

ALGORITM EVOLUTIV DE IDENTIFICARE NEURALĂ

Mihai Tertişco

tertisco_mihai@yahoo.com

George Condur

g_condur@yahoo.com

Universitatea Politehnica Bucureşti

Rezumat: Se prezintă un nou algoritm de tip evolutiv pentru identificarea neurală a unui proces stocastic. Algoritmul sugerează procesul evoluţiei biologice a unor modele parţiale simple (neuroni artificiali) care formează populaţia din care prin încrucişare apar noi generaţii de neuroni artificiali, destinaţi straturilor ascunse a unei reţele neurale. Prin acest algoritm, se urmăreşte obţinerea unei structuri optimale a unei reţele neurale **feedforward** în raport cu structura obţinută prin aplicarea teoremei Kolmogorov de aproximare a unei funcţii. Este prezentat studiul de caz privind modelarea neurală a unui proces stocastic neliniar. Este propus un algoritm hibrid, compus din combinarea algoritmului backpropagation cu algoritmul evolutiv din aceasta lucrare.

Cuvinte cheie: reţele neurale, model neural, feedforward, algoritm evolutiv, backpropagation.

Abstract: The paper presented an evolutionary algorithm guy concerning the neural model structure projection. Trough this algorithm calculated of the optimal nets structures in report with structure obtained trough the adhibition of the Kolmogorov theorem. Is proposed an empirical criterion of flapped the experimental data in 2 sets: **set of neural net training** and **set for verification of the neural model**. Is present the examples and study of case. Is propose for training a hybrid algorithm of training result through the combinations of the **backpropagation** algorithm with the evolucional suggested algorithm of the author work.

Keywords: Neural nets, neural models, feedforward, evolutionary algorithm, backpropagation.

1. Elemente introductive în calculul evolutiv

Calculul evolutiv este folosit în principal în rezolvarea problemelor bazate pe căutarea soluţiei într-un spaţiu de soluţii potenţiale. Sursa de inspiraţie o reprezintă principiile evoluţionismului darwinist. Principiul fundamental este de a dezvolta sisteme complexe pornind de la reguli simple [1]. Majoritatea algoritmilor evolutivi sunt inspiraţi din comportamentul insectelor în cadrul unor colectivităţi de albine, termite etc., din activitatea socială umană sau din experienţa milenară a oamenilor, acumulată în rezolvarea problemelor prin căutarea soluţiei acceptabile într-un spaţiu al soluţiilor plauzibile [1]. Ilustrăm aceasta prin următorul exemplu sugestiv. Pentru obţinerea „lalelei negre” s-a folosit o operaţie de „încrucişare biologică” prin cultivarea unei suprafeţe cu **lalele albe** şi **lalele roşii** distribuite aleator pe suprafaţa cultivată. Răsadurile cultivate reprezintă **părinţii** lalelelor din prima recoltă (prima generaţie de lalele) culeasă de pe suprafaţa dată. La prima generaţie de lalele recoltate, se constată că, printre lalelele recoltate, în afara celor albe şi roşii, apar unele exemplare care au suferit unele „**mutaţii genetice**” exprimate prin aceea că nu sunt nici albe, nici roşii, ci mai degrabă „**gri**” „**gri mai închis**” sau „**gri deschis**” etc. Apoi, se aplică un procedeu de **selecţie** a exemplarelor care sunt cele mai apropiate de scopul căutării (obţinerea lalelei negre). Populaţia lalelelor selectate din prima recoltă sunt „**copiii**” obţinuţi din încrucişarea de lalele albe şi roşii, dar ei vor forma **mulţimea părinţilor** pentru o nouă recoltă de lalele şi mai apropiate „laleaua neagră”. Se constată aici un **proces evolutiv treptat** spre o **soluţie acceptabilă a problemei** formulate. Cei ce se ocupă cu astfel de procese de ameliorare a unor specii prin încrucişări şi selecţii repetate au înţeles mai demult că, din fiecare generaţie, se alege nu numai o singură pereche de reprezentanţi pentru o nouă încrucişare, ci se alege o mulţime de exemplare care satisfac în cea mai mare măsură scopul urmărit, iar încrucişarea se face aleatoriu. Experienţa în acest domeniu a arătat că această mulţime selectată pentru o nouă încrucişare nu trebuie să fie foarte diversă, alegerea deci nefiind foarte liberă. O mare libertate de alegere poate dăuna. Există un optimum din punctul de vedere al „libertăţii de alegere” pentru fiecare din generaţii [2].

