pangaldus, *Artificial Neural Network with Steepest Descent Backpropagation Training Algorithm for Modeling Inverse Kinematics of Manipulator*, World Academy of Science, Engineering and Technology Vol. 60, 2009, https://www.researchgate.net/publication/43649867_Artificial_Neural_Network_with_Steepest_Descent_Backpropagation_Training_Algorithm_for_Modeling_Inverse_Kinematics_of_Manipulator
- [8] Dosoftei Constantin-Cătălin, *Utilizarea inteligenței computaționale în conducerea proceselor*, Teză de doctorat, Universitatea tehnică Gh.Asachi, Iași, 2009, <http://www.ace.tuiasi.ro/users/103/2009-Dosoftei%20Catalin%20PhD%202009.pdf>
- [9] Uve Windhort, Hakan Johansson, *Modern techniques in neuroscience research*, Springer-Verlag Berlin Heidelberg 1999, ISBN 978-3-642-63643-1, ISBN 978-3-642-58552-4 (eBook), DOI 10.1007/978-3-642-58552-4, <https://books.google.ro/books?id=xfXsCAAAQBAJ&pg=PA592&lpg=PA592&dq=Static+neuron+is+the+simplest+model+of+artificial+neuron&source=bl&ots=luOYwdAvl->

[&sig=jq20kt30bmtV-](#)

[UUKBPz4gyg8hqw&hl=ro&sa=X&ved=0ahUKEwjOwsSc5MrVAhUCshQKHf5LAgIQ6AEIWDAH#v=onepage&q&f=false](#)

[10] Daniela Danciu, *Probleme calitative în dinamica rețelelor neuronale*, Teză de doctorat, Universitatea din Craiova, 2003, <http://www.automation.ucv.ro/Romana/membri/Daniela%20Danciu/DDRezumato.pdf>

[11] Ciocoiu Iulian-Aurelian, *Curs, Cap.2 Caracteristici fundamentale ale rețelelor neurale artificiale*, http://scs.etc.tuiasi.ro/iciocoiu/courses/DSP/course5/capitol2_NN.pdf

[12] Klaus Debes, Alexander Koenig, Horst-Michael Gross, *Transfer Functions in Artificial Neural Networks. A Simulation-Based Tutorial*. Supplementary Material for urn:nbn:de:0009-3-1515, <http://www.brains-minds-media.org/archive/151/supplement/bmm-debes-suppl-050704.pdf>

[13] Laura Dioșan, *Inteligență artificială. Curs 9*, Universitatea Babeș-Bolyai, 2013, http://www.cs.ubbcluj.ro/~lauras/test/docs/school/IA/lectures2013/lectures/09_10_ML_ANN.pdf

[14] Masahiko Morita, Shuji Yoshizawa, Kaoru Nakano, *Memory of Correlated Patterns by Associative Neural Networks with Improved Dynamics*, The International Neural Network Society (INNS), the IEEE Neural Network Council Cooperating Societies ISBN: 978-0-7923-0831-7 (Print) 978-94-009-0643-3 (Online), https://link.springer.com/chapter/10.1007/978-94-009-0643-3_115

[15] <http://www.tc.etc.upt.ro/teaching/or/or.pdf> sau http://shannon.etc.upt.ro/laboratoare/or/or_laborator.pdf

[16] László Bakó, *Sisteme adaptive cu rețele neuronale artificiale neuromorfe. Realizări cu dispozitive hardware reconfigurabile*, Teză de doctorat, Universitatea Transilvania din Brașov, 2009, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.705.8811&rep=rep1&type=pdf>

[17] Prostean Gabriela Ioana, *Sisteme inteligente în electrotehnică*, http://www.mpt.upt.ro/doc/curs/gp/Sisteme_inteligente_in_electrotehnica/Inteligenta_artificiala_si_Retele_neuronale_cap1.pdf

[18] Madan M. Gupta, Liang Jin, Noriyasu Homma, *Static and dynamic neural networks. From fundamentals to advanced theory*, IEEE Press, John Wiley & Sons, Inc., Hoboken, New Jersey, 2003, ISBN 0-471-21948-7, <https://doc.lagout.org/science/Artificial%20Intelligence/Neural%20networks/Static%20and%20Dynamic%20Neural%20Networks%20From%20Fundamentals%20to%20Advanced%20Theory%20-%20Madan%20M.%20Gupta%20Liang%20Jin%20Noriyasu%20Homma.pdf>

[19] Krzysztof Patan, *Artificial Neural networks for the modeling and fault diagnosis of technical processes*, Springer-Verlag Berlin Heidelberg, 2008, ISBN 978-3-540-79871-2, e-ISBN 978-3-540-79872-9, DOI 10.1007/978-3-540-79872-9, http://lab.fs.uni-lj.si/lasin/wp/IMIT_files/neural/doc/Patan2008.pdf

- [20] Knapp, *Fuzzy Sets and Pattern Recognition*, Copyright © 1996-2004, <http://www.cs.princeton.edu/courses/archive/fall07/cos436/HIDDEN/Knapp/fuzzy004.htm>
- [21] Kohonen, *Self-Organizing Maps*, Second Edition, Springer Series in Information Sciences, Vol. 30, 2001, ISSN 0720-678X, ISBN-13: 978-3-540-62017-4, e-ISBN-13: 978-3-642-97966-8, DOI: 10.1007/978-3-642-97966-8, <https://www.scribd.com/document/328251982/Professor-Teuvo-Kohonen-Auth-Self-Organizing-Maps>
- [22] Kunihiko Fukushima, *Artificial vision by multi-layered neural networks: Neocognitron and its advances*, Neural Networks, Vol.37, pp 103-119, 2013, www.elsevier.com/locate/neunet, https://pdfs.semanticscholar.org/a98f/62e544068a6ce03b051387554b6bbd23646a.pdf?_ga=2.255846536.1462574674.1502625292-1173714993.1502625292
- [23] Enăchescu Călin, http://upm.ro/intranet/ecalin/cd_educational/cd/neuron/cap2/cap2.htm#II.2
- [24] Raúl Rojas, Neural Networks, *A Systematic Introduction*, Springer-Verlag, Berlin, 1996, <https://page.mi.fu-berlin.de/rojas/neural/neuron.pdf>
- [25] Constanța Bodea, *Optimizarea cu ajutorul rețelelor neuronale recurente de tip Hopfield*, Revista Informatica Economica, nr. 1 (21)/2002, <http://revistaie.ase.ro/content/21/bodea.pdf>
- [26] Lisboa, P. G., S. J. Perantonis, *Complete Solution of the Local Minima in the XOR Problem*, Network – Computation in Neural Systems, Vol. 2, No. 1, pp. 119–124,1991, https://www.researchgate.net/publication/232050328_Complete_solution_of_the_local_minima_in_the_XOR_problem
- [27] Ebrahim Soujeri, Huseyin Bilgekul, *Multiuser Detection of Synchronous MC-CDMA in Multipath Fading Channels Using Hopfield Neural Networks*, Kluwer Academic Publishers, Neural Processing Letters 18, pp. 49–63, 2003, <https://link.springer.com/content/pdf/10.1023/A:1026215305227.pdf>
- [28] Russell Beale, Tom Jackson, *Neural Computing: An Introduction*, Pub. Taylor& Francis Group, New York, 1990, ISBN: 978-0-85274-262-4, eBook ISBN: 978-1-4200-5043-1, <https://doi.org/10.1201/9781420050431>, <https://bayanbox.ir/view/7901640340179926235/Neural-Computing-An-Introduction.pdf>
- [29] *Rețele neuronale. Modelarea și simularea sistemelor dinamice*, www.infoap.utcluj.ro/SO/cap2-f.doc, www.infoap.utcluj.ro/SO/cap4-f.doc
- [30] Vladimir Perervenko, *Third Generation Neural Networks: Deep Networks*, 2015, https://www.mql5.com/en/articles/1103#1_2_3
- [31] Zaharie Daniela, Implementarea memoriilor asociative cu rețele de tip Hopfield.
- [32] Adina Magda Florea, *Sisteme de programe pentru timp real*, Universitatea “Politehnică” din București, 2005, http://turing.cs.pub.ro/sptr_05/SPTR_Lect_6.ppt
- [33] Ling Li, Zhidong Deng, Bo Zhang, *A Fuzzy Elman Neural Network*, <http://www.work.caltech.edu/~ling/pub/97fenn.pdf>

- [34] Jeffrey L. Elman, *Finding Structure in Time*, Cognitive Science Vol. 14, nr.2, pp. 179-211, 1990, ISSN 1551-6709, doi:10.1207/s15516709cog1402_1, <http://psych.colorado.edu/~kimlab/Elman1990.pdf>
- [35] David M. Lane et all, *Introduction to Statistics*, Online Edition, http://onlinestatbook.com/Online_Statistics_Education.pdf
- [36] Deborah J. Rumsey, *Statistics For Dummies*, 2nd Edition, ISBN: 978-1-119-29352-1, <http://www.dummies.com/education/math/statistics/how-to-calculate-a-regression-line/>
- [37] http://www.umfiasi.ro/Rezidenti/suporturidecurs/Facultatea%20de%20Farmacie/Specialitatea%20Farmacie%20clinica/matematica%20si%20biostatistica/biostatistica_2.pdf
- [38] <https://onlinecourses.science.psu.edu/stat501/node/253>
- [39] https://en.wikipedia.org/wiki/Coefficient_of_determination
- [40] http://scikit-learn.org/stable/auto_examples/linear_model/plot_theilsen.html
- [41] Hanxiang Peng, *The Theil-Sen Estimators In Linear Regression*, 2008, <http://www.math.iupui.edu/~hpeng/Talks/TSETalk.pdf>
- [42] Peter J. Rousseeuw, Annick M. Leroy, *Robust regression and outlier detection*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2003, ISBN 0-47-48855-0, https://books.google.ro/books?id=IK9gHXwYnqgC&pg=PA67&redir_esc=y#v=onepage&q&f=false
- [43] https://en.wikipedia.org/wiki/Theil%E2%80%93Sen_estimator#CITEREFRousseeuwLeroy2003
- [45] Andrew Greatorex, *Linear Regression Sucks*, <https://medium.com/towards-data-science/linear-regression-sucks-27a5215e50c0>
- [44] James A. Ohlson, Seil Kim, *Linear Valuation without OLS: The Theil-Sen Estimation Approach*, 2014, <https://research.mbs.ac.uk/accounting-finance/Portals/0/docs/Linear%20Valuation%20without%20OLS.pdf>
- [45] <http://visualgenedeveloper.net/index.html>
- [46] Alexandru-Daniel TUFAN, *Automatizarea integrată a sistemelor de acționare bazate pe logică programabilă și configurarea parametrilor specifici*, Teză de doctorat, Universitatea POLITEHNICA din București, Facultatea Ingineria și Managementul Sistemelor Tehnologice, aprilie 2018
- [47] Daniel-Petru GHENCEA, *Modelare-simulare și predicția datelor experimentale specifice sistemelor de fabricație utilizând tehnici hibride bazate pe inteligență artificială*, Teză de doctorat, Universitatea POLITEHNICA din București, Facultatea de Ingineria și Managementul Sistemelor Tehnologice, aprilie 2018.