

# MODEL DE CERCETARE INTERDISCIPLINARĂ A FACTORILOR INTERNI ȘI EXTERNI AI FIRMEI UTILIZÂND REȚELE NEURONALE

Daniel-Petru GHENCEA<sup>1</sup>

Conducător științific: Prof.dr.ing. Miron ZAPCIU

**REZUMAT:** În prezent economia mondială este caracterizată de turbulențe datorită recente crize economice care a afectat majoritatea țărilor mai puțin zona asiatică (China, India, Japonia). China a fost motorul care a reușit să absoarbă resursele mondiale și să le redistribuie făcând posibilă evitarea unui dezastru economic mondial. Pentru a încerca să se prevadă și să se prevină astfel de efecte la nivel economic mondial în prezent se utilizează diferite softuri bazate pe fuzzy logic, rețele neuronale, inteligență artificială, algoritmi genetici sau mix-uri (Matlab, Visual Gene Developer, JAVA Network Simulator, Genetic Pattern Finder, etc.). Ideea de a utiliza aceste softuri pentru a prognoza evoluții economice a plecat de la utilizarea softurilor pentru a prognoza vremea, unde s-au obținut prognoze ale vremii destul de precise (traietorie, intensitate, timp de manifestare, locul unde se sfârșește).

**CUVINTE CHEIE:** rețele neuronale, predicție, visual gene developer, indicele prețurilor de consum, Indicele prețurilor producției industriale

## 1 INTRODUCERE

Având în vedere evoluția discontinuă a economiei românești, am urmărit să realizez o prognoză a cifrei de afaceri a S.C. Black Sea Suppliers S.R.L. utilizând un soft bazat pe rețele neuronale (Visual Gene Developer) și să obțin o eroare cât mai mică posibil. Obiectivul a fost obținerea aceluiaș rezultat în condiții inițiale identice (trei intrări cu șase componente fiecare, o ieșire cu două componente, rata de învățare și număr de noduri pe strat – constante). Aceste predicții au fost realizate pe bază de iterații succesive până în momentul în care a fost atinsă prognoza cea mai bună (eroarea cea mai mică). De asemenea am urmărit să obțin aceleași rezultate utilizând între unul și trei straturi ascunse.

## 2 STADIUL ACTUAL

Rețelele neuronale artificiale (RNA) sunt recunoscute ca modele dominante (paradigme) ale Inteligenței Artificiale (IA) [Tiponuş & Căleanu, 2002].

Principalele tipuri de rețele neuronale artificiale sunt:

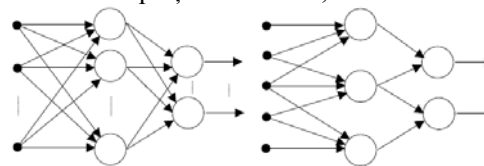
- ↪ RNA de tip perceptron;
- ↪ RNA bazate pe funcții radiale;
- ↪ RNA recurente;
- ↪ RNA cu auto-organizare;

De asemenea sunt utilizate și modalități hibride de implementare a RNA în conjuncție cu alte paradigme ale inteligenței artificiale cum ar fi:

- ❖ Logica fuzzy;
- ❖ Algoritmii genetici;
- ❖ Învățarea prin întărire;

Principalele tipuri de arhitecturi RNA sunt:

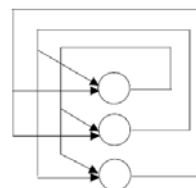
- RNA feedforward (cu propagare înainte) – total sau parțial conectate;



a) total conectată      b) parțial conectată

**Fig. 1 RNA feedforward**

- RNA feedback (recurente), cu propagare înapoi);



**Fig. 2 RNA feedback**

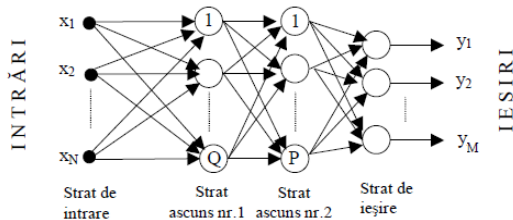
Există o mare diversitate de opinii în ceea ce privește modul de clasificare al algoritmilor și tipurilor de învățare. În Fig.3 se prezintă o sinteză a principalelor direcții.

Perceptronul multistrat (RNA-MLP, Multilayer Perceptron), figura 3, reprezintă o generalizare a perceptronului simplu fiind un RNA de tip feedforward (cu propagare înainte a semnalului) compusă din:

- ✓ un strat de intrare;

Specializarea Concepție și Management în  
Productică, Facultatea IMST; E-mail:  
[daniel.ghencea@blackseasuppliers.ro](mailto:daniel.ghencea@blackseasuppliers.ro)

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



**Fig. 3 Perceptron cu două straturi**

Numărul optim de straturi ascunse și de neuroni/strat ascuns este dificil de precizat apriori [Tiponuş & Căleanu, 2002].

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 [3].

În general, 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 foarte important 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 antrenament. 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 antrenament și implicit la o eroare mare corespunzătoare nu numai datelor de test ci și celor de antrenament. În concluzie, numărul optim de neuroni ascunși se va determina experimental.

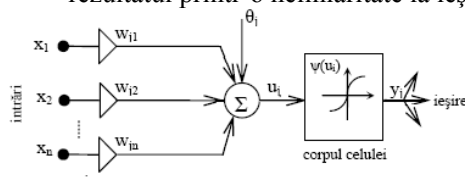
### 2.1 Modele de neuroni artificiali

Sunt cunoscute mai multe modele de neuroni artificiali, precum [2]:

**A. Neuronul static** - este cel mai simplu model de neuron artificial; este alcătuit dintr-o unitate de procesare tipică pentru o rețea neuronală artificială. Fiecare intrare are asociată o pondere a conexiunii, exprimată ca  $w_{ij}$  unde  $i$  este unitatea destinatară, iar  $j$  este unitatea emițătoare. Unitatea de procesare are mai

multe intrări și o singură ieșire care se poate răspândi la multe alte unități din rețea. Semnalul de intrare primit de unitatea  $i$  de la neuronul  $j$  este notat cu  $x_j$ . Această valoare reprezintă ieșirea nodului  $j$ , iar ieșirea unității  $i$  este etichetată  $x_i$ .

**B. Neuronul McCulloch – Pitts (MCP)** - este primul model formal al neuronului artificial care poate fi modelat ca un dispozitiv cu mai multe intrări, neliniar, cu conexiuni ponderate  $w_{ij}$ . Cel mai simplu model al unui neuron artificial sumează cele  $n$  intrări ponderate și trimite rezultatul printr-o neliniaritate la ieșire.



**Fig. 4 Modelul structural al neuronului**

**McCulloch – Pitts**

$$y_i = \Psi \left[ \sum_{j=1}^n w_{ji} \cdot x_j + \theta_j \right] \quad (1)$$

unde:

- ↳  $\Psi$  este funcția de limitare sau prag, numită și funcție de activare sau funcție de funcție de transfer neliniară;
- ↳  $\theta_j$  este un prag extern numit offset sau bias corespunzător unității bias fictive;
- ↳  $w_{ij}$  sunt ponderile sinaptice ale conexiunilor;  $x_i$  sunt intrările unității de procesare;
- ↳  $n$  este numărul de intrări ale unității de procesare;
- ↳  $y_j$  este ieșirea unității de procesare;

Neuronul artificial formal este caracterizat de neliniaritatea sa și de offsetul  $\theta_j$ . 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  $y_{max} > y_{min}$ .

**C. Neuronul dinamic** - este cea mai completă reprezentare a neuronului ca element de bază al unei RNA. El conține trei componente: un sumator ponderat, un sistem liniar dinamic și o funcție neliniară de ieșire. Sistemul liniar dinamic este de tip SISO (Single Input Single Output) și are rolul de a imprima neuronului o anumită comportare dinamică.

**D. 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  $b_{ji}$ .

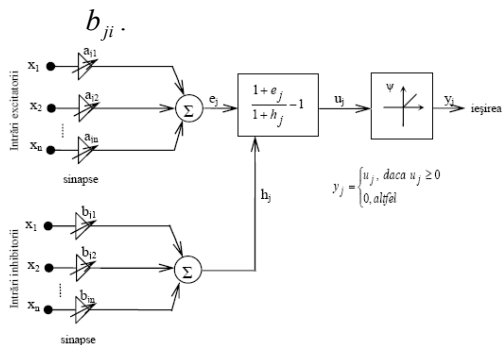


Fig. 5 Modelul structural al neuronului Fukushima

Ieșirea neuronului este descrisă de relația:

$$y_j = \Psi \left( \frac{1 + \sum_{i=1}^n a_{ji} \cdot x_i}{1 + \sum_{i=1}^m b_{ji} \cdot v_i} - 1 \right) \text{ unde } \Psi(u_j) = \begin{cases} u_j, & \text{daca } u_j \geq 0 \\ 0, & \text{altfel} \end{cases} \quad (2)$$

Ponderile sinaptice sunt în general variabile pe parcursul procesului de autoorganizare a unei rețele neuronale.

**E. Neuronul ADALINE (Adaptive Linear Element)** - este folosit ca un dispozitiv logic antrenabil: poate fi antrenat pentru a realiza diferite funcții logice, ajustând în mod adecvat ponderile sinaptice. Un singur dispozitiv ADALINE este capabil să realizeze o gamă redusă de funcții logice - funcții logice liniar separabile (AND, NOT, OR).

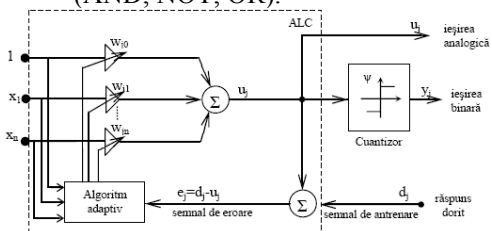


Fig. 6 Modelul structural al neuronului ADALINE

Dispozitivul este compus din două părți: un element de combinare liniar adaptiv, care este blocul fundamental pentru multe sisteme adaptive, conectat în cascadă cu un cuantizor cu două niveluri (funcție de activare prag).

Este esențial faptul că ponderile sinaptice sunt variabile în mod continuu; ele pot lua valori pozitive sau negative, de obicei fiind optimizate astfel încât să minimizeze o funcție de eroare.

## 2.2 Tipuri și algoritmi de instruire

Conform Tîponuț & Căleanu, 2002, avem:

- I. **Învățarea de tip supervizat.** Este caracterizată de prezența unui supervisor care cunoaște cu exactitate modul de asociere al intrărilor RNA cu ieșirile acesteia. Parametrii RNA sunt modificați sub influența combinată a vectorilor de antrenament și a semnalului de eroare (diferența dintre răspunsul dorit și cel actual). Scopul final al algoritmului de antrenament este ca RNA să emuleze, optim în sens statistic, supervisorul.
- II. **Învățarea de tip nesupervizat (cu autoorganizare).** Este caracterizată de absența unui semnal sau supervisor care să aprecieze corectitudinea asociațiilor intrare-ieșire. RNA va descoperii singură legăturile conținute în datele de intrare printr-o reprezentare internă adecvată a trăsăturilor vectorului de intrare.

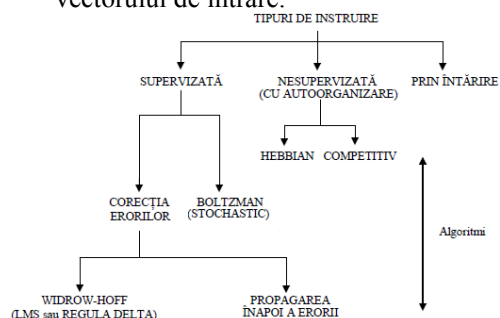


Fig. 7 Tipuri și algoritmi de instruire

- III. **Învățarea prin întărire.** Urmărește maximizarea unei 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.
- 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.** Mașina Boltzmann operează prin alegerea aleatoare a unui neuron și schimbarea stării acestuia. Astfel schimbarea ponderilor se va face ținând cont de corelațiile dintre starea neuronului “i” și cea a neuronului “j”.

- 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 acesteia 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. Fiecare neuron al unei astfel de RNA va deveni specializat, în urma procesului de învățare, în recunoașterea unei anumite **trăsături** prezentă în datele de intrare.

### 2.3 Rețele neuronale recurente

#### 2.3.1 RNA total recurente de tip Hopfield

Rețelele neuronale de tip Hopfield reprezintă rețele recurente, simetrice, total conectate și fără autoasocieri [Constanta Bodea, 2002].

Simetria conexiunilor se exprimă prin egalitatea:

$$w_{ij} = w_{ji} \quad (3)$$

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

Lipsa autoasocierii se exprimă prin:

$$w_{ii} = 0 \quad (4)$$

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.

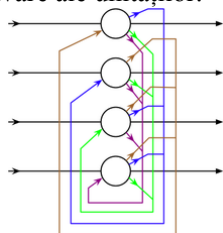


Fig. 8 Rețea Hopfield cu patru noduri

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) și (4). Poate fi văzută ca o **memorie asociativă** sau ca o 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 [Tiponut & Căleanu, 2002].

Fiecare neuron, de tip McCulloch-Pitts, al RNA Hopfield (Fig.8) este caracterizat prin una din cele două stări posibile: activ  $s_i = 1$  și inactiv  $s_i = -1$ , unde starea neuronului este definită de vectorul:

$$s = [s_1, s_2, \dots, s_N]^T \quad (5)$$

iar potențialul intern al neuronului  $j$ :

$$s_j = \begin{cases} +1 & \text{daca } v_j > 0 \\ -1 & \text{daca } v_j < 0 \end{cases} \quad (6)$$

unde  $\theta_j$  reprezintă pragul neuronului.

Neuronul  $j$  își modifică starea conform următoarei reguli:

$$v_j = \sum_{i=1}^n w_{ij} \cdot s_i - \theta_j \quad (7)$$

În funcționarea RNA Hopfield se realizează în două etape:

1. **Faza de memorare** descrisă de ecuația:

$$v_j = \frac{1}{N} \cdot \sum_{\mu=1}^p \xi_{\mu,j} \cdot \xi_{\mu,i} \quad \text{unde } w_{ii} = 0, (\forall) i \quad (8)$$

Din ecuația de calcul pentru ponderile RNA se constată următoarele:

ieșirea fiecărui neuron se constituie în intrare pentru toți ceilalți neuroni ai rețele;

↳ nu există autoexcitație ("self-feedback"), adică  $w_{ii} = 0, (\forall) i$ ;

↳ matricea ponderilor RNA este simetrică (3) adică influența exercitată de neuronul  $i$  asupra neuronului  $j$  este egală cu influența exercitată de neuronul  $j$  asupra neuronului  $i$ .

2. **Faza de utilizare**

Procesul de regăsire se desfășoară în mod dinamic: fiecare neuron al rețelei, în mod aleator ales, estimează propriul potențial de activare și își stabilește starea finală. Acest proces asincron (serial) de modificare a stărilor se va opri în momentul în care vectorul de stare nu se va mai modifica. Aceasta înseamnă că RNA Hopfield a produs un vector de stare  $y$  invariant în timp, ale cărui elemente satisfac condiția de stabilitate:

$$y_j = \text{sgn} \left( \sum_{i=1}^N w_{ij} \cdot y_i - \theta_j \right) \quad \text{unde } j = 1, 2, \dots, N \quad (9)$$

Vectorul de stare  $y$  este denumit stare stabilă a spațiului fazelor sistemului.

Un model selecție-mutație deterministă în abordarea spațiu secvență este studiat în [Tini, 2006] prin identificarea genotipurilor cu secvențe de două litere. Pot apărea praguri de eroare datorită modului de alegere a selecției-mutație. Spre deosebire de alte studii ale modelului Hopfield, lucrarea prezintă sistemul comportamental pragului de eroare nu

pentru toate funcțiile care prezintă finețe în prelucrarea datelor, ci numai pentru anumite valori ale parametrilor. Finețea unei secvențe nu este determinată numai de numărul de mutații comparativ cu o secvență de referință, ci de numărul de secvențe predefinite [Hopfield, 1982].

Cu toate acestea, atunci când prezic molecule native, modelul nu este întotdeauna corect și metoda euristică a rețelei Hopfield nu este întotdeauna stabilă [Quan s.a., 2009], deoarece această clasă de informații este pierdută iar precizia nu este determinată numai de numărul de perechi de baze.

### 2.3.2 RNA parțial recurente de tip Elman

Rețeaua neuronală Elman este una din cele mai simple, putând fi antrenată cu ajutorul algoritmului standard cu propagarea înapoi a erorii [3].

O rețea Elman este constituită din două sau mai multe nivele de unități funcționale în care pe lângă conexiunile specifice rețelelor feed-forward există și conexiuni inverse de la nivelele de unități ascunse către nivelul de intrare [Zaharia, 2009]. Pentru a ține minte ieșirile unităților ascunse Elman introduce *unități de context* care reprezintă un fel de memorie de scurtă durată (short-term memory) [Florea, 2005]. Unitățile contextuale și conexiunile inverse (prin care starea de la momentul anterior a unităților ascunse influențează starea curentă) permit rețelei să detecteze "structura" unor serii temporale. Unitățile ascunse sunt modelat de funcții sigmoide de activare (funcția implicită este tanh) iar unitățile de ieșire au funcții liniare de activare (purelin)

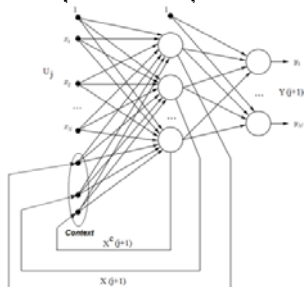


Fig. 9 Rețea neuronală de tip Elman

Într-o rețea Elman neuroni context sunt alimentați de neuroni de ieșire, nu de cei ascunși [Perervenko, 2015]. În plus, nu există nici un feedback direct în neuroni context. Într-o rețea Elman numărul de context și ascunse neuroni trebuie să fie aceeași. Principalul avantaj al rețelelor Elman este faptul că numărul de neuroni de context nu se definește prin numărul de ieșiri ceea ce face mai flexibilă rețea. Neuroni ascunse pot fi ușor adăugate sau luat, spre deosebire de numărul de ieșiri.

Funcționarea rețelei este descrisă de ecuațiile [3]:

$$X(j+1) = \varphi\{W^{xc} \cdot X^c(j+1), W^{xu} \cdot U(j)\} \quad (10)$$

$$X^c(j+1) = X(j) \quad (11)$$

$$Y(j+1) = W^{yx} \cdot X(j+1) \quad (12)$$

unde:

$W^{xc}$  - matricile ponderilor legăturilor dintre stratul de context și stratul ascuns;

$W^{xu}$  - matricile ponderilor legăturilor dintre stratul de intrare și stratul ascuns;

$W^{yx}$  - matricile ponderilor legăturilor dintre stratul ascuns și stratul de ieșire;

$\varphi$  - este o funcție vectorială liniară sau neliniară;

Dacă rețeaua Elman conține doar neuroni cu funcții de activare liniare (10) devine:

$$X(j+1) = W^{xc} \cdot X^c(j+1), W^{xu} \cdot U(j) \quad (13)$$

Conform algoritmului standard de antrenare cu propagare înapoi a erorii, modificarea matricii (vectorului) ponderilor se face după relația:

$$\Delta W = -\eta \cdot \frac{\partial E(j)}{\partial W} \quad (14)$$

Rezultând următorul algoritm de adaptare (în cazul liniar):

$$\begin{cases} \Delta W^{yx} = \eta \cdot (d(j) - y(j)) \cdot X^T(j) \\ \Delta W^{xu} = \eta \cdot (d(j) - y(j)) \cdot X^{yxT} \cdot u(j) \\ \Delta w_m^{xc} = \eta \cdot (d(j) - y(j)) \cdot w_m^{yx} \cdot \frac{\partial x_m(j)}{\partial w_m^{xc}} \end{cases} \quad (15)$$

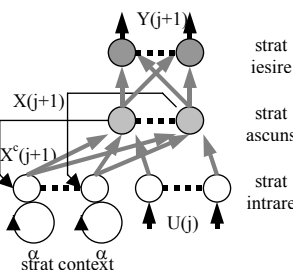


Fig. 10 Rețeaua Elman extinsă

Prin adăugarea la nivelul neuronilor stratului de context, a unei conexiuni de autoreacție, ponderată prin intermediul unei valori fixe, subunitare  $\alpha$  se obține Rețeaua Elman extinsă [Ling Li s.a.], descrisă de ecuațiile:

$$\begin{cases} \Delta W^{yx} = \eta \cdot (d(j) - y(j)) \cdot X^T(j) \\ \Delta W^{xu} = \eta \cdot (d(j) - y(j)) \cdot X^{yxT} \cdot u(j) \\ \Delta w_m^{xc} = \eta \cdot (d(j) - y(j)) \cdot w_m^{yx} \cdot \frac{\partial x_m(j)}{\partial w_m^{xc}} \end{cases} \quad (16)$$

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

În concluzie și rețelele neuronale Elman sunt aproximatori universali, putând fi teoretic antrenate să aproximeze cu o eroare  $\epsilon$  orice funcție continuă, liniară sau neliniară, mono sau multivariabilă.

O analiză a pieței de capital Eurostoxx50 utilizând algoritmi de învățare cu propagare înapoi în rețele neuronale Elman este realizată în [Marques &

Gomes, 2010], unde prin aplicarea aplicarea strategiei stoparea pierderii cu reîntoarcere maximă (stop-loss maximum return – SLMR) s-a realizat o îmbunătățire cu 58% a indicatorului IMA (moving average calculation) adică cu 15 % mai bun decât cel mai bun indicator de deplasare mediu.

### 3 STUDIUL DE CAZ

Având în vedere relația de interdependență care există între fenomenele macroeconomice și microeconomice, în cercetare am luat în considerare și indicatori macroeconomici care au influență asupra unei agenților economici.

Indicatorii luați în considerare pentru perioada 2007-2014 sunt [14, 15]:

- Indicele prețurilor de consum (media pe an)
- Indicele prețurilor producției industriale:
  1. Total industrie (media pe an)
  2. Industria bunurilor de folosință îndelungată (media pe an).

Acești indicatori împreună cu cifra de afaceri realizată de S.C. Black Sea Suppliers S.R.L., pe aceeași perioadă, au fost considerați ca bază de date pentru predicție.

Pentru studiu am utilizat softul Visual Gene Developer 1.7 - VGD (Build 762 – 28 nov 2014, soft care nu necesită licență – freeware) dezvoltat de către *Department of Chemical Engineering and Materials Science – University of California-Davis* și este realizat pe baza unui algoritm standard de învățare cu propagare înapoi.

În tabelul 1 sunt prezentate seriile de numere reale.

**Tabelul 1. Datele de intrare și ieșire**

An	Indicele prețurilor producției industriale		Indicele prețurilor de consum	Cifra de afaceri a S.C. Black Sea Suppliers S.R.L.
	Total industrie	Industria bunurilor de folosință îndelungată		
2007	100,8191667	100,4633333	101,0258333	413693820
2008	101,5291667	101,3766667	101,3316667	455247219
2009	100,3375000	100,4425000	100,3783333	293670549
2010	100,7691667	100,3833333	100,6441667	262008469
2011	100,5408333	100,2616667	100,2583333	289288529
2012	100,3966667	100,2433333	100,4050000	278114728
2013	115,2466667	109,6383333	100,1308333	274130350
2014	117,6191667	111,0566667	100,0691667	280871767

Deoarece VGD lucrează cu serii de numere cuprinse în intervalul închis [-1, 1] acestea trebuie demultiplicate astfel încât să se obțină serii subunitare (ex.: 785,56 trebuie demultiplicat cu 1000 și se obține 0,78556)

**Tabelul 2 Datele de intrare și ieșire demultiplicate**

An	INPUT			OUTPUT
	Total industrie	Industria bunurilor de folosință îndelungată	Indicele prețurilor de consum	Cifra de afaceri a S.C. Black Sea Suppliers S.R.L.
2007	0,0081917	0,0046333	0,0102583	0,4136938
2008	0,0152917	0,0137667	0,0133167	0,4552472
2009	0,0033750	0,0044250	0,0037833	0,2936705
2010	0,0076917	0,0038333	0,0064417	0,2620085
2011	0,0054083	0,0026167	0,0025833	0,2892885
2012	0,0039667	0,0024333	0,0040500	0,2781147
2013	0,1524667	0,0963833	0,0013083	0,2741304
2014	0,1761917	0,1105667	0,0006917	0,2808718

În studiu am luat în considerare următoarele configurații de rețele neuronale (neural network configuration):

- a) Număr de variabile de intrare – 3 (Total industrie, Industria bunurilor de folosință îndelungată, Indicele prețurilor de consum)
- b) Număr de variabile de ieșire – 1 (cifra de afaceri a S.C. Black Sea Suppliers S.R.L.)
- c) Numărul de straturi ascunse – 1, 2 și 3
- d) Numărul de noduri pe strat ascuns – 10
- e) Rata de învățare – 0,01
- f) Funcția de transfer – tangentă hiperbolică
- g) Numărul de cicluri:
  - 1) 639.000 - 3 straturi / 10 noduri;
  - 2) 790.000 - 2 straturi / 10 noduri;
  - 3) 1.535.000 - 1 strat / 10 noduri;
- h) Eroarea țintă – 0,00001

După stabilirea configurației rețelei neuronale, au fost încărcate setul de date de antrenare (Fig.11 a) și setul de date pentru care se va face predicția (Fig.11 b).

	In 1	In 2	In 3	Out 1	Pred. Out 1
1	0.0081917	0.0046333	0.0102583	0.4136938	
2	0.0152917	0.0137667	0.0133167	0.4552472	
3	0.0033750	0.0044250	0.0037833	0.2936705	
4	0.0076917	0.0038333	0.0064417	0.2620085	
5	0.0054083	0.0026167	0.0025833	0.2892885	
* 6	0.0039667	0.0024333	0.0040500	0.2781147	

a) setul de date de antrenare

	In 1	In 2	In 3	Pred. Out 1
1	0.1524667	0.0963833	0.0013083	
* 2	0.1761917	0.1105667	0.0006917	

b) setul de date pentru predicția

**Fig. 11 Introducerea datelor experimentale**

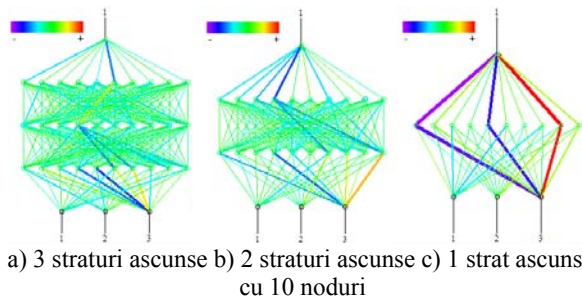
În urma rulării s-au obținut datele prezentate în tabelul următor.

**Tabelul 3 Predicția**

Real	Predicție					
	3 straturi ascunse	%	2 straturi ascunse	%	1 strat ascuns	%
0.27413035	0.2800955	102.176%	0.2819980	102.870%	0.2815778	102.717%
0.280871767	0.2747167	97.809%	0.2737176	97.453%	0.2740587	97.574%
0.555002117	0.5548122	99.966%	0.5557156	100.129%	0.5556365	100.114%
iterații	639000		790000		1535000	
Time predicție		12 min 49 sec		4 min 1 sec		28 min 4 sec

Se observă că predicția cu 3 straturi ascunse are cea mai mică eroare totală (0,034%) și ciclul cel mai mic de iterații. Timpul de predicție este influențat de granulația de iterații. Astfel cu cât se dorește un număr de iterații mai precis pentru a avea eroare cât mai mică, timpul de învățare crește. Se observă că la experimentul cu 3 și 1 straturi ascunse pentru iterație au fost luate în considerare miile, pe când la 2 straturi ascunse zecile de mii.

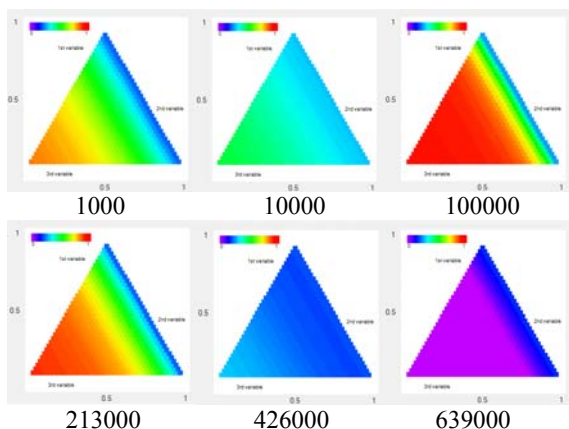
Modul de învățare precum și traseele utilizate în procesul de predicție sunt prezentate în figura 12.



**Fig. 12 Fluxul de informații între straturile și nodurile ascunse**

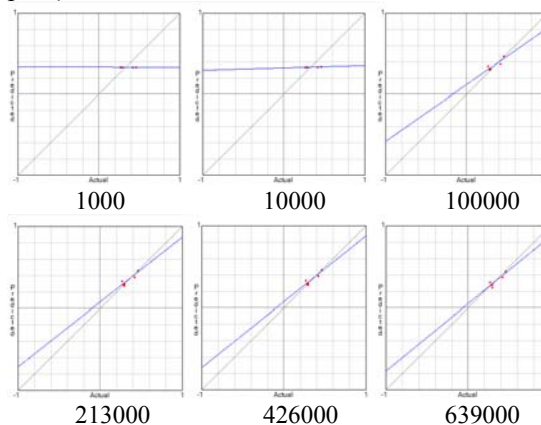
În diagrame, culoarea roșie corespunde numerelor pozitive mari (care tind către +1) iar violet numerelor negative mari (care tind către -1). Lățimea liniei este proporțională cu factorul de greutate în valoare absolută sau valoarea de prag.

Harta de analiză a predicției a fost setată pentru 3 intrări, hartă care analizează influența reciprocă datelor de intrare. În figura 13 sunt prezentate evoluția influențelor reciproce ale intrărilor sub forma de triunghi. În prima treime a iterațiilor (0-231000) se observă tatonarea întregii suprafețe a triunghiului pe intervalul [0, 1] pentru stabilirea și învățarea algoritmului de predicție. În ultimele două treimi ale iterațiilor (231000-693000) se observă că rețeaua neuronală a învățat modul de construcție al algoritmului iar variante de creștere a performanței nu mai sunt disponibile.



**Fig. 13 Evoluția influențelor reciproce ale intrărilor**

Evoluția coeficientului de regresie  $r^2$  și a pantei sunt prezentate în figura 14. Se observă că după primele 10.000 de iterații coeficientii de regresie au ajuns în apropierea pozițiilor finale iar panta se apropie după 100.000 de iterații și ea de această poziție.

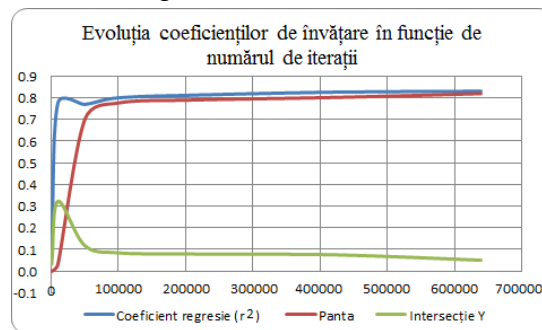


**Fig. 14 Evoluția coeficientului de regresie și a pantei**

**Tabelul 4 valorile coeficienților la diferite iterații**

Iteratia	Coeficient regresie ( $r^2$ )	Panta	Intersecție Y
1000	0.1173852	-0.0010200	0.03323326
10000	0.7744763	0.0292900	0.32218950
50000	0.7702196	0.6994250	0.11836070
100000	0.8006868	0.7767784	0.08454540
213000	0.8129984	0.7906055	0.07917986
426000	0.8270035	0.8024670	0.07527208
639000	0.8310586	0.8197635	0.05050075

Grafic evoluția celor trei indicatori este prezentată în figura 15.



**Fig. 15 Reprezentarea grafică a celor trei indicatori**

Graficul din figura 15 este caracterizat de două zone: zona de căutare (ripuri moderate) și zona de stabilitate cu evoluție către atenuare și liniaritate. Se observă că evoluția pantei și a coeficientul de regresie se urmăresc reciproc iar coeficientul de intersecție al axei Y tinde către punctul 0 (origine),

moment în care predicția este foarte precisă având o eroare de 0,00001.

#### 4 CONCLUZII

Astfel de modele oferă **avantaje** datorită flexibilității conexiunilor între straturile escuse și noduri a capacității lor de a copia modul de funcționare al neuronilor deci a se comporta ca un întreg.

Se observă că fiecare componentă are **limitări** în prelucrarea informației de intrare, ascuse și de ieșire.

Rezultatele studiului demonstrează că se pot obține rezultate cu eroare foarte mică dacă se iau în considerare seturi de date care să caracterizeze cât mai bine fenomenul analizat în cazul în care este numai o singură ieșire.

În cazul în care există mai multe ieșiri (4-6), rezultate experimentale actuale care sunt în fază incipientă (nefinalizate), prezintă erori mari datorită complexității fenomenului. De menționat este faptul că graficul test încearcă să copieze graficul original cu un anumit grad de multiplicare/demultiplicare a setului de date.

Analiza seturilor de date de tip Multi-Input–Multi-Output (MIMO) cu ajutorul rețelelor neuronale fac obiectul cercetării viitoare. Ținta finală o reprezintă o analiză hibridă rețele neuronale - fuzzy logic.

#### 5 BIBLIOGRAFIE

- [1] Tiponut Virgil, Căleanu Cătălin-Daniel, *Rețelele neuronale - Arhitecturi și algoritmi*, Ed. Politehnică Timișoara, 2002, ISBN 973-9389-66-X, disponibil la: <http://ro.scribd.com/doc/227260899/retele-neurale#scribd>, Accesat la data: 01.05.2015
- [2] *Rețele neuronale artificiale*, disponibil la: <https://xa.yimg.com/kq/groups/23876473/1339028079/name/Curs+10-11.pdf>.
- [3] *Rețele neuronale. Modelarea și simularea sistemelor dinamice*, disponibil la: [www.infoap.utcluj.ro/SO/cap2-f.doc](http://www.infoap.utcluj.ro/SO/cap2-f.doc), Accesat la data: 01.05.2015
- [4] Constanta Bodea, *Optimizarea cu ajutorul rețelelor neuronale recurente de tip Hopfield*, Revista Informatică Economică, nr. 1 (21)/2002, disponibil la: <http://revistaie.ase.ro/content/21/bodea.pdf>, Accesat la data: 01.05.2015
- [5] Hopfield network, Wikipedia, the free encyclopedia, disponibil la: [http://en.wikipedia.org/wiki/Hopfield\\_network](http://en.wikipedia.org/wiki/Hopfield_network), Accesat la data: 01.05.2015

- [6] Tini Garske, *Error thresholds in a mutation-selection model with Hopfield-type fitness*, disponibil la: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.243.647>, Accesat la data: 01.05.2015
- [7] Hopfield Jhon.J, *Neural networks and physical systems with emergent collective computational properties*, Proc. Natl. Acad. Sci. USA, vol.79, Biophysics,1982, disponibil la: <http://www.pnas.org/content/79/8/2554.full.pdf+html>, Accesat la data: 02.05.2015
- [8] Quan Zou, Tuo Zhao, Yang Liu, Maozu Guo, *Predicting RNA secondary structure based on the class information and Hopfield network*, Journal Computers in Biology and Medicine, Volume 39 Issue 3, March, 2009, disponibil la: <http://dl.acm.org/citation.cfm?id=1513330> Accesat la data: 02.05.2015
- [9] Zaharia D., *Implementarea memoriilor asociative cu rețele de tip Hopfield. Prelucrarea seriilor temporale cu rețele de tip Elman*, disponibil la: [http://web.info.uvt.ro/~dzaharie/rn2009\\_lab7.pdf](http://web.info.uvt.ro/~dzaharie/rn2009_lab7.pdf), Accesat la data: 02.05.2015
- [10] Adina Magda Florea, *Sisteme de programe pentru timp real*, Universitatea "Politehnică" din București, 2005, disponibil la: [http://turing.cs.pub.ro/sptr\\_05/SPTR\\_Lect\\_6.ppt](http://turing.cs.pub.ro/sptr_05/SPTR_Lect_6.ppt), Accesat la data: 03.05.2015
- [11] Vladimir Perervenko, *Third Generation Neural Networks: Deep Networks*, 2015, disponibil la: [https://www.mql5.com/en/articles/1103#1\\_2\\_3](https://www.mql5.com/en/articles/1103#1_2_3), Accesat la data: 03.05.2015
- [12] Ling Li, Zhidong Deng, Bo Zhang, *A Fuzzy Elman Neural Network*, disponibil la: <http://www.work.caltech.edu/~ling/pub/97fenn.pdf>, Accesat la data: 03.05.2015
- [13] Nuno C. Marques, Carlos Gomes, *Maximus-AI: Using Elman Neural Networks for Implementing a SLMR Trading Strategy*, Lecture Notes in Computer Science, Volume 6291, 2010, pp 579-584, disponibil la: [http://link.springer.com/chapter/10.1007%2F978-3-642-15280-1\\_55](http://link.springer.com/chapter/10.1007%2F978-3-642-15280-1_55) Accesat la data: 03.05.2015
- [14] disponibil la: <http://www.insse.ro/cms/ro/content/indicele-preturilor-de-consum>, Accesat la data: 01.05.2015
- [15] disponibil la: <http://www.insse.ro/cms/ro/content/indicele-preturilor-productiei-industriale>, Accesat la data: 01.05.2015
- [16] disponibil la: <http://www.visualgenedeveloper.net/>, Accesat la data: 01.05.2015
- [17] disponibil la: <http://doingbusiness.ro/financiar/raport/321094/blac-k-sea-suppliers-srl/>, Accesat la data: 01.05.2015