2. Structura algoritmului evolutiv de identificare neurală

Algoritmul evolutiv, propus în lucrare, este un *algoritm euristic de căutare*, folosit în rezolvarea problemei de optimizare a structurii unei reţele neurale, care modelează un proces stocastic. Datele problemei sunt compuse din valorile unui semnal aleator, la momente succesive de timp. Problema constă în căutarea structurii plauzibile a unei reţele neurale, capabile să recunoască nu numai aceste valori date ale semnalului, ci şi valori viitoare, care nu au făcut parte din mulţimea de date ale problemei. Obţinerea unei structuri acceptabile a reţelei se face prin selecţii succesive, din rândul unor modele parţiale simple, şi prin combinarea şi variaţia parametrilor acestora folosind mecanisme care sugerează procesul evoluţiei biologice, descris mai sus.

1. **GENERAREA POPULAȚIEI ÎNȚIALE**
2. **EVALUAREA REPREZENTANȚILOR POPULAȚIEI PRIN CALCULUL VALORII FUNCȚIEI DE ADAPTABILITATE ASOCIATE REPREZENTANȚILOR**
 - (început de ciclu)
 1. **SELECȚIA INDIVIZILOR DIN GENERAȚIA CURENTĂ**
 2. **INCRUCIȘAREA și \sau MUTAȚIA**
 3. **CALCULUL FUNCȚIEI DE ADAPTABILITATE PENTRU TOȚI REPREZENTANȚII GENERAȚIEI**
 4. **FORMAREA POPULAȚIEI NOII GENERAȚII**
 - **DACA SE INDEPLINEȘTE CONDIȚIA DE STOP ATUNCI (sfârșit de ciclu), ALTFEL (început de ciclu)**

Figura 1. Structura algoritmului evolutiv pentru determinarea modelului neural adecvat

Particularitatea algoritmului evolutiv (EA) este folosirea operației de *recombinare* a modelelor plauzibile parțiale (din rândul celor mai acceptabile pentru scopul urmărit de căutare). Rolul operației de recombinare este similar cu cel al operației de „încrucișare” din sistemele biologice naturale. Acest algoritm este un model informatic, care emulează modelul biologic evoluționist. Se generează o mulțime aleatoare de modele parțiale, numită *populație* inițială. Aceste modele parțiale sunt evaluate folosind o *funcție de adaptabilitate* numită *fitness* și se asociază fiecărui model parțial o anumită valoare a gradului de adaptabilitate care exprimă probabilitatea de supraviețuire. Pe baza acestor valori ale adaptabilității, se execută *selecția* mutanților (modele parțiale rezultate din încrucișare), care sunt admiși pentru o nouă *încrucișare* restul modelelor parțiale sunt considerate exemplare „inadaptabile” care pier. Reprezentanții noii generații sunt, de asemenea evaluați după care se aplică *selecția* și se aplică operatorii de evaluare etc. Astfel, este modelat procesul evolutiv care continuă câteva cicluri de viață (generații), până se îndeplinește condiția de oprire a algoritmului. O astfel de *condiție de stop* poate fi:

- găsirea soluției globale ori a unei soluții suboptimale;
- parcurgerea unui număr prescris de generații destinat evoluției;
- terminarea duratei timpului prescris pentru evoluție.

Structura algoritmului propus pentru identificare neurală este prezentată în figura 1 și din câte se observă este un algoritm ciclic. Fiecare ciclu al algoritmului evolutiv are ca rezultat o nouă generație de modele parțiale, aferente unui nou strat de neuroni ai rețelei care modelează procesul real.

3. Caracteristicile unei rețele neurale artificiale(RN)

O rețea neurală artificială este alcătuită dintr-o mulțime de noduri în care se afla neuronii artificiali, elemente de procesare neliniare, care operează în paralel și care au ca punct principal de inspirație sistemul nervos. Prin analogie cu neuronul biologic, un neuron artificial are un număr de intrări și o singură ieșire, care se poate conecta la intrarea mai multor neuroni. Fiecare intrare are asociată o anumită pondere care reprezintă importanța pe care o are impulsul prezent pe linia respectivă, la activarea neuronului. Cu toate că asemănarea între sistemul nervos și RN este relativ mică, RN prezintă un număr surprinzător de mare de caracteristici ale creierului. De exemplu, RN pot învăța din experiență, pot generaliza și, din anumite exemple, pot recunoaște altele noi și pot sintetiza caracteristicile esențiale din intrări ce conțin și date irelevante. Un mare avantaj al RN este că pot descrie și rezolva o problemă parcurgând, în prealabil, un proces de învățare (de antrenament al rețelei). Caracteristicile cele mai importante ale rețelelor neurale sunt: *capacitatea de a învăța* (prin antrenarea RNA); *capacitatea de generalizare* (dacă au fost antrenate corespunzător, rețelele sunt capabile să dea răspunsuri corecte și pentru intrări diferite față de cele cu care au fost antrenate); *capacitatea de sinteză* atunci când sunt confruntate cu informații complexe sau cu zgomote irelevante. În rezolvarea unei probleme prin intermediul rețelelor neurale, se identifică trei etape:

1. obținerea unui set de date de antrenament și a unui set de test cât mai semnificative pentru problema respectivă;

2. stabilirea arhitecturii rețelei (număr de intrări și ieșiri, număr de straturi și de neuroni în fiecare strat);
3. antrenarea și testarea rețelei pe baza datelor conținute în setul de antrenament și cel de test.

O rețea neurală artificială este reprezentată printr-o mulțime de noduri care modelează neuroni artificiali. Aceste noduri sunt conectate printr-o mulțime de ponderi de conexiune sau ponderi sinaptice [3].

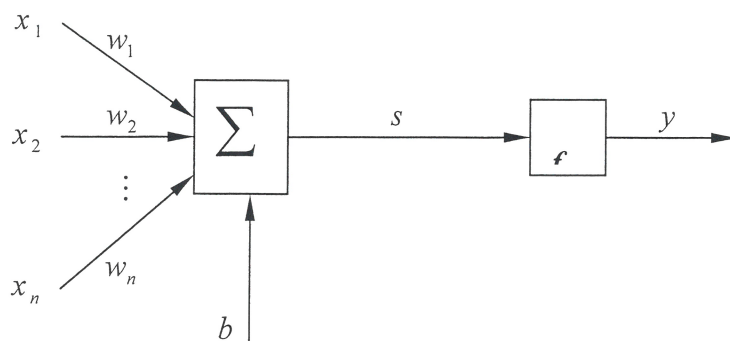


Figura 2. Modelul neuronal McCulloch-Pitts

Fiecare neuron i are n_i **intrări** care reprezintă fie semnale venite de la alți neuroni cu care este conectat, fie semnale din mediul exterior, mai conține n_i **ponderi sinaptice** (câte una pentru fiecare intrare a sa) și o **ieșire** y_i . Ponderile se notează cu $w_1^i, w_2^i, \dots, w_{n_i}^i$ și reprezintă numere reale care ponderează semnalul de intrare corespunzător. Dacă $w_j^i > 0$ avem o pondere sinaptică **excitatoare**, iar dacă $w_j^i < 0$ avem de a face cu o **pondere inhibitoare**.

Primul model de neuron artificial a fost propus de McCulloch și Pitts în 1943, fiind compus dintr-un set de intrări și o ieșire, ca în figura 2. Fiecare intrare $x_1, x_2, \dots, x_n$ a unui neuron este ponderată, adică valoarea sa este multiplicată cu o valoare corespunzătoare $w_1, w_2, \dots, w_n$, apoi toate intrările ponderate sunt însumate. Acestei sume s se adaugă un parametru b numit **bias** (deplasare de scală), rezultând ceea ce se numește activarea neuronului, notată cu s . Valorii de activare a neuronului, astfel obținute s se aplică o funcție de activare f , rezultând valoarea y a ieșirii neuronului respectiv. Parametrul b are rolul de a deplasa originea funcției de activare producând un efect similar cu modificarea pragului de activare al neuronilor și permițând, în același timp, o convergență mai rapidă în procesul de antrenament. **Funcția prag** asociată modelului McCulloch-Pitts este $f: \mathbb{R} \rightarrow \{0, 1\}$ având forma:

$$f(x) = \begin{cases} 1, & \text{daca } x \geq 0 \\ 0, & \text{daca } x < 0 \end{cases}$$

Modelul matematic al neuronului prezentat în figura 1 este descris de relația:

$$y = f\left(\sum_{j=1}^n w_j x_j + b\right) \quad (1)$$

Topologia rețelelor neurale exprimă modul în care neuroni sunt conectați în cadrul rețelei, formând căi de legătură prin intermediul cărora ieșirea unui neuron este transmisă spre un altul din stratul următor, ca intrare a acestuia. Aceste căi de legătură sunt, de obicei, unidirecționale, dar sunt modele de rețele neurale în care pot exista legături bidirecționale între doi neuroni din straturi succesive. Un neuron recepționează la intrările sale semnale provenite de la mai mulți neuroni cu care este conectat, dar produce o singură ieșire care va fi transmisă mai departe la intrările altor neuroni din rețea. O rețea neurală este un complex de neuroni artificiali, organizați pe niveluri (**straturi**), nivelurile situate între stratul de intrare și cel de ieșire alcătuind așa numitele **niveluri ascunse**. Primul strat primește intrările din mediul exterior. Ieșirile neuronilor dintr-un strat constituie intrări pentru neuroni din stratul următor. Ieșirea rețelei este formată din ieșirile neuronilor ultimului strat. Acest tip de rețea la care semnalele sunt transmise numai de la intrare spre ieșire se numesc **rețele feedforward**. Ele nu sunt recurente și nu prezintă feedback, deci nu au memorie. **Ieșirea lor este determinată numai de valorile curente ale**

intrării și de cele ale ponderilor. Ele se pretează pentru probleme care pot fi modelate prin funcții liniar separabile, devenind ineficiente pentru probleme care nu sunt liniar separabile. *Pentru acest tip de probleme se pot folosi rețele feedforward cu mai multe niveluri.*

5. Etapele standard în proiectarea RN de modelare a proceselor

Proiectarea RN pentru modelare implică parcurgerea etapelor standard, prezentate în figura 3. Acestea sunt valabile în cazul instruirii cu instructor (instruire supervizată) a rețelei neurale. Prezenta lucrare tratează o procedură nouă de proiectare, bazată pe algoritmi evolutivi de căutare a unei structurări optime a modelului neural pentru prognoză și identificare, în cazul proceselor stocastice MIMO (multi-input-multi-output). În această nouă procedură, se pune accentul mai mult pe etapele 4, 5 și 6 din structura algoritmului, prezentată în figura 3, presupunând, în general, parcurse primele trei etape.

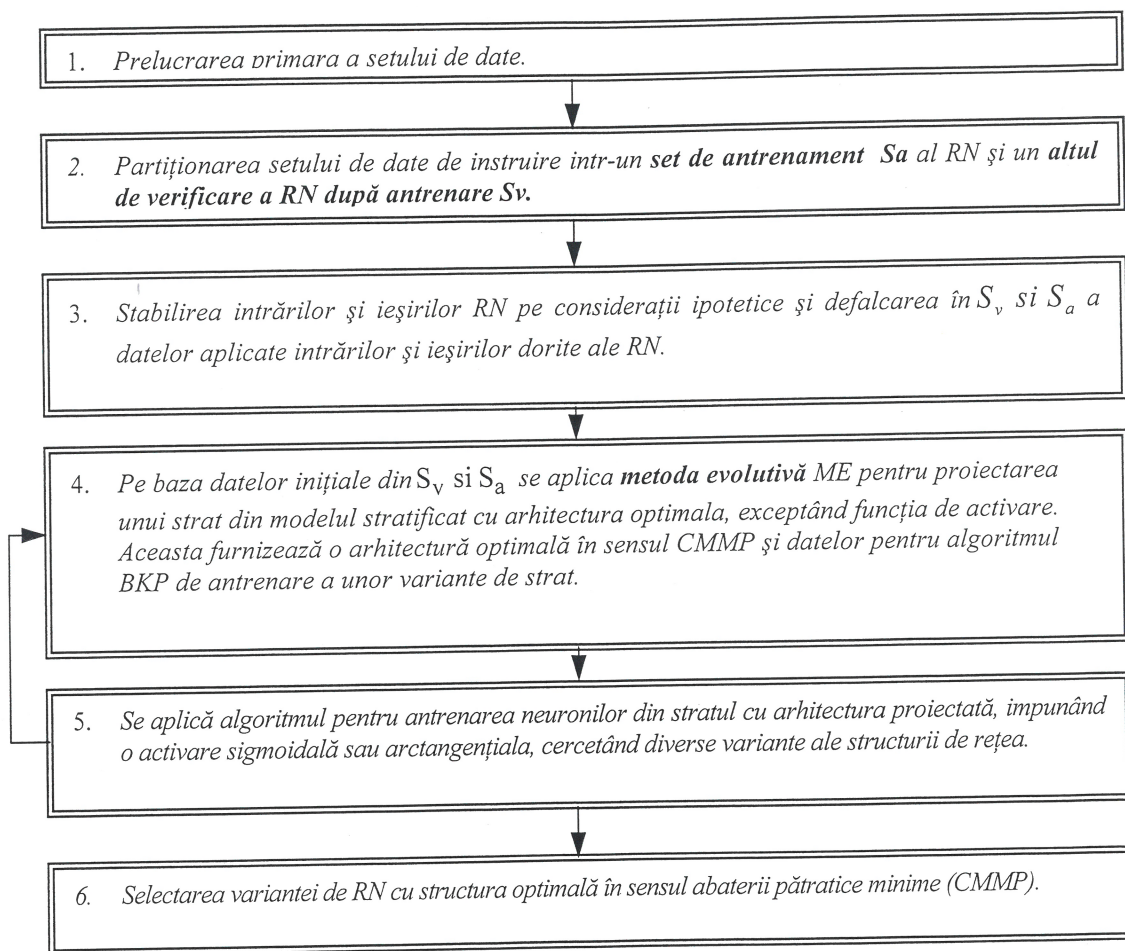


Figura 3. Structura algoritmului evolutiv de proiectarea RN

Metoda se bazează pe faptul că, pentru obținerea modelului unui proces cu o ieșire y și una sau mai multe mărimi de intrare $\{U_1, U_2, \dots, U_n\}$. Structura modelului poate fi liniară sau neliniară și poate fi exprimată neural. Există disponibile $n+1$ șiruri de câte N valori succesive ale mărimilor de intrare U și de ieșire Y : unde matricea U de date de intrare măsurate fără erori, iar vectorul Y conține un șir de N valori ale ieșirii perturbate aditiv de un zgomot aleator, în sensul că tendința o dată extrasă, rămâne numai componenta aleatoare staționară a semnalului de ieșire. Șirul de valori date urmează să asigure determinarea modelului neural, cu ajutorul căruia să se poată prezice valoarea procesului aleator y la momentul imediat următor $t = N + 1$.

Se consideră că setul de date $y_1 \dots y_N$ provine dintr-un proces aleator neliniar, descris de ecuația neliniară:

$$y_t = \underbrace{F(y_{t-1}, y_{t-2}, \dots, y_{t-n})}_{\text{parte prognozabilă}} + \underbrace{e_t}_{\text{parte neprognozabilă}} \quad (2)$$

în care e_t este zgomot alb care este nepredictabil, iar n este ordinul autoregresiei neliniare, care este necunoscut.

6. Ipoteze de lucru pentru evaluarea ordinului modelului autoregresiv și structurii modelului de prognoză

Ipoteza 1

Se consideră că $n = 1$. Aceasta înseamnă că valoarea la momentul $t = N + 1$ ar putea fi prognozată cu modelul de prognoză:

$$\hat{y}_{N+1} = F(y_N) \quad (3)$$

Banalizând problema, am putea considera că prognoza $\hat{y}_{N+1}$ pentru momentul $N + 1$ este exprimată prin valoarea lui y în anul precedent ($\hat{y}_{N+1} = y_N$). Este evident că prognoza va fi puternic eronată din două motive: nu se cunoaște F , care s-a presupus $\hat{F}(y) = y$; nu se cunoaște n , care s-a presupus $n = 1$.

Ipoteza 2

Ar mai fi o soluție, să se presupună $n = 1$, iar F o funcție liniară, cu parametri necunoscuți:

$$\hat{y}_{N+1} = \hat{F} = a + b \cdot y_N \quad (4)$$

în care $y_N = F(U_1(N), \dots, U_n(N))$ iar $U_i(N) = y_{N-i}, i = 1, 2, 3, \dots, n$ sunt intrări ale rețelei neurale RN. În acest caz, se cere să se precizeze: care sunt valorile parametrilor a și b , care asigura o „bună” prognoză. Această precizare se poate face dacă stabilim un criteriu de evaluare cantitativă a preciziei de prognoză. O modalitate ar fi să calculăm eroarea de prognoză, folosind măcar o parte din datele $y_1 \dots y_N$, de care dispunem. Intuiția ne conduce la a evalua eroarea de prognoză apelând la **criteriul celor mai mici pătrate (CMMP)**. Erorile pot avea valori nule, negative sau pozitive și în ansamblul lor eroarea putând să fie nulă, cu toate că modulul erorilor punctiforme ($|\varepsilon_k| \gg 0$) este foarte mare, deci suntem obligați să apelăm la criteriul **CMMP** pentru a estima niște valori pentru a și b pe care le numim *estimații în sensul CMMP* [4]. Dezvoltând funcția F din (2) în serie Taylor și trunchiind seria la termenii pătratici, modelul de prognoza al procesului aleator (2) la momentul $N+1$, pe baza ultimelor două valori măsurate la momentele N și $N-1$ ar putea fi:

$$y_{N+1} = a_0 + a_1 y_N + a_2 y_{N-1} + a_{12} y_N y_{N-1} + a_{11} y_N^2 + a_{22} y_{N-1}^2 \quad (5)$$

În cazul în care predicția y_{N+1} depinde de mai multe valori întârziate y_{N-1}, y_{N-2} , etc., modelul devine deosebit de laborios și poate fi obținut numai prin metode evolutive de proiectare, care se bazează pe **asamblarea unor modele parțiale simple**. Asemenea model regresional neliniar de prognoză poate fi implementat pe rețele neurale multistrat (cu mai multe niveluri ascunse). Metoda evolutivă (ME) este utilă pentru cazurile în care secvența de date disponibile aparține unui proces stocastic neliniar, de forma:

$$y_t = F(y_{t-1}, y_{t-2}, \dots, y_{t-n}) + e_t$$

pentru care ordinul modelului autoregresiv al acestuia este de ordin $n > 3$.

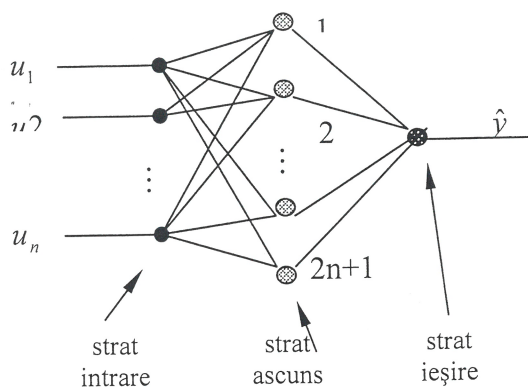


Figura 4. Structura RN proiectată în conformitate cu teorema Kolmogorov

În acest caz, conform teoremei Kolmogorov, rețeaua cu arhitectura maximală neoptimală conține: **un strat inactiv cu n noduri de intrare, un strat ascuns cu $2n+1$ neuroni activi în nodurile total interconectate cu cele de intrare și un strat de ieșire cu un singur neuron conectat la ieșirile tuturor neuronilor din stratul ascuns, ieșirea acestui ultim neuron este ieșirea y a rețelei**, ca în figura 4. Matricea C a conexiunilor dintre ieșirile nodurilor din stratul ascuns are dimensiunea $n \times (2n+1)$ și conține elemente cu valoare 1, (atunci când există conexiune între intrarea corespunzătoare liniei și neuronul din stratul ascuns corespunzător coloanei). Numărul de astfel de conexiuni exprimă complexitatea rețelei. O rețea total conectată are toate elementele matricei de conexiuni C egale cu 1 ca în figura 4. În cazurile când $n > 3$, numărul de neuroni din stratul ascuns, precum și numărul de conexiuni este prohibitiv. Teorema Kolmogorov fiind o teoremă **numai de suficiență**, este posibilă găsirea unei rețele neurale cu structura optimă, cu un număr mai mic de conexiuni și de neuroni în stratul ascuns, care să asigure o eroare de aproximare suficient de mică chiar **mult mai mică** decât structura obținută prin aplicarea teoremei Kolmogorov. În continuare, se prezintă un modul în care se construiește primul strat ascuns (în cazul unei RN cu $n = 4$ intrări posibile) folosind modele neurale parțiale, care se antrenează pe baza a N perechi de date experimentale intrare-ieșire, rezultate din măsurători succesive la intervale constante de timp. Din aceste N date de instruire $N_a < N$ perechi sunt destinate antrenării RN, iar restul de N_v perechi de date sunt folosite pentru verificare. Numărul total de date din setul de instruire este $N = N_a + N_v$. **Stratul ascuns se construiește numai din neuroni cu două intrări din cele patru ale rețelei**, fiecare neuron fiind un posibil model neuronal parțial de prognoză a ieșirii rețelei, construit pe baza datelor din setul de instruire. Pot fi construite (pentru cazul din exemplul în care $n = 4$) în total $C_4^2 = \frac{4!}{2! \cdot 2!} = \frac{12}{2} = 6$ modele parțiale de prognoză pentru

procesul stocastic $y_{N+1} = F(u_1, u_2, \dots) + e_t$ în care e_t este zgomot alb, iar u_1 și u_2 pot fi perechi din $\{y_N, y_{N-1}, y_{N-2}, y_{N-3}\}$. Cele 6 astfel de modele posibile, formate prin încrucișarea perechilor, sunt următoarele:

- neuronul (1) $\hat{y}_{N+1} = f(w_1(1) \cdot y_N + w_2(1) \cdot y_{N-1} + b(1))$
- neuronul (2) $\hat{y}_{N+1} = f(w_1(2) \cdot y_N + w_2(2) \cdot y_{N-2} + b(2))$
- neuronul (3) $\hat{y}_{N+1} = f(w_1(3) \cdot y_N + w_2(3) \cdot y_{N-3} + b(3))$
- neuronul (4) $\hat{y}_{N+1} = f(w_1(4) \cdot y_{N-1} + w_2(4) \cdot y_{N-2} + b(4))$
- neuronul (5) $\hat{y}_{N+1} = f(w_1(5) \cdot y_{N-1} + w_2(5) \cdot y_{N-3} + b(5))$
- neuronul (6) $\hat{y}_{N+1} = f(w_1(6) \cdot y_{N-2} + w_2(6) \cdot y_{N-3} + b(6))$

Acestor modele parțiale le corespund structurile neuronale, prezentate grafic în figura 5.

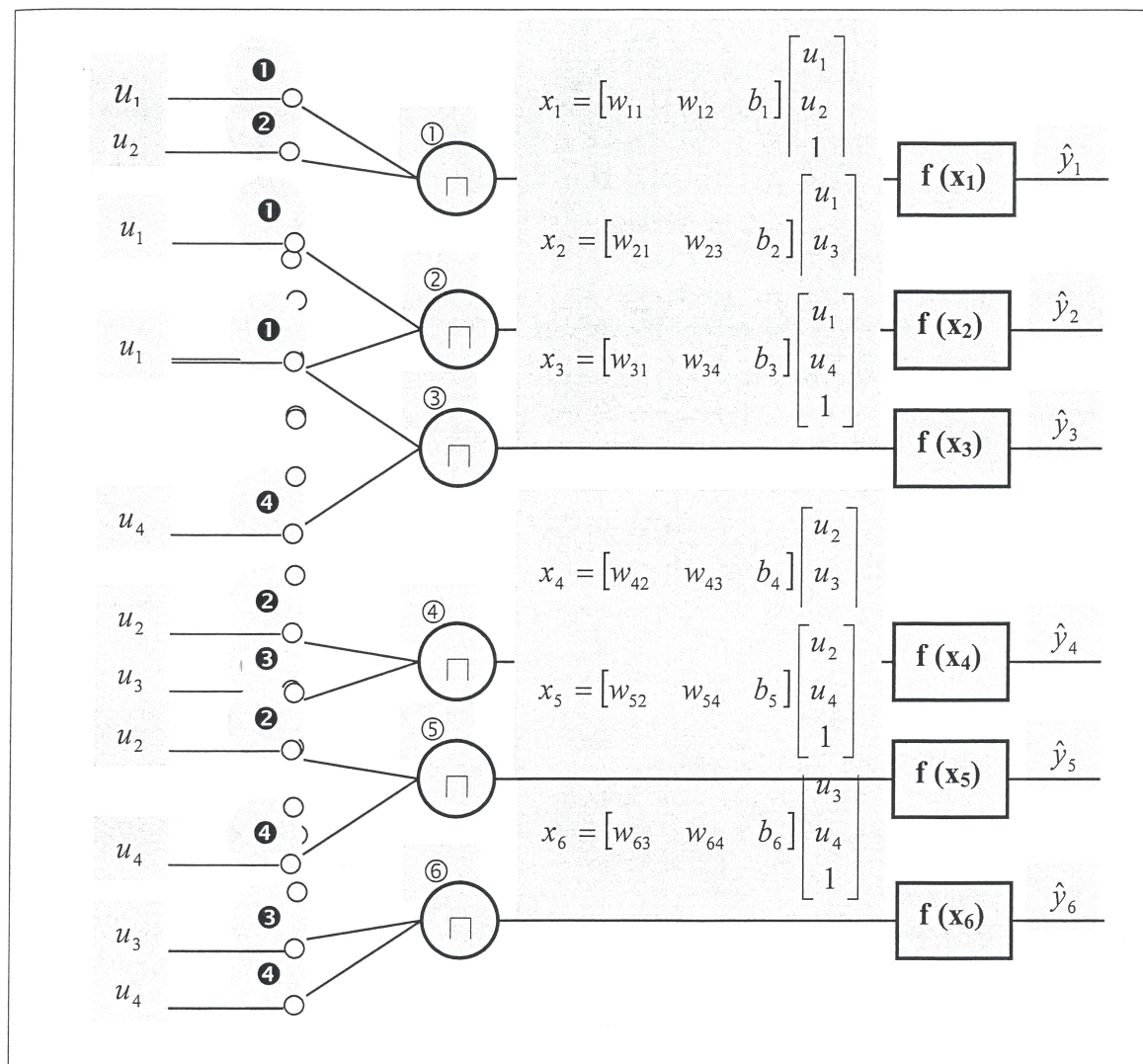


Figura 5. Șase modele parțiale pentru prognoza procesului stocastic

Deoarece alegerea lui n este arbitrară (ipotetic), este perfect posibil ca unul din aceste modele să fie acceptabil ca model pentru procesul aleator supus analizei pentru modelare neurală.

7. Pregătirea modelelor parțiale pentru competiție și selecție

Pentru pregătirea fiecăruia din modelele parțiale de forma:

$$y_i^j = w_{ji} \cdot u_i + w_{jk} \cdot u_k + b_j = \begin{bmatrix} w_{ji} & w_{jk} & b_j \end{bmatrix} \begin{bmatrix} u_i \\ u_k \\ 1 \end{bmatrix} \quad (6)$$

în care $i, k = 1 \dots n$; $k \neq i$; $k, j \in \{1, 2, \dots, C_n^2\}$, se folosesc datele inițiale $y_1 \dots y_N$ decupate dintr-un proces stocastic care trebuie prognozat, proces reprezentat în figura 6.

$$\begin{cases} w_{11}y_2 + w_{12}y_1 + b_1 = y_3 \\ w_{11}y_3 + w_{12}y_2 + b_1 = y_4 \\ w_{11}y_4 + w_{12}y_3 + b_1 = y_5 \\ \vdots \\ w_{11}y_7 + w_{12}y_6 + b_1 = y_8 \end{cases} \quad (7)$$

De exemplu, pentru modelul # 1 din figura 5, în acest caz, datele disponibile din figura 7 permit scrierea a 8 egalități (7) bazate pe datele din setul de antrenament $\{y_1 \dots y_8\}$. Sistemul de ecuații (7) poate fi scris în forma matriceală:

$$U \cdot \Theta = Y \quad (8)$$

în care, Θ este vectorul parametrilor, U este matricea 6×3 a datelor de intrare aplicate la intrările u_1, u_2 și biasului b_1 , iar Y este vectorul valorilor dorite la ieșirea neuronului respectiv. Acest sistem conține singurele necunoscute w_{11}, w_{12} și b_1 , dar nu poate fi rezolvat deoarece matricea U nu este pătrată, (numărul de ecuații este mai mare decât numărul necunoscutelor). Pentru a evita acest inconvenient, se înmulțește ecuația matriceală (9) la stânga cu U^T și rezultă:

$$U^T U \Theta = U^T Y \quad (9)$$

în care

$$U^T = \begin{bmatrix} y_2 & y_3 & y_4 & y_5 & y_6 & y_7 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Sistemul (9) este compatibil, noua matrice $U^T U$ având dimensiunea 3×3 . Sistemul poate fi rezolvat, cu condiția ca aceasta matrice să nu fie singulară, (determinantul ei să nu fie nul).

Soluția sistemului (9) furnizează estimări ale parametrilor predictorului.

$$U^T U = \begin{bmatrix} \sum_{k=2}^7 y_k^2 & \sum_{k=2}^7 y_k y_{k-1} & \sum_{k=2}^7 y_k \\ \sum_{k=2}^7 y_k y_{k-1} & \sum_{k=2}^7 y_{k-1}^2 & \sum_{k=2}^7 y_{k-1} \\ \sum_{k=2}^7 y_k & \sum_{k=2}^7 y_{k-1} & 6 \end{bmatrix}$$

$$U^T Y = \begin{bmatrix} \sum_{k=2}^7 y_k y_{k+1} \\ \sum_{k=2}^7 y_{k-1} y_{k+1} \\ \sum_{k=2}^7 y_{k+1} \end{bmatrix}$$

În mod similar, se calculează parametrii pentru celorlalte cinci modele din figura 5. Având valorile parametrilor $\hat{w}_{11}, \hat{w}_{12}, \hat{b}_1$ se pot calcula valorile $\hat{y}$ pentru datele de intrare din setul de verificare. Valorile $\hat{y}_1^1, \hat{y}_1^2, \dots, \hat{y}_1^6$ obținute la ieșirea neuronului se pot compara cu valorile dorite din setul de verificare, rezultând vectorul erorilor E_1 și suma erorilor pătratice E_1^2 pentru acest model:

$$E_1 = \begin{bmatrix} y_3 - \hat{y}_1^1 \\ y_4 - \hat{y}_1^2 \\ y_5 - \hat{y}_1^3 \\ y_6 - \hat{y}_1^4 \\ y_7 - \hat{y}_1^5 \\ y_8 - \hat{y}_1^6 \end{bmatrix} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{bmatrix} \quad \text{și} \quad E_1^2 = \sum_{k=1}^6 \varepsilon_k^2$$

În mod analog, se obțin valorile $E_2^2 \dots E_6^2$ pentru celelalte cinci modele considerate la concursul de selecție pentru ocuparea unui loc în primul strat ascuns al rețelei neurale.

Conform criteriului erorii pătratice, pe primul loc se află candidatul cu eroarea pătratică cea mai mică, pe ultimul loc aflându-se modelul cu eroarea pătratică cea mai mare. În faza a doua a concursului, se alege maxim n modele neuronale (unde conform exemplului considerat în figura 5 numărul total de intrări al rețelei este $n=4$), dintre modelele cu erorile cele mai mici dar respectând și „**Criteriul reprezentării globale**”, care asigură pentru neuronii aleși prezența tuturor variabilelor de intrare ($u_1 \dots u_4$). În cazul din figura 4, am putea alege conform acestui criteriu trei neuroni din cei șase. Perechile formate din trei neuroni ar putea fi, de exemplu:

① ② ③ sau ① ③ ④ sau, ③ ⑤ ⑥ etc.

Să presupunem că, din punctul de vedere al preