

**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

**METODOLOGIA CERCETĂRII PENTRU PREDICȚIA DATELOR
INCOMPLETE DIN SISTEMELE AUTOMATIZATE INDUSTRIALE
UTILIZÂND REȚELE NEURONALE ARTIFICIALE**

Autor:

S.L. dr. ing. Daniela **TUĐOSE** – Universitatea Politehnica București – Departamentul Rezistența Materialelor

Profesor coordonator:

Prof. univ. dr. ing. Miron **ZAPCIU** – Universitatea Politehnica București – IMST / Academia Oamenilor de Știință din România

CUPRINS

CAPITOLUL 1 REȚELE NEURONALE ARTIFICIALE	3
1.1 Introducere	3
1.2 Conceptul de Rețele Neuronale Artificiale	3
1.2.1 Neuronul artificial static.....	5
1.2.2 Funcții de activare a neuronului.....	7
1.3 Modele de neuroni artificiali.....	8
1.4 Tipuri și algoritmi de instruire	11
1.5 Rețele neuronale recurente	12
1.5.1 RNA cu propagare înapoi	12
1.5.2 RNA total recurente de tip Hopfield.....	13
1.5.3 RNA parțial recurente de tip Elman.....	15
1.6 Regresia liniară simplă și corelația.....	16
CAPITOLUL 2 LOGICA FUZZY	19
2.1 Introducere	19
2.2 Conceptul de logică fuzzy.....	19
2.3 Caracterizarea submulțimilor fuzzy	20
2.4 Funcții de apartenență și numere fuzzy.....	21
2.5 Componentele structurii informaționale de bază fuzzy logic.....	22
2.6 Metode de inferență.....	24
2.6.1 Metoda Mamdani	24
2.6.2 Metoda Takagi-Sugeno-Kang	24
2.7 Metode de defuzzyficare	25
2.7.1 Defuzzyficare de tip Mamdani	25
2.7.2 Defuzzyficare de tip Takagi-Sugeno-Kang	26
CAPITOLUL 3 DETERMINAREA VARIANTEI OPTIME A RNA-BP PENTRU OBȚINEREA UNEI PRECIZII RIDICATE DE PREDICȚIE A PARAMETRILOR ÎN CAZUL OȚELULUI UTILIZAT PENTRU ȘINA DE CALE FERATĂ	27
3.1 Metodologia cercetării.....	27
3.2 Predicția parametrilor SLD pe baza SRD	30

3.3 Predicția parametrilor SLD pe baza SRD	32
CONCLUZII FINALE	35
BIBLIOGRAFIE	36

CAPITOLUL 1 REȚELE NEURONALE ARTIFICIALE

1.1 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, problema 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.

1.2 Conceptul de Rețelele Neuronale Artificiale

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

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

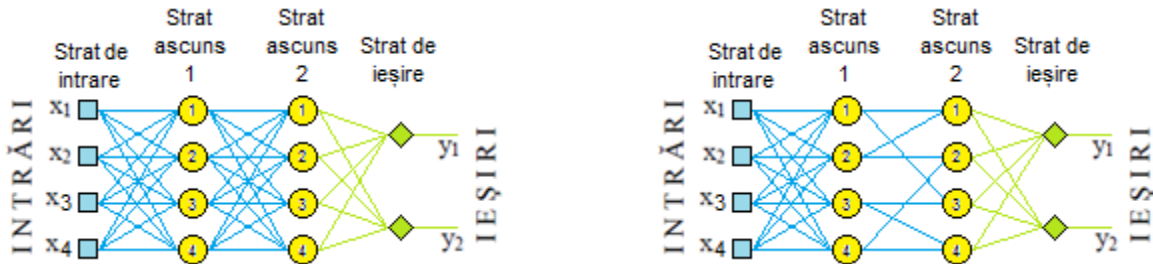
- ✓ *învățarea și adaptarea* sunt cele mai importante proprietăți;
- ✓ *distribuția informației în structura*: informația din rețea este stocată în mod distribuit făcând ca efectul unei anumite intrări asupra ieșirii să depindă de restul ponderilor din rețea.
- ✓ *capacitatea de generalizare* în cazul 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 etapa 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;

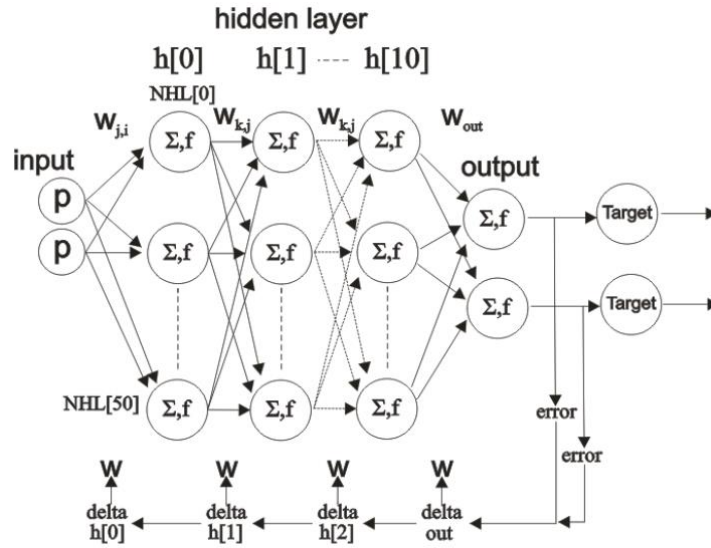
Principalele tipuri de RNA sunt: tip perceptron, bazate pe funcții radiale, recurente și cu auto-organizare.

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.

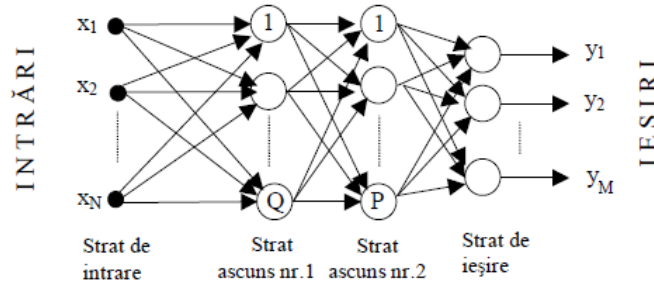


Figura 1. 2 Perceptron cu două straturi
Sursa: [6]

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 determină experimental.

1.2.1 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:

$$y = a \cdot u \quad a = ct. \quad (1.1)$$

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

$$Y = A \cdot U \quad (1.2)$$

unde:

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

Pentru sistemul static din Figura 1.3 avem relația:

$$y = u_1 \cdot w_1 + u_2 \cdot w_2 + u_3 \cdot w_3 \quad (1.3)$$

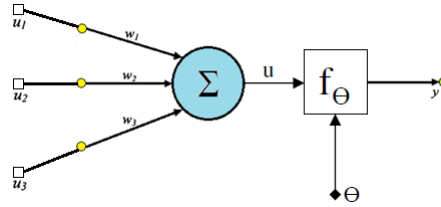


Figura 1. 3 Sistem static liniar
Sursa: [9]

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.

$$y = f(u - \theta) \quad (1.4)$$

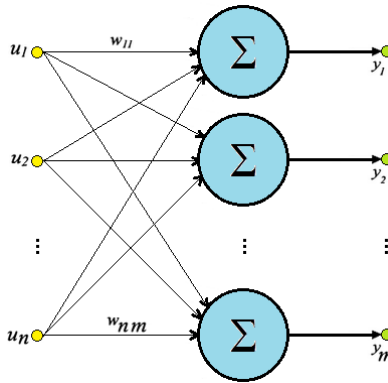


Figura 1. 4 Rețea neuronală artificială liniar asociativă
Sursa: [9]

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:

$$y_j = \sum_i w_{ij} u_i \quad (1.5)$$

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

$$Y = W^T \cdot U \quad (1.6)$$

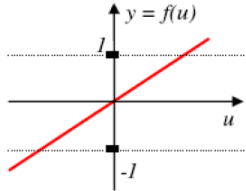
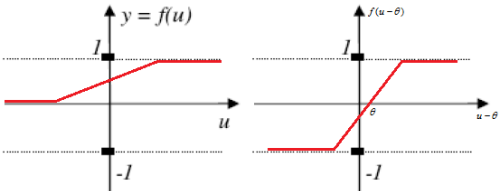
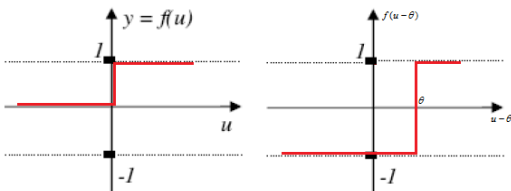
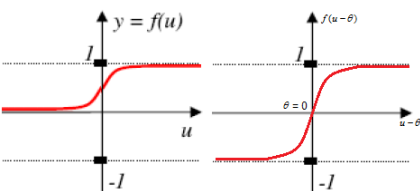
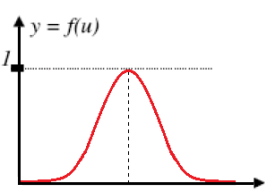
sau sub formă matricială:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \quad W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix} \quad W^T = \begin{bmatrix} w_{11} & w_{21} & \cdots & w_{n1} \\ w_{12} & w_{22} & \cdots & w_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1m} & w_{2m} & \cdots & w_{nm} \end{bmatrix} \quad (1.7)$$

1.2.2 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].

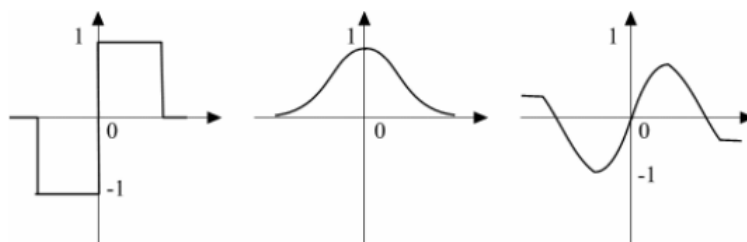


Figura 1.5 Funcții de activare nonmonotone
Sursa [11], [14], [15]

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 pulsanti** 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.

1.3 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 u_i se aplică o funcție de activare (polarizare) și trimite rezultatul printr-o neliniaritate la ieșire.

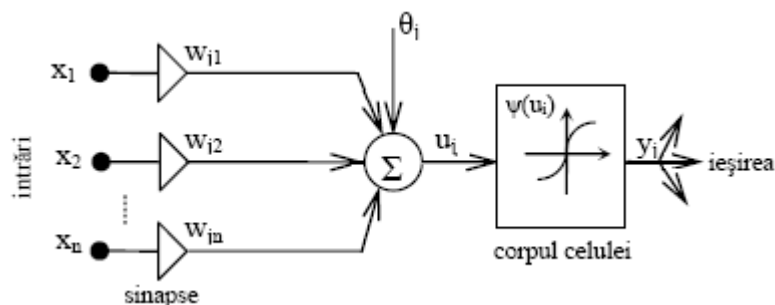


Figura 1.6 Modelul structural al neuronului McCulloch – Pitts
Sursa: [6]

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.

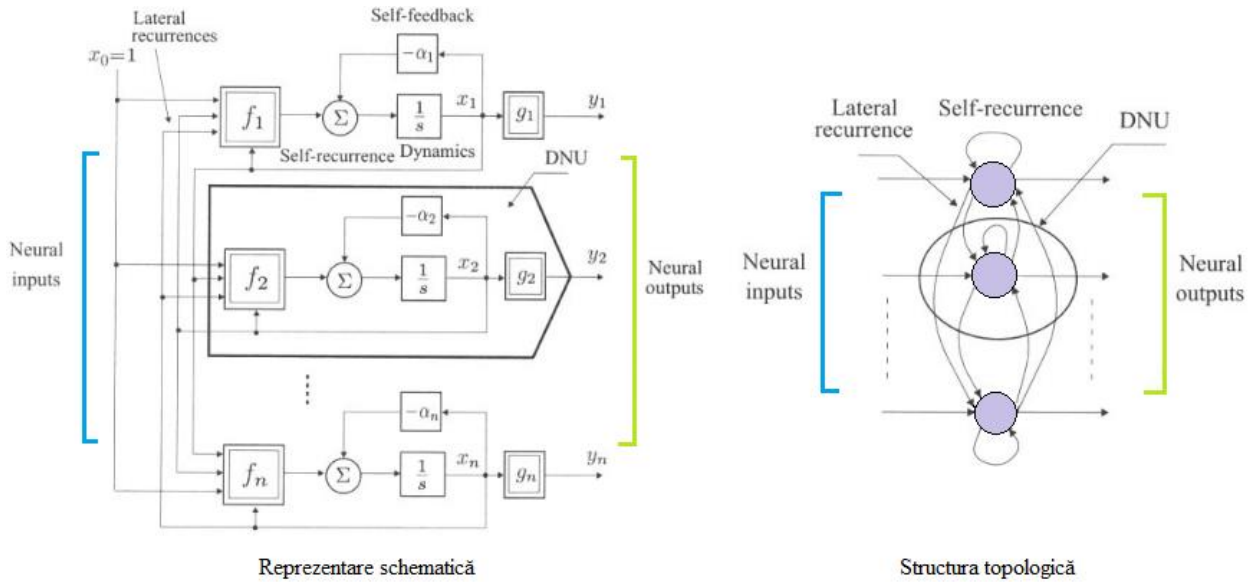


Figura 1. 7 Conectarea unității dinamice neuronale (DNU) în rețeaua neuronală artificială
Sursa: [18].

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:

$$y_j = \Psi \left(\frac{1 + \sum_{i=1}^n a_{ji} \cdot x_i}{1 + \sum_{i=1}^m b_{ji} \cdot v_i} - 1 \right) \quad \text{unde} \quad \Psi(u_j) = \begin{cases} u_j, u_j \geq 0 \\ 0, u_j < 0 \end{cases} \quad (1.8)$$

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

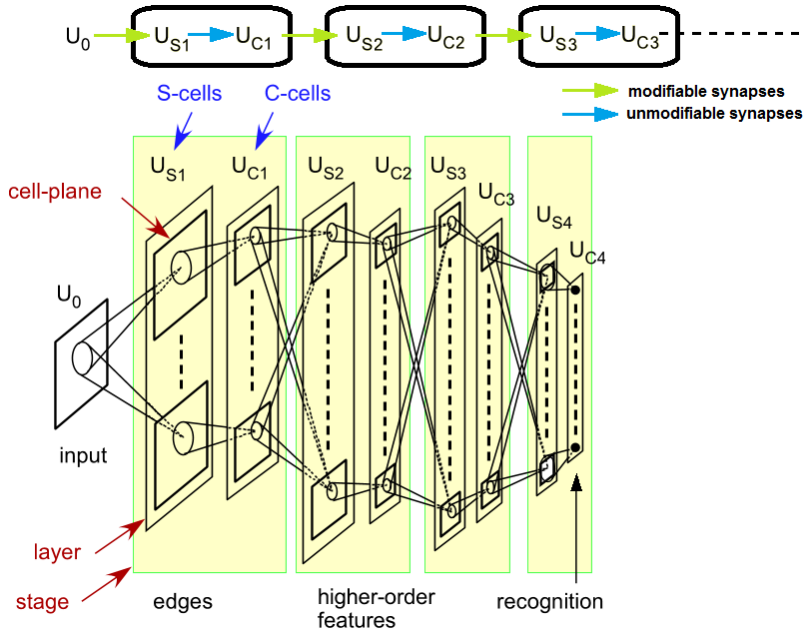


Figura 1.8 Structura de rețea ierarhică a Neocognitronului
Sursa: [20], [21], [22]

K. Fukushima a elaborat un sistem de calcul neuronal pentru recunoasterea 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).

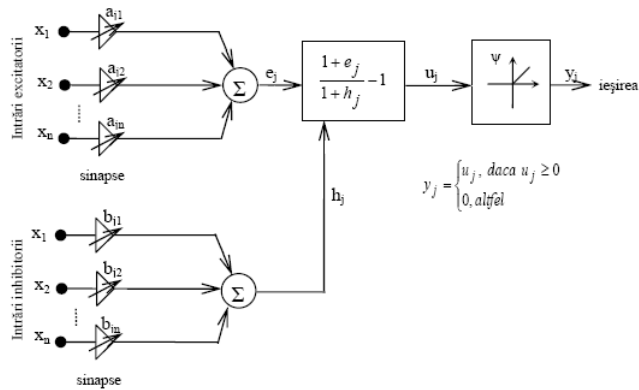


Figura 1.9 Modelul structural al neuronului Fukushima
Sursa: [3]

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.

1.4 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 imprecisi. 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:

$$E = -\frac{1}{2} \cdot \sum_i \sum_j w_{ji} \cdot s_j \cdot s_i \quad (1.9)$$

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.

1.5 Rețele neuronale recurente

1.5.1 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 $\{(x_1, t_1), \dots, (x_p, t_p)\}$ 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:

$$E = \frac{1}{2} \cdot \sum_{i=1}^p \|o_i - t_i\|^2 \quad (1.10)$$

După minimizarea acestei funcții pentru setul de instruire, noi modele de intrare necunoscute sunt prezentate RNA și ne așteptăm ca aceasta să interpoleze 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:

$$\nabla_{w_i} = \left(\frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_i} \right) \quad (1.11)$$

Fiecare pondere este actualizată cu ajutorul incrementului:

$$\Delta w_i = -\gamma \cdot \frac{\partial E}{\partial w_i} \quad i = 1, \dots, l \quad (1.12)$$

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.

1.5.2 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:

$$w_{ij} = w_{ji} \quad (1.13)$$

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

Lipsa autoasocierii se exprimă prin:

$$w_{ij} = w_{ji} \quad (1.14)$$

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.

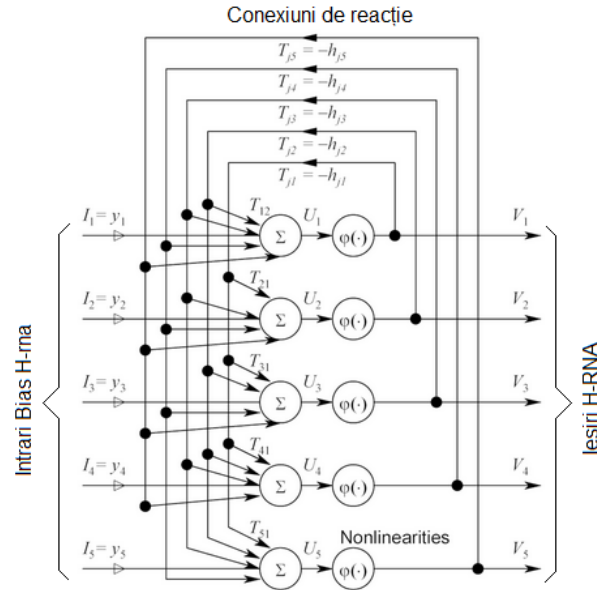


Figura 1. 10 RNA Hopfield cu cinci noduri

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 Hopfielf este următorul [28]:

1. Atribuirea de ponderi conexiunilor (legăturilor)

$$w_{ij} = \begin{cases} \sum_{s=0}^{M-1} w_i^s \cdot w_j^s & i \neq j \\ 0 & i = j \quad 0 \leq i \quad j \leq M - 1 \end{cases} \quad (1.15)$$

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

$$\mu_i(0) = x_i \quad 0 \leq i \leq N - 1 \quad (1.16)$$

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

3. Iterarea până la convergență

$$\mu_i(t + 1) = f_h \cdot \left(\sum_{j=0}^{N-1} w_{ij} \cdot \mu_j(t) \right) \quad 0 \leq i \leq N - 1 \quad (1.17)$$

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.

1.5.3 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].

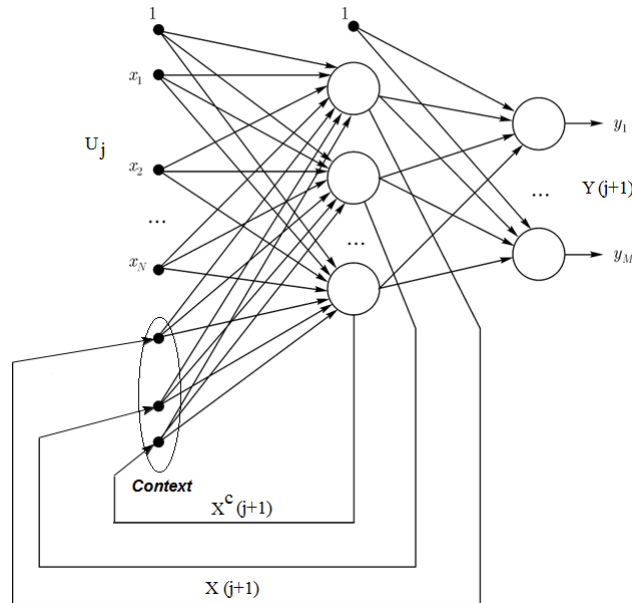


Figura 1. 11 Rețeaua neuronală de tip Elman
Sursa: [30]

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:

$$\frac{\partial x_m(j)}{\partial w_m^{xc}} = X^{ct}(j) = X^T(j-1) + \alpha \cdot \frac{\partial x_m(j-1)}{\partial w_m^{xc}} \quad (1.18)$$

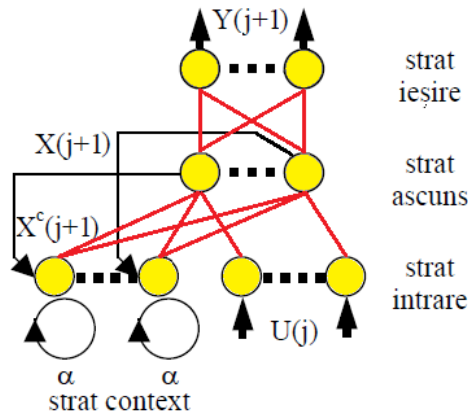


Figura 1. 12 RNA Elman extinsă
Sursa: [29]

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;

1.6 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:

$$\mu_y = E(Y) = \beta_0 + \beta_1 \cdot x \quad (1.19)$$

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.

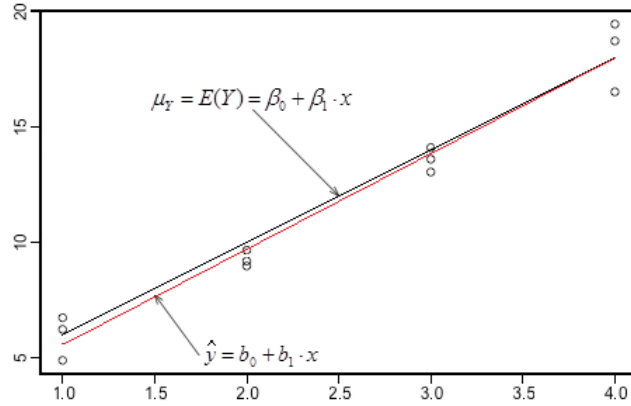


Figura 1. 13 Comparație între linia de regresie utilizând metoda celor mai mici pătrate și linia de regresie a populației
Sursa: [38]

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

$$y_i = E(Y_i) + \varepsilon_i = \beta_0 + \beta_1 \cdot x_i + \varepsilon_i \quad (1.20)$$

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

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

$$r = \pm\sqrt{r^2} \quad (1.21)$$

Coefficientul 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:

$$r = \frac{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \cdot b_1 \quad (1.22)$$

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 medianei 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]:

$$\hat{\theta}_1 = \text{med}_{1 \leq i < j \leq n} \frac{y_j - y_i}{x_j - x_i} \quad (1.23)$$

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

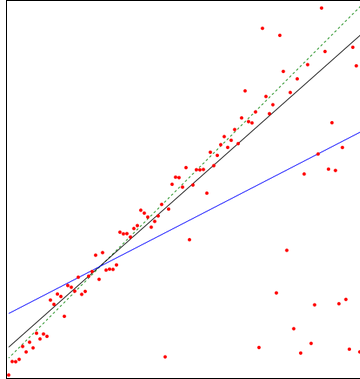


Figura 1. 14 Estimatorul Theil-Sen
Sursa [43]

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.

Concluzie 1.3: 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].

CAPITOLUL 2 LOGICA FUZZY

2.1 Introducere

Logica bazată pe analiza a trei valori (*three-valued logic*) (adevărat, fals și nedeterminat) a intrat în atenția cercetătorilor Jan Lukasiewicz și Alfred Tarski în anii 1920-1930. Cercetările lor au cunoscut o revigorare în anii 1950-1960 datorită lui **Lotfali Askar Zadeh** [48] (profesor la Universitatea din California, Berkeley) care a introdus termenul de logică fuzzy pentru a explica teoria logicii cu un număr infinit de valori (*infinitely-many valued logics*) [49].

Conceptul de set fuzzy are în vedere o clasa de obiecte cu un continuum de grad de membru astfel încât funcția de apartenență să atribuie fiecărui obiect un grad de apartenență având rangul cuprins între 0 și 1. Incluziunea, reuniunea, intersecția, complementul și convexitatea sunt extinse la întregul set iar varietatea proprietăților acestor noțiuni sunt stabilite în contextul setului fuzzy [48].

2.2 Conceptul de logică fuzzy

Realizarea unei metode flexibile de rezolvare a problemelor de incertitudine s-a realizat prin dezvoltarea sistemelor fuzzy, care au la bază logica fuzzy, fiind un caz particular al sistemelor expert [50].

Logica fuzzy operând cu elementele $A = \{x/x \in [0,1]\}$ atribuie obiectului un grad de apartenență la mulțime. Robuștețea logicii fuzzy este evidențiată de controlul simultan de date numerice și cunoștințe lexicale (variabile lingvistice) prin interpretarea termenilor cantitativi în termeni calitativi [51].

Variabila lingvistică este o proprietate iar ca structură cuprinde [52]:

1. **Valoarea lingvistică** u este un adverb, adjectiv asociat variabilei lingvistice, care dă numele mulțimii fuzzy asociate;
2. **Domeniul de reprezentare** U este o mulțime clasică, pe care se definesc mulțimile fuzzy. Mulțimea U se mai numește: *domeniu de reprezentare*, *univers de discurs* sau *mulțime referențial*.
3. **Funcția de apartenență** μ_F asociază fiecărui element u gradul de apartenență la mulțimea fuzzy F ;
4. **Gradul de apartenență** μ reprezintă măsura în care un element aparține unei mulțimi fuzzy;

Pentru a înțelege teoria logicii fuzzy și a setului fuzzy este necesară prezentarea elementelor pe care aceasta se bazează [53], [54], [55].

Fie U o mulțime de obiecte notată generic $\{u\}$, care poate fi discretă sau continuă. U se numește domeniu de reprezentare (univers de discurs) iar u reprezintă elementele generice ale lui U .

Definiția 1. *Set fuzzy:* Un set fuzzy F inclus în domeniul de reprezentare U este caracterizat de funcția de apartenență μ_F care ia valori în intervalul $[0, 1]$, adică $\mu_F : U \rightarrow [0,1]$.

Un set fuzzy poate fi interpretat ca o generalizare a conceptului de set comun unde funcția de apartenență poate lua doar două valori $\{0, 1\}$. De asemenea setul fuzzy F inclus în domeniul de reprezentare U , poate fi reprezentat printr-un set de perechi ordonate ale elementului generic u și arată gradul de apartenență al funcției.

$$F = \{(u, u_F(u)) \mid u \in U\} \quad (2.1)$$

Observație: Definiția propune înlocuirea enumerării mulțimii cu definirea mulțimii prin gradul de apartenență la această mulțime [56].

Pentru perechea ordonată $(u, u_F(u))$ se mai utilizează notația $\mu(u)/u$ [57].

Dacă U este *continuă*, setul fuzzy F poate fi scris ca:

$$F = \int_U \mu_F(u) / u \quad (2.2)$$

Dacă U este *discretă*, setul fuzzy poate fi reprezentat ca:

$$F = \sum_{i=1}^n \mu_F(u) / u \quad (2.3)$$

Prin \int_U și $\sum_{i=1}^n$ nu s-a notat *integrala* și *suma*, ci ele reprezintă relația de corespondență dintre valorile fuzzy și valorile exacte (crisp) ale domeniului de reprezentare (univers de discurs) [58].

Definiția 2. *Support, Punct de legătură* (crossover point) și *Fuzzy singleton* (cu un element). *Supportul* unui set fuzzy F este setul clar al mulțimii obiectelor $u \subset U$ astfel încât $\mu_F(u) > 0$. Dacă $u \subset U$ există următoarele cazuri particulare:

a) $\mu_F = 0,5$ - *punct de legătură*;

$\mu_F = 1$ - *fuzzy singleton* (un singur element) – este setul fuzzy pentru care suportul are un singur punct (element);

2.3 Caracterizarea submulțimilor fuzzy

Caracterizarea submulțimilor fuzzy ale lui U se realizează cu următoarele mărimi [59], [60], [61]:

a) **Supportul** unei funcții de apartenență pentru un set fuzzy U este definit ca acea regiune a domeniului de reprezentare care este caracterizată de un membru diferit de zero la setul U . Suportul mulțimii U notat $Supp(U)$ este tăietura strictă de nivel 0 a mulțimii U :

$$Supp(U) = \{u \in U \mid \mu_A(u) > 0\} \quad (2.4)$$

b) Se numește **tăietura de prag α** mulțimea valorilor clare:

$$[\mu]_\alpha = \{u \in U \mid \mu_A(u) \geq \alpha\} \quad (2.5)$$

Dacă inegalitatea este strictă se spune că α - *tăietura* este de tip tare și va fi notată $[\mu]_{+\alpha}$.

c) **Nucleul** unei funcții de apartenență pentru un set fuzzy U este definit ca acea regiune a domeniului de reprezentare care se caracterizează prin aderarea totală la setul U .

Nucleul mulțimii U este notat $Ker(U)$ și cuprinde elementele care satisfac relația:

$$Ker(U) = \{u \in U \mid \mu_A(u) = 1\} \quad (2.6)$$

- d) **Înălțimea** mulțimii U notată $h(U)$ reprezintă cea mai mare valoare luată de funcția sa de apartenență:

$$h(U) = \sup\{\mu_A(u) \mid u \in U\} \quad (2.7)$$

- e) **Frontiera** (Limita) unei funcții de apartenență pentru un set fuzzy U este definită ca acea regiune a domeniului de reprezentare care se caracterizează printr-un membru diferit de zero dar care nu aderă în totalitate la setul U . **Frontiera** mulțimii U notată $Fr(U)$ este mulțimea valorilor clare (crisp) a elementelor ce au grad de apartenență intermediar, cu un anumit grad de neclaritate, între 0 și 1:

$$Fr(U) = \{u \in U \mid \mu_A(u) \in (0,1)\} \quad (2.8)$$

2.4 Funcții de apartenență și numere fuzzy

Un **număr fuzzy** A este o submulțime fuzzy a mulțimii numerelor reale, care satisface condițiile: să fie o funcție de apartenență **convexă și continuă** cu suport mărginit [62].

Un număr fuzzy A se numește **număr fuzzy triunghiular** cu centrul c , lățimea la stânga $\alpha > 0$, lățimea la dreapta $\beta > 0$, dacă **funcția de apartenență** are forma:

$$\mu_A(x) = \begin{cases} 1 - \frac{c-x}{\alpha}, & c - \alpha \leq x \leq c \\ 1 - \frac{x-c}{\beta}, & c < x \leq c + \beta \\ 0, & \text{altfel} \end{cases} \quad (2.9)$$

sau utilizând funcțiile min și max :

$$\mu_A(x, c, \alpha, \beta) = \max\left(\min\left(\frac{c - \alpha + x}{\alpha}, \frac{c + \beta - x}{\beta}\right), 0\right) \quad (2.10)$$

Semnificația acestei mulțimi fuzzy cu centrul c este ” x este aproximativ egal cu c ”.

Un număr fuzzy A se numește **număr fuzzy trapezoidal** cu intervalul de toleranță $[c, d]$, lățimea la stânga $\alpha > 0$, lățimea la dreapta $\beta > 0$, dacă are următoarea **funcție de apartenență**:

$$\mu_A(x) = \begin{cases} 1 - \frac{c-x}{\alpha}, & c - \alpha \leq x \leq c \\ 1, & c < x \leq d \\ 1 - \frac{x-d}{\beta}, & d < x \leq d + \beta \\ 0, & \text{altfel} \end{cases} \quad (2.11)$$

sau utilizând funcțiile min și max :

$$\mu_A(x, c, d, \alpha, \beta) = \max\left(\min\left(\frac{c - \alpha + x}{\alpha}, 1, \frac{d + \beta - x}{\beta}\right), 0\right) \quad (2.12)$$

Semnificația acestei mulțimi fuzzy cu intervalul de toleranță $[c, d]$ este ” x este aproximativ între c și d ”.

Funcțiile triunghiulare/trapezoidale sunt generate pe baza funcțiilor liniare pe porțiuni [63].

Funcția de apartenență gaussiană: se definește prin intermediul a doi parametri $\{c, \sigma\}$ astfel:

$$\mu_A(x, c, \sigma) = e^{-\frac{1}{2} \left(\frac{x-c}{\sigma} \right)^2} \quad (2.13)$$

Parametrul c se numește centrul funcției de apartenență, iar σ determină lărgimea funcției de apartenență.

Funcția de apartenență de tip bell (clopot) astfel:

$$\mu_A(x, \sigma, b, c) = \frac{1}{1 + \left| \frac{x-c}{\sigma} \right|^{2b}} \quad (2.14)$$

unde c - centrul funcției de apartenență, σ - lărgimea funcției de apartenență, b - panta trecerii de la 0 la 1.

Funcția de apartenență de tip sigmoidală: se definește prin intermediul a doi parametri reali $\{a, b\}$ astfel:

$$\mu_A(x, a, b) = \begin{cases} 0, & x < a \\ 2 \cdot \left(\frac{x-a}{b-a} \right)^2, & a \leq x \leq \frac{a+b}{2} \\ 1 - 2 \cdot \left(\frac{x-a}{b-a} \right)^2, & \frac{a+b}{2} \leq x \leq b \\ 1, & x > b \end{cases} \quad (2.15)$$

Funcția de apartenență de tip sigmoidal poate fi deschisă la stânga sau la dreapta.

Funcțiile sigmoide/clopot (bell) sunt generate fie pe baza funcțiilor sigmoideale, fie pe baza funcțiilor polinomiale (pătratică sau cubice) [64].

Funcția de apartenență de tip singleton (un singur element) se definește astfel:

$$\mu_A(x) = \begin{cases} 1, & x = a \\ 0, & \text{in_rest} \end{cases} \quad (2.16)$$

2.5 Componentele structurii informaționale de bază fuzzy logic

I. Blocul de fuzzificare reprezintă blocul de intrare a informației, cu rol de transformare a acestora sub forma variabilelor lingvistice, a termenilor lingvistici și a funcțiilor de apartenență dintr-o valoare numerică.

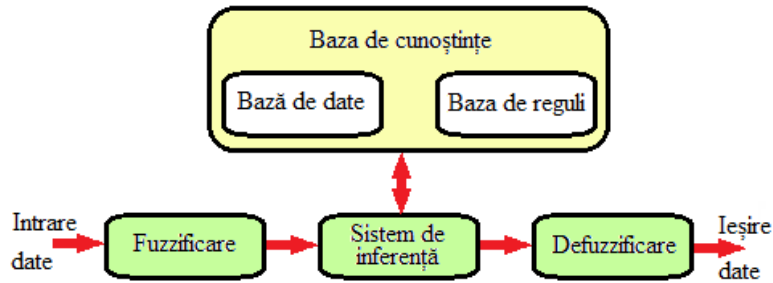


Figura 2. 1 Diagrama bloc a unui controler fuzzy logic

Sursa: [64]

Aceste informații fuzzy sunt comparate cu premisele regulilor de tipul "if (dacă) ...then (atunci) ..." cuprinse în baza de reguli și utilizate de mecanismul de inferență pentru activarea și aplicarea acestora [65].

II. Blocul baza de reguli conține setul de reguli de tipul "if (dacă) ...then (atunci) ..." stabilite de expert prin corelarea logică a mulțimilor fuzzy asociate variabilelor de intrare și ieșire. Baza de reguli realizează corespondența cu logica fuzzy a descrierii lingvistice. Numărul de reguli ale unei baze de reguli complete este dată de relația [65]:

$$n_{reguli} = \prod_{i=1}^n n_i \quad (2.17)$$

unde:

- ↳ n_i - numărul termenilor lingvistici definiți pentru fiecare variabilă lingvistică de intrare;
- ↳ n - numărul de variabile lingvistice de intrare;

III. Mecanismul de inferență sunt strategii de control sau tehnici de căutare, care traversează baza de cunoștințe pentru a trage concluzii [66], altfel spus operația logică care face legătura între o premisă și o concluzie se numește inferență logică [65]. Procesul de inferență manipulează simboluri prin selecția de reguli, potrivit simbolurile cu faptele și apoi stabilind fapte noi. Cele mai cunoscute metode de inferență sunt:

- a) înlănțuirea înapoi - este un **proces** condus de un **scop** în ordinea în care apare în baza de cunoștințe;
- b) înlănțuirea înainte - este un **proces** condus de **date**. Utilizatorul sistemului trebuie să dea datele disponibile înainte de începerea inferenței. Mecanismul de inferență încearcă să stabilească faptele așa cum apar în baza de date până când ultimul scop este atins.

IV. Blocul de defuzzificare asigură faptul că rezultatul obținut din blocul de decizie, o valoare fuzzy, este convertit într-o valoare fizică reală ce se va transmite procesului/elementului de execuție.

Se poate face analogia: codificare (fuzzyficare) – decodificare (defuzzyficare).

Deoarece modelele obținute nu sunt întotdeauna precise este nevoie de asigurarea robusteții astfel încât să se păstreze anumite proprietăți în cazul apariției unor variații între sistemul real și modelul utilizat. Robustețea, proprietate opusă senzitivității, depinde de proprietățile normei triunghiulare (t-normă sau t-conormă) alese, o normă care are proprietatea

de absorbție (min-max) fiind mai robustă decât o normă care nu are această proprietate (produs).

2.6 Metode de inferență

2.6.1 Metoda Mamdani

Această metodă folosește operatorul *min* pentru **implicație** și operatorul *min-max* pentru **compunere** [67], [68].

Regula de inferență, R_i , este:

„dacă u este A_i și v este B_i atunci w este C_i ” pentru $(\forall)u \in U, v \in V, w \in W; i = \overline{1, n}$.

Deci $R_i = (A_i, B_i) \rightarrow C_i$ este definită de $\mu_{R_i} = \mu_{(A_i B_i) \rightarrow C_i}(u, v, w)$.

Dacă fiecare set de date de intrare este definit de o singură funcție de apartenență $u = u_0, v = v_0$ obținem:

$$\mu_{C_i}(w) = [\mu_{A_i}(u_0), \mu_{B_i}(v_0)] \rightarrow \mu_{C_i}(w) \quad (2.18)$$

Operatorul *min* pentru **implicație** este definit prin relațiile:

$$\begin{aligned} \mu_{C_i}(w) &= \alpha_i \wedge \mu_{C_i}(w) \text{ unde } \alpha_i = \mu_{A_i}(u_0) \wedge \mu_{B_i}(v_0) \\ \mu_{C'}(w) &= \bigvee_{i=1}^n [\alpha_i \wedge \mu_{C_i}(w)] = \bigvee_{i=1}^n \mu_{C_i}(w) \\ C' &= \bigcup_{i=1}^n C_i \end{aligned} \quad (2.19)$$

Operatorul *min-max* pentru **compunere** este definit prin relațiile:

$$\begin{aligned} \mu_{C_i}(w) &= \min\{\mu_{A_i}(u) \circ \min_u \{\mu_{A_i}(u), \mu_{C_i}(w)\}, \mu_{B_i}(v)\} \circ \min_v \{\mu_{B_i}(v), \mu_{C_i}(w)\}\} \\ \Rightarrow \mu_{C_i}(w) &= \alpha \wedge \mu_{C_i}(w) \end{aligned} \quad (2.20)$$

2.6.2 Metoda Takagi-Sugeno-Kang

Această metodă a fost propusă de Takagi, Sugeno și Kang (TSK). O regulă fuzzy în acest model are forma:

$$\text{IF } u \text{ is } A \text{ and } v \text{ is } B \text{ THEN } w = f(u, v)$$

unde A și B sunt mulțimi fuzzy iar $f(u, v)$ este o funcție strictă (crisp). Deoarece $f(u, v)$ este un polinom de variabile u și v această metodă necesită intrări impuls (singleton) [68].

În acest caz nu este necesară definirea de termeni lingvistici pentru ieșirea sistemului, iar funcția $f(u, v)$ este, de cele mai multe ori, o funcție neliniară. În cazul sistemelor liniare, funcția $f(u, v)$ este o combinație liniară a intrărilor sistemului fuzzy [69].

Fie cele două reguli ale R-bazei cu forma:

$$R_1 : \text{IF } u \text{ is } A_1 \text{ and } v \text{ is } B_1 \text{ THEN } w_1 = f_1(u, v) = p_1 \cdot u + q_1 \cdot v + r_1$$

$$R_2 : \text{IF } u \text{ is } A_2 \text{ and } v \text{ is } B_2 \text{ THEN } w_2 = f_2(u, v) = p_2 \cdot u + q_2 \cdot v + r_2$$

unde: p_i, q_i, r_i sunt constante.

Valoarea obținută în urma aplicării regulii R_i pentru intrarea impuls (singleton) u_0, v_0 este obținută valoarea numerică $f_i(u_0, v_0)$ cu gradul de potrivire α_i . Rezultatul agregării acestor ieșiri este dat de media ponderată:

$$w_0 = \frac{\alpha_1 \cdot f_1(u_0, v_0) + \alpha_2 \cdot f_2(u_0, v_0)}{\alpha_1 + \alpha_2} = \frac{\alpha_1 \cdot w_1 + \alpha_2 \cdot w_2}{\alpha_1 + \alpha_2} \quad (2.21)$$

2.7 Metode de defuzzyficare

Obținerea unei valori stricte (crisp) dintr-o mulțime fuzzy, ca valoare reprezentativă, se realizează prin conversia fuzzy-crisp de către modulul de defuzzyficare. Acțiunea de control fuzzy Y dedusă din sistemul de control fuzzy este transformată într-o acțiune de control strictă:

$$y_0 = \text{defuzzyfier}(Y) \quad (2.22)$$

unde y_0 este rezultatul controlului nonfuzzy iar *defuzzyfier* este operatorul de defuzzyficare [53], [70].

2.7.1 Defuzzyficare de tip Mamdani

Metodele de defuzzyficare de tip Mamdani cele mai utilizate sunt [53], [65], [71]:

1. Centrul ariei (centroid of area – COA)

Metoda de defuzzyficare returnează o ieșire prin calcularea centrului de simetrie al zonei delimitate prin agregarea consecințelor setului fuzzy astfel:

$$y_{COA} = \frac{\int_Y y \cdot \mu_Y(y) \cdot dy}{\int_Y \mu_Y(y) \cdot dy} \quad \text{- continuu} \quad (2.23)$$

$$y_{COA} = \frac{\sum_{i=1}^N y \cdot \mu_Y(y) \cdot dy}{\sum_{i=1}^N \mu_Y(y) \cdot dy} \quad \text{- discret} \quad (2.24)$$

2. Bisectoarea ariei (bisector of area – BOA)

Această operație poate fi exprimată astfel:

$$\int_{\alpha}^{y_{BOA}} \mu_Y(y) \cdot dy = \int_{y_{BOA}}^{\beta} \mu_Y(y) \cdot dy \quad (2.25)$$

unde $\alpha = \min\{u | u \in U\}$, $\beta = \max\{u | u \in U\}$.

COA și BOA se utilizează în aplicații de control, deoarece nu se produc salturi în suprafața de control [65].

3. Cel mai mic maxim în valoare absolută (smallest of maximum – SOM)

Această metodă generează o ieșire strictă prin luarea celei mai mici valori care să ofere gradul maxim de apartenență setului fuzzy agregat.

$$y_{SOM} = \min\{y | \mu_Y(y) = \max(\mu_Y(y))\} \quad (2.26)$$

4. Cel mai mare maxim în valoare absolută (largest of maximum - LOM)

Această metodă generează o ieșire strictă prin luarea celei mai mari valori care să ofere gradul maxim de apartenență setului fuzzy agregat.

$$y_{LOM} = \max \{y | \mu_Y(y) = \max(\mu_Y(y))\} \quad (2.27)$$

5. Media maximelor (mean of maximum – MOM)

În această defuzzyficare, media maximă este luată ca o ieșire clară.

$$y_{MOM} = \frac{y_{SOM} + y_{LOM}}{2} \quad (2.28)$$

MOM, LOM, SOM se utilizează în aplicații de decizie deoarece se pot produce salturi în suprafața de control.

2.7.2 Defuzzyficare de tip Takagi-Sugeno-Kang

Metoda permite reducerea numărului de relații liniare și interconectarea lor. Este foarte important reducerea numărului de relații liniare în cazurile multidimensionale. Algoritmul de identificare al implicației este divizat în trei secvențe [72]:

- 1) Alegerea variabilelor de bază;
- 2) Identificarea parametrilor de bază;
- 3) Identificarea consecințelor parametrilor;

În cazul sistemelor fuzzy cu mecanism de inferență TSK, compoziția regulilor se obține cu ajutorul unei funcții, în loc de defuzzyficare deci nu conțin defuzzyficator [69].

a) weighted average WA

Această metodă de defuzzyficare generează rezultatul final pentru o ieșire FIS Sugeno prin metoda ponderării centrelor de greutate ale suprafețelor individuale.

$$y_{WA} = \frac{\sum_{i=1}^M w_i \cdot y_i}{\sum_{i=1}^M w_i} \quad (2.29)$$

b) weighted sum WS

Pentru a reduce calculul WA, metoda WS are nevoie ca regulă doar de suma ieșirilor ponderate.

$$y_{WS} = \sum_{i=1}^M w_i \cdot y_i \quad (2.30)$$

CAPITOLUL 3 DETERMINAREA VARIANTEI OPTIME A RNA-BP PENTRU OBTINEREA UNEI PRECIZII RIDICATE DE PREDICȚIE A PARAMETRILOR ÎN CAZUL OȚELULUI UTILIZAT PENTRU ȘINA DE CALE FERATĂ

3.1 Metodologia cercetării

Pentru determinările experimentale au fost realizate epruvete din toate cele trei părți ale unei șine de cale ferată tip Vignole (ciupercă, talpă, inimă – Figura 3.1) [73]. Scopul determinărilor experimentale a fost realizarea curbei caracteristice, precum și determinarea parametrilor elastici ai materialului din care sunt realizate aceste șine (oțel).

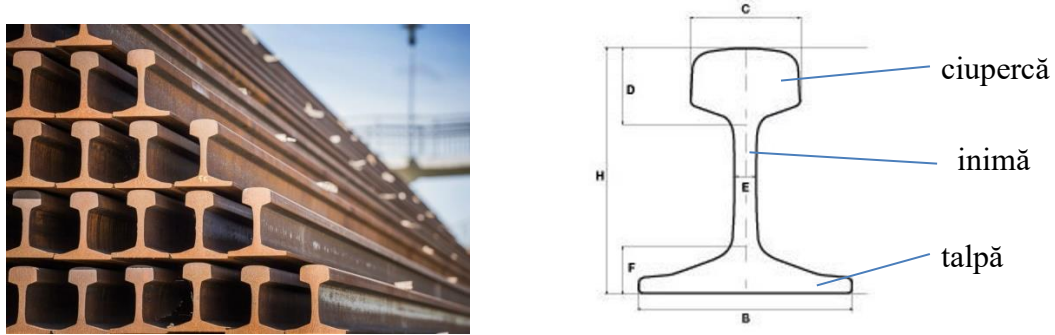


Figura 3. 1 Șină de cale ferată de tip Vignole

Încercările mecanice pentru determinarea curbei caracteristice au fost efectuate pe mașina universală de încercat Instron 8801 (Figura 3.2), conform cu standardele SR EN 10002-1/1994 și SR EN 10002-5/1995 [74], [75]. Principalele caracteristici ale acestei mașini sunt: forța maximă în regim static este de 100 [kN], forța maximă în regim dinamic este de ± 50 [kN], bacurile au prindere hidraulică etc. Testele care pot fi realizate cu ajutorul acestei mașini sunt teste de tracțiune la temperatura mediului ambiant și la temperaturi ridicate, teste de compresiune, teste de încovoiere în trei puncte, teste de oboseală cu regimuri variabile.

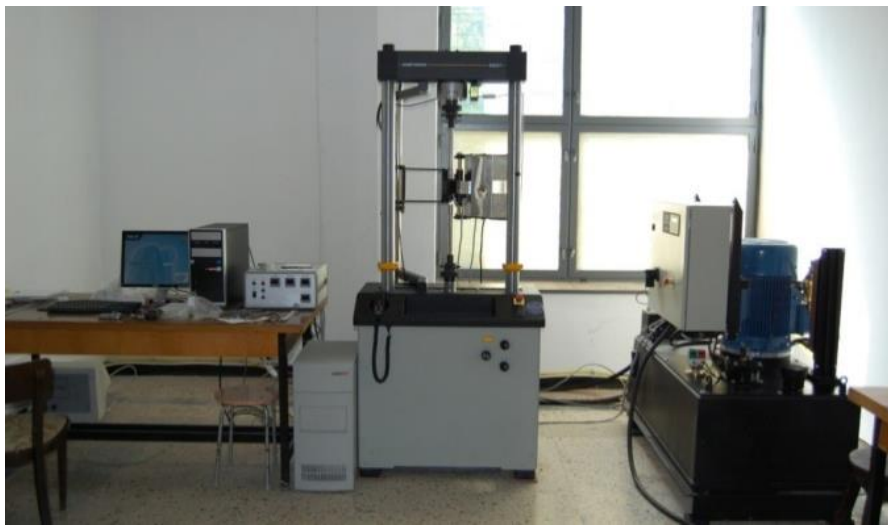


Figura 3. 2 Mașina de încercat Instron 8801

În Figura 3.3 este prezentat tipul de epruvetă utilizat pentru încercarea experimentală (a)- înainte, b) după încercarea distructivă). Testele au avut loc la temperatura mediului ambiant, cu o viteză de deformare de 0,1 [mm/min]. Forma și dimensiunile epruvetelor au fost conforme cu standardele amintite.

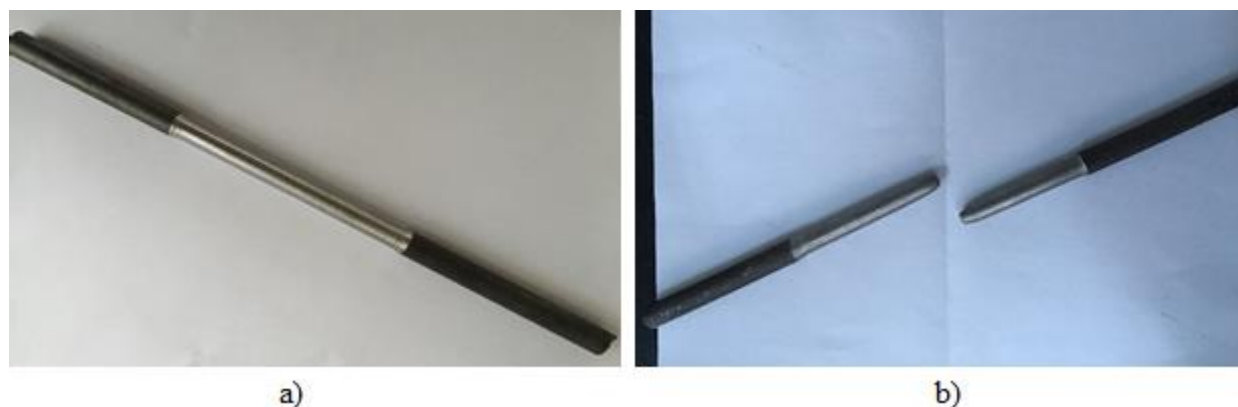


Figura 3. 3 Tipul epruvetei utilizate în încercarea la tracțiune

Curba caracteristică obținută în urma determinărilor experimentale pentru oțelul din care este realizată șina de cale ferată de tip Vignole este prezentată în Figura 3.4.

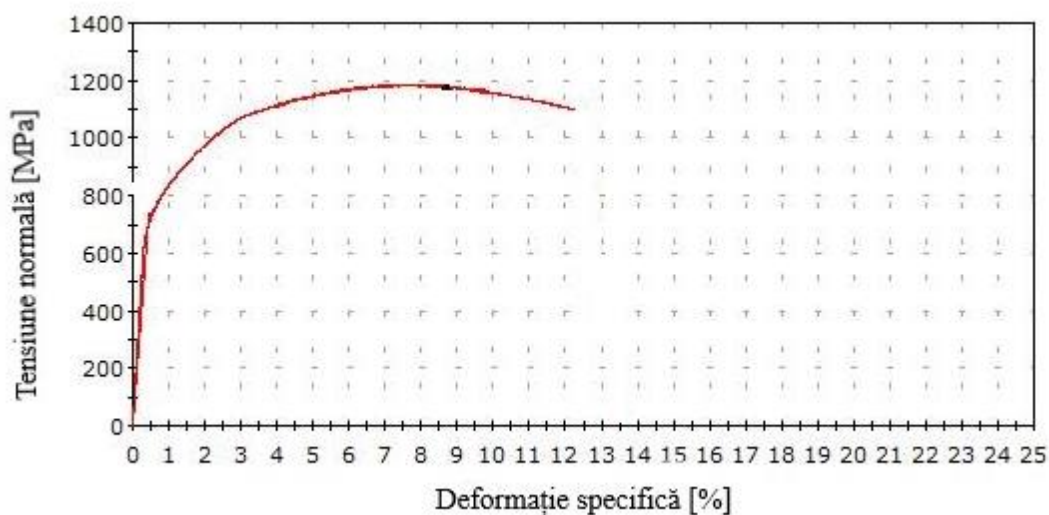


Figura 3. 4 Curba caracteristică la tracțiune pentru oțelul șinei de cale ferată de tip Vignole

În Tabelul 3.1 sunt definite caracteristicile medii obținute în urma încercărilor pentru acest material.

Tabelul 3. 1 Valorile caracteristicilor elastice ale oțelului șinei de cale ferată

Modul de elasticitate longitudinal	Modul de elasticitate longitudinal – zona liniară	Limita de curgere	Rezistența la rupere	Deformația specifică
MPa	MPa	MPa	MPa	%
204383	176484.4	723.7448	1163.677	8.710183

Studiul cuprinde două seturi de date determinate în zile diferite pentru epruvete din oțel utilizat la construcția șinei de cale ferată.

Setul de date obținut în prima zi conține datele de la 7 experimente (set redus de date - SRD), Tabelul 3.2, iar cel din a doua zi de la 17 experimente (set lărgit de date - SLD), Tabelul 3.3.

Studiul este împărțit în 4 categorii de interes, defalcate astfel:

A. *Predicția parametrilor SRD pe baza SLD în două situații:*

1. predicția simultană a două variabile de ieșire;
2. predicția individuală a avariabilelor de ieșire;

B. *Predicția parametrilor SLD pe baza SRD în două situații:*

1. predicția simultană a două variabile de ieșire;
2. predicția individuală a avariabilelor de ieșire;

Parametrii sunt următorii:

a) variabile de intrare:

- ↪ Modul de elasticitate longitudinal - Zona liniară (Segment 0.01 % - 0.1 %) – MELZL;
- ↪ Limita de curgere (Offset 0.2 %) – LC;
- ↪ Deformația specifică la rupere – DSR;

b) variabile de ieșire:

- ↪ Modul de elasticitate longitudinal – MEL;
- ↪ Rezistența la rupere – RR;

Tabelul 3. 2 Setul redus de date - SRD

Variabile intrare			Variabile ieșire	
MELZL	LC	DSR	RR	MEL
Modul de elasticitate longitudinal - Zona liniara (Segment 0.01 % - 0.1 %)	Limita de curgere (Offset 0.2 %)	Deformatia specifica la rupere	Rezistenta la rupere	Modul de elasticitate longitudinal
[Mpa]	[Mpa]	[%]	[Mpa]	[Mpa]
266938.4415	909.16800	6.73181	1258.41888	195289.90
142081.4352	873.56893	6.95277	1252.10531	202021.87
147598.6538	918.76746	8.70982	1267.44398	201570.27
128476.4219	775.12079	9.24478	1139.83447	204094.55
131657.3247	777.17458	6.89265	1161.85106	202539.68
149911.4903	687.78206	8.44409	1090.16922	199735.93
125277.2586	689.74692	7.73038	1088.15691	204267.87

În Tabelul 5.2 se observă, în cazul MELZL, o diferență foarte mare între valoarea din primul rând față de restul valorilor, diferență care este de 2,13 ori mai mare decât cea mai mică valoare și de 1,78 ori față de a doua valoare mare. Având în vedere acest fapt ne putem aștepta la apariția unei erori mari în cazul în care RNA are la intrare SLD.

Analiza predicției este realizată în două cazuri: cu erori grosolane (CU_EG) și fără erori grosolane (FĂRĂ_EG), sunt eliminate capetele de interval ale erorilor, adică două erori.

Tabelul 3. 3 Setul lărgit de date - SLD

Variabile intrare			Variabile ieșire	
MELZL	LC	DSR	RR	MEL
Modul de elasticitate longitudinal - Zona liniara (Segment 0.01 % - 0.1 %)	Limita de curgere (Offset 0.2 %)	Deformația specifică la rupere	Rezistența la rupere	Modul de elasticitate longitudinal
[Mpa]	[Mpa]	[%]	[Mpa]	[Mpa]
169893.0493	763.39922	8.71941	1177.70286	208706.7099
202800.6382	776.05097	8.08814	1227.01800	208024.6367
193592.1713	866.37686	7.80482	1289.44324	214326.2983
194980.6158	836.05789	9.55074	1304.35850	213601.1487
203501.5797	814.88673	2.76630	1098.86958	210444.9718
197117.9038	688.91168	9.78879	1145.52285	209234.8261
208873.8036	552.61713	8.27557	1031.60272	212689.9210
124485.1198	988.73848	7.40952	1327.75055	202915.1359
126972.9974	687.56886	9.72804	1120.87963	197574.6996
147579.2410	583.55417	10.30533	1054.89244	198664.6603
193939.7237	742.19193	8.93897	1205.40259	212606.8263
206985.1896	620.50883	10.03973	1080.75853	212368.4164
198996.8969	649.73631	9.11668	1117.22832	210066.8115
206454.2157	567.86289	9.76458	1135.10562	186225.9781
196608.9587	370.10161	9.58518	1129.44314	130236.1207
208434.7859	481.59656	10.54079	1073.16248	200104.9092
200921.4857	659.73628	9.80222	1183.12192	203676.0311

3.2 Predicția parametrilor SLD pe baza SRD

În acest paragraf realizăm predicția parametrilor SLD pe baza SRD și interpretăm rezultatele obținute. Numărul de cicluri de antrenare optim este de 125.000 sau 150.000.

Tabelul 3. 4 Caracteristicile arhitecturii RNA în cazul predicției SRD pe baza SLD

Predicția SRD pe baza SLD	Simultan RR și MEL	Individual RR	Individual MEL
Tipologia setărilor - Parametri			
Număr variabile intrare	3		
Număr variabile ieșire	2	1	
Număr straturilor ascunse	5		
Numărul de noduri la primul strat ascuns	20		
Numărul de noduri la al doilea strat ascuns	20		
Numărul de noduri la al treilea strat ascuns	20		
Numărul de noduri la al patrulea strat ascuns	20		
Numărul de noduri la al cincilea strat ascuns	20		
Setările de instruire - Parametri			
Rata de învățare	0.01		
Momentul de impuls	0.1		
Fucția de transfer	Tangentă Hiperbolică		
Numărul maxim de cicluri de antrenare	150000	125000	
Eroarea țintă	0.00001		
Metoda de inițializare a pragului	Aleator		
Metoda de inițializare a importanței (greutății)	Aleator		
Intervalul de analiză pentru evoluție (cicluri)	500		
Stările de instruire - Parametri			
Suma erorilor	0.0008359940	0.0001181241	0.0010399855
Eroarea medie per ieșire per set de date	0.0000245880	0.0000069485	0.0000611756
Timp procesare (sec)	1146	1048	971

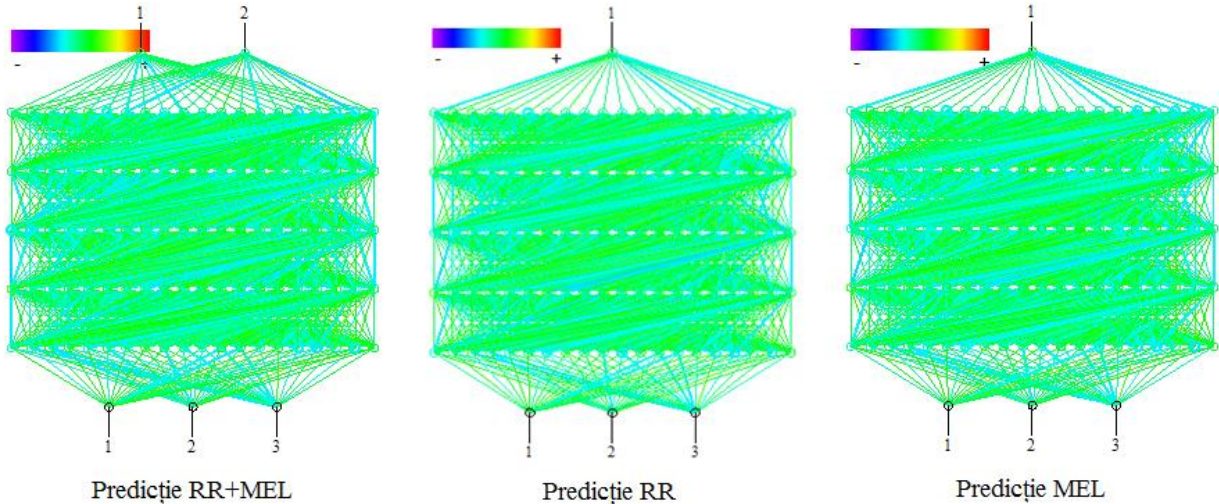


Figura 3. 5 Arhitectura și structura fluxului informațional a RNA în cazul SRD pe baza SLD

În figura 3.5 se observă că majoritatea importanței poderilor între conexiunile dintre noduri indiferent de strat (intrare, ascuns, ieșire) este centrat în jurul valorii de $\pm 0,5$, deci nu există flux informațional către extremele domeniului, respectiv ± 1 .

Tabelul 3. 5 Valorile obținute pentru coeficientul de regresie, pantă și intercepția axei y în cazul SRD pe baza SLD

Predicția SRD pe baza SLD	Variabila	Coeficientul de regresie (CR)	Panta	Intercepția axei y
SIMULTAN	RR	0.9373072	0.939086	0.01026814
	MEL	0.9081377	0.903298	0.01912414
	MEDIA CR	0.923		
INDIVIDUAL	RR	0.9381203	0.933981	0.00921902
	MEL	0.9116638	0.910060	0.01283507
	MEDIA CR	0.925		

Din punct de vedere al coeficientului de regresie r^2 acesta este apropiat de valoarea de 0,93 în ambele cazuri, cu un plus de precizie mai ridicat în cazul predicției individuale. Se remarcă în ambele cazuri că coeficientului de regresie r^2 pentru parametrul RR depășește 0,93.

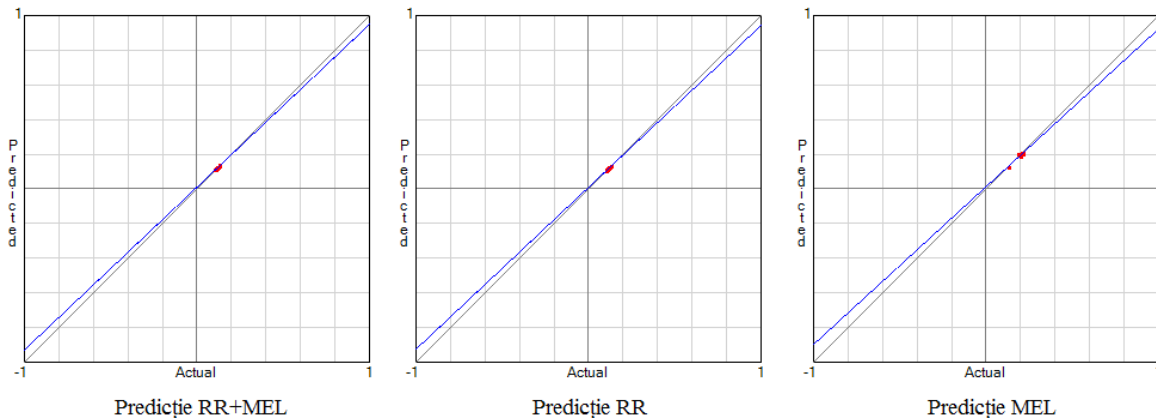


Figura 3. 6 Poziționarea setului de date predicționat pe panta de regresie în cazul SRD pe baza SLD

Din punct de vedere al pantei de regresie faptul că setul de date predicționat este distribuit pe poli (Figura 3.6 MEL) și nu este concentrat (Figura 3.6 RR+MEL și RR) conduce la o precizie de scăzută.

Tabelul 3. 6 Tabel cu sinteza mărimii erorilor predicționate în cazul SRD pe baza SLD fără eliminarea erorilor grosolane

		CU EG		FĂRĂ EG	
Predicția SRD pe baza SLD		RR	MEL	RR	MEL
SIMULTAN	eroare globala	4.24%	1.43%	3.86%	-0.19%
	ecart erori	9.96%	19.27%	7.11%	4.94%
	interval erori	0.44 ÷ 9.52 %	-3.9 ÷ 15.36 %	0.70 ÷ 6.41 %	-2.54 ÷ 2.40 %
	media erorilor individuale	4.15%	1.51%	3.82%	1.57%
INDIVIDUAL	eroare globala	0.01%		0.28%	
	ecart erori	9.63%		7.46%	
	interval erori	-4.42 ÷ 5.21 %		-4.21 ÷ 3.25 %	
	media erorilor individuale	-0.11%		2.83%	
INDIVIDUAL	eroare globala		4.69%		3.15%
	ecart erori		17.97%		6.32%
	interval erori		-0.19 ÷ 17.78 %		0.88 ÷ 5.45 %
	media erorilor individuale		4.77%		3.15%

Concluzie: în cazul eliminării erorilor grosolane: precizia globală a predicției este mai bună cu aproximativ 28,74%, ecartul și intervalul erorilor scade semnificativ iar media erorilor individuale scade cu 4÷8%.

3.3 Predicția parametrilor SLD pe baza SRD

În acest paragraf realizăm predicția parametrilor SRD pe baza SLD și interpretăm rezultatele obținute.

În Figura 5.7 se observă că caracteristica privind importanța fluxului de date între nodurile dintre straturi este asemănătoare cu cea prezentată în Figura 5.5, deci nu conține elemente care să iasă în evidență.

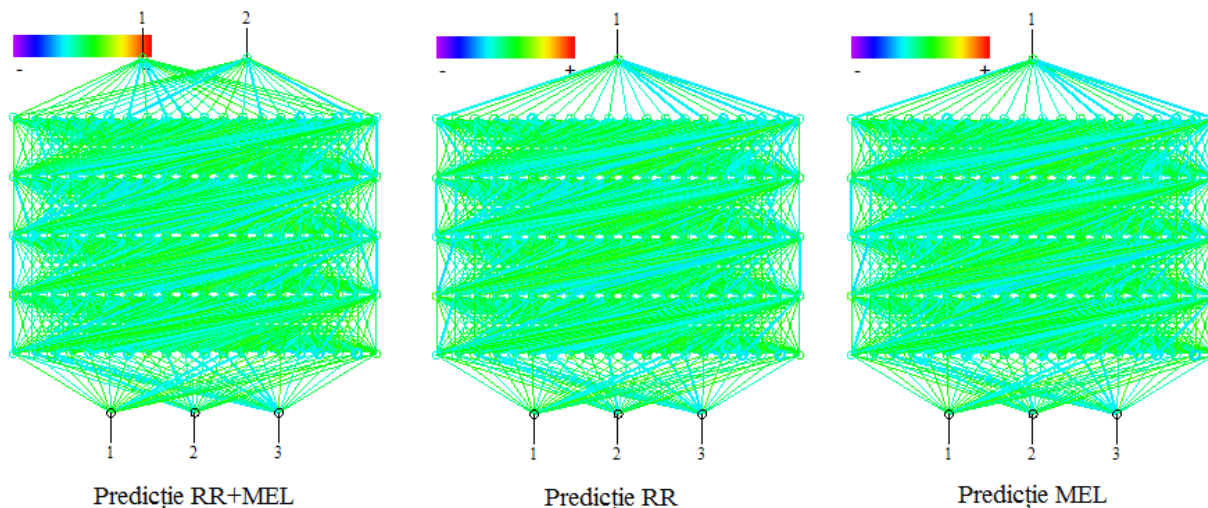


Figura 3. 7 Arhitectura și structura fluxului informațional a RNA în cazul SLD pe baza SRD

Tabelul 3. 7 Caracteristicile arhitecturii RNA în cazul predicției SLD pe baza SRD

Predicția SLD pe baza SRD	Simultan RR și MEL	Individual RR	Individual MEL
Tipologia setărilor - Parametrii			
Număr variabile intrare	3		
Număr variabile ieșire	2	1	
Număr straturilor ascunse	5		
Numărul de noduri la primul strat ascuns	20		
Numărul de noduri la al doilea strat ascuns	20		
Numărul de noduri la al treilea strat ascuns	20		
Numărul de noduri la al patrulea strat ascuns	20		
Numărul de noduri la al cincilea strat ascuns	20		
Setările de instruire - Parametrii			
Rata de învățare	0.01		
Momentul de impuls	0.1		
Fucția de transfer	Tangentă Hiperbolică		
Numărul maxim de cicluri de antrenare	150000	7564	95364
Eroarea țintă	0.00001		
Metoda de inițializare a pragului	Aleator		
Metoda de inițializare a importanței (greutății)	Aleator		
Intervalul de analiză pentru evoluție (cicluri)	500		
Stările de instruire - Parametrii			
Suma erorilor	0.0000143801	0.0000099976	0.0000099994
Eroarea medie per ieșire per set de date	0.0000010272	0.0000014282	0.0000014285
Timp procesare (sec)	494	55	477

În cazul predicției SLD pe baza SRD se evidențiază ecartul mare de cicluri de antrenare, de la 7.500 până la 150.000. În cazul predicției simultane numărul de cicluri de antrenare rămâne constant, iar cel pentru predicția individuală variază foarte mult.

Tabelul 3. 8 Valorile obținute pentru coeficientul de regresie, pantă și intercepția axei y în cazul SLD pe baza SRD

Predicția SLD pe baza SRD	Variabila	Coeficientul de regresie (CR)	Panta	Intercepția axei y
SIMULTAN	RR	0.9852455	0.989152	0.00127963
	MEL	0.8455955	0.850053	0.03019339
	MEDIA CR	0.915		
INDIVIDUAL	RR	0.9734746	0.986317	0.00161427
	MEL	0.8257339	0.848064	0.03059381
	MEDIA CR	0.900		

Se remarcă faptul că coeficientul de regresie are al parametrului RR ($0,93_{optim} < 0,973 < 1$) diferă foarte mult de al parametrului MEL ($0,825 < 0,93_{optim} < 1$), deci pentru parametrul RR se obțin performanțe superioare față de MEL. Acest fapt este susținut de panta de regresie predicționată care este aproape identică cu panta de regresie de referință (Figura 5.8 RR+MEL și RR).

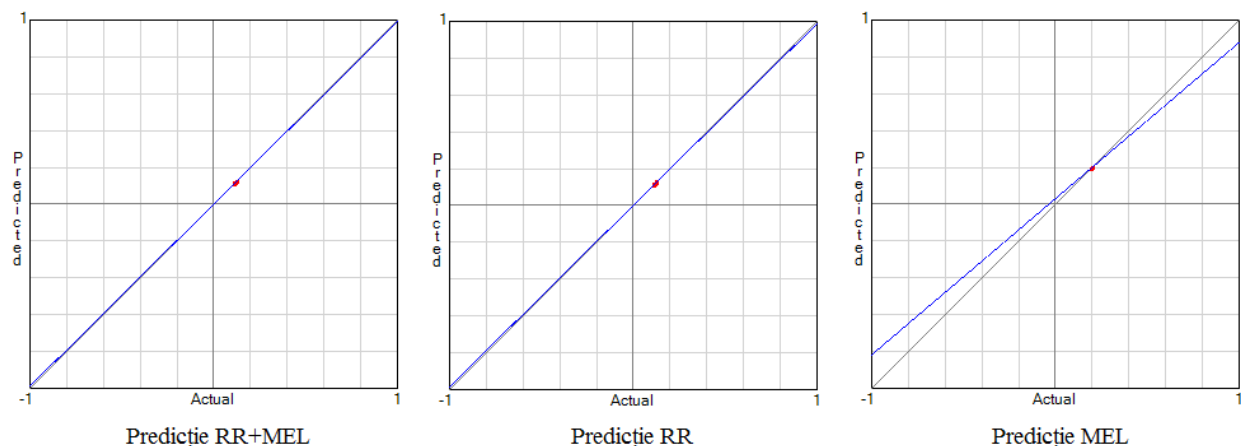


Figura 3. 8 Poziționarea setului de date predicționat pe panta de regresie în cazul SLD pe baza SRD

Tabelul 3. 9 Tabel cu sinteza mărimii erorilor predicționate în cazul SLD pe baza SRD

Predicția SLD pe baza SRD		CU EG		FĂRĂ EG	
		RR	MEL	RR	MEL
SIMULTAN	eroare globala	7.65%	1.04%	7.86%	2.91%
	ecart erori	27.76%	59.24%	16.49%	13.16%
	interval erori	-19.66 ÷ 8.10 %	-6.67 ÷ 52.57 %	-15.01 ÷ 1.47 %	-6.64 ÷ 6.51 %
	media erorilor individuale	8.82%	6.32%	8.15%	3.21%
INDIVIDUAL	eroare globala	2.40%		1.91%	
	ecart erori	33.16%		12.56%	
	interval erori	-9.51 ÷ 23.66 %		-7.38 ÷ 19.93 %	
	media erorilor individuale	7.59%		6.39%	
INDIVIDUAL	eroare globala		0.39%		1.88%
	ecart erori		74.70%		17.06%
	interval erori		-7.52 ÷ 67.18 %		-7.39 ÷ 9.67 %
	media erorilor individuale		8.78%		4.97%

Concluzie: în cazul eliminării erorilor grosolane: precizia globală a predicției este mai scăzută cu aproximativ 24,1%, ecartul și intervalul erorilor scade foarte mult iar media erorilor individuale în cazul parametrului MEL scade cu aproximativ 50%, fapt ce conduce la creșterea preciziei predicției.

CONCLUZII FINALE

↪ alegerea setului de date supus analizei RNA-BP trebuie să fie realizat de persoane cu o foarte bună experiență în domeniul statistic și al prelucrărilor datelor. Acestea trebuie să înțeleagă interdependențele dintre parametrii, a modului cum se influențează reciproc, astfel încât în momentul în care apar erori mari în analiza datelor să știe motivul apariției (necorelarea pentru situații aproximativ identice) și să le elimine din concluziile generale.

↪ studiul a demonstrat că poate fi utilizată aceeași arhitectură și caracteristici pentru seturi de date cu valori apropiate pentru aceeași poziție de rotație și greutate diferite ale mesei rotative.

↪ eroarea obținută este de aproximativ 10% și un coeficient de regresie în majoritatea cazurilor de peste 0,92. Există cazuri când utilizarea unui sigur parametru de ieșire (predicție) extras dintr-un set de parametri aflați într-o interdependență foarte puternică să conducă la o creștere accelerată a gradului de apariție al erorilor grosolane.

↪ metoda hibridă de analiză abordată de tip ANFIS, este o metodă robustă caracterizată de o precizie a erorii între 5÷10% pentru valori care prezintă interes în predicție.

↪ SRD pe baza SLD în cazul eliminării erorilor grosolane prezintă o precizie globală a predicției mai bună cu aproximativ 25%, ecartul și intervalul erorilor scade semnificativ iar media erorilor individuale scade cu 5%.

↪ SLD pe baza SRD în cazul eliminării erorilor grosolane prezintă o globală a predicției mai scăzută cu aproximativ 25%, ecartul și intervalul erorilor scade foarte mult iar media erorilor individuale în cazul parametrului MEL scade cu aproximativ 50%, fapt ce conduce la creșterea preciziei predicției.

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/Inteligența_artificială_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=xfXsCAAQBAJ&pg=PA592&lpg=PA592&dq=Static+neuron+is+the+simplest+model+of+artificial+neuron&source=bl&ots=1uOYwdAvl-&sig=jq20kt30bmtV-UUKBPz4gyg8hqw&hl=ro&sa=X&ved=0ahUKEwjOwsSc5MrVAhUCshQKHf5LAgIQ6AEIWDAAH#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://citeserx.ist.psu.edu/viewdoc/download?doi=10.1.1.705.8811&rep=rep1&type=pdf>
- [17] Proștean 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%20C%20Liang%20Jin%20C%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] Kuniyiko 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 politehnică 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 politehnică București, Facultatea Ingineria și Managementul Sistemelor Tehnologice, aprilie 2018
- [48] Lotfali Askar Zadeh, *Fuzzy sets**, Information and Control, Vol. 8, Issue 3, June 1965, pp.338-353, <http://www.sciencedirect.com/science/article/pii/S001999586590241X?via%3Dihub>
- [49] Francis Jeffrey Peletier, *Metamathematics of Fuzzy Logic*, The Bulletin of Symbolic Logic, Vol. 6, No.3, Sep. 2000, pp. 342-346, <http://www.sfu.ca/~jeffpell/papers/ReviewHajek.pdf>;

- [50] *** https://ro.wikipedia.org/wiki/Logic%C4%83_fuzzy
- [51] ***
http://www.mpt.upt.ro/doc/curs/gp/Sisteme_inteligente_in_electrotehnica/Sisteme_Fuzzy_cap2.pdf
- [52] Laura-Nicoleta Ivanciu, *Sisteme inteligente de suport decizional*, 2014,
http://www.bel.utcluj.ro/dce/didactic/sln/02_MultimiFuzzy.pdf
- [53] Chennakesava R. Alavala, *Fuzzy logic and neural networks. Basic concepts & applications*, New Age International Publishers, 2008,
http://www.academia.edu/1435724/Fuzzy_Logic_and_Neural_Networks_by_Chennakesava_R._Alavala
- [54] Sorin Georgescu, *Sisteme fuzzy. Fuzzy ARTMAP*, Revista Informatică Economică, Nr.4, 1997,
<http://revistaie.ase.ro/content/4/5.pdf>
- [55] Chuen Chien Lee, *Fuzzy Logic in Control Systems: Fuzzy Logic Controller – Part I*, IEEE Transactions on Systems, Man, and Cybernetics (Volume: 20, Issue: 2, Mar/Apr 1990),
<http://ieeexplore.ieee.org/document/52551/>
- [56] Răzvan Andone, Angel Cațaron, *Inteligență computațională*, Universitatea "Transilvania" Brasov, 2002, http://vega.unitbv.ro/~cataron/Publications/curs_rn.pdf
- [57] Andreea-Mihaela Roman, *Contribuții la modelarea sistemelor de conducere automată utilizând algoritmi neuro-fuzzy*, Teză de doctorat, Universitatea Tehnică de Construcții București, 2014,
http://instal.utcb.ro/Documente_Website/teze/Teza_Andreea_Iftene.pdf
- [11] Qiu Jie, *Scheduling flexible manufacturing systems using fuzzy heuristics*, Teză de doctorat, The University of Hong Kong, 2003, <http://hdl.handle.net/10722/35271>,
<http://hub.hku.hk/handle/10722/35271>,
http://www.researchgate.net/publication/29843232_Scheduling_flexible_manufacturing_systems_using_fuzzy_heuristics
- [59] Florin Leon, *Inteligența artificială*, http://florinleon.byethost24.com/curs_ia.htm
- [60] Daniela Popescu, *Tehnici de inteligența artificială. Curs și aplicații*.
<http://elth.ucv.ro/student1/Cursuri/Popescu%20Daniela/Tehnici%20de%20inteligenta%20artificiala/Tehnici%20de%20inteligenta%20artificiala.pdf>
- [61] Timothy J. Ross, *Fuzzy Logic with Engineering Applications, Second Edition*, John Wiley & Sons, Ltd ISBNs: 0-470-86074-X (HB); 0-470-86075-8 (PB),
https://www.researchgate.net/publication/309967685_Fuzzy_Logic_with_Engineering_Applications_John_Wiley_Sons_Ltd_The_Atrium_Southern_Gate_Chichester_West_Sussex_PO19_8SQ
- [62] *** http://www.ai.pub.ro/resources/files/RNSF/RNSF_curs8n.pdf
- [63] *** <https://www.scribd.com/presentation/98961045/Logica-Fuzzy>
- [64] 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

- [65] Stelian-Emilian Oltean, *Control inteligent și adaptiv*, <http://docshare04.docshare.tips/files/30508/305082796.pdf>
- [66] C.S. Krishnamoorthy, S. Rajeev, *Artificial Intelligence and Expert Systems for Engineers*, CRC Press, CRC Press LLC, ISBN: 0849391253, 1996, https://doc.lagout.org/science/0_Computer%20Science/8_Electronics%20%26%20Robotics/Artificial%20Intelligence%20and%20Expert%20Systems%20for%20Engineers.pdf
- [67] Mihaela Colhon, *Elemente de logică fuzzy*, Craiova, 2012, <http://inf.ucv.ro/~ghindeanu/lab/sicc/carteb5.pdf>
- [68] Chi-Yuan Yeh, *Fuzzy Inference*, http://itlab.ee.nsysu.edu.tw/ch/chap/99a_AI/Fuzzy_ch9.ppt.
- [69] *** <https://myslide.es/documents/what-is-fuzzy.html>
- [70] Ion Iancu, *A Mamdani Type Fuzzy Logic Controller*, <http://cdn.intechopen.com/pdfs/34221.pdf>
- [71] Yang Wang, Yanyan Chen, *A Comparison of Mamdani and Sugeno Fuzzy Inference Systems for Traffic Flow Prediction*, Journal of Computers, Vol. 9, No. 1, January 2014, <http://ojs.academypublisher.com/index.php/jcp/article/view/jcp09011221>
- [72] Tomohiro Takagi, Michio Sugeno, *Fuzzy identification of systems and its applications to modeling and control*, IEEE Transactions on Systems, Man, and Cybernetics (Volume: SMC-15, Issue: 1, Jan.-Feb.1985), https://www.scss.tcd.ie/khurshid.ahmad/Teaching/Lectures_on_Fuzzy_Logic/Takagi%20Sugeno%20Modelling.pdf
- [73] SR EN 13674 – 1/2011
- [74] SR EN 10002 – 1/1994
- [75] SR EN 10002 – 5/1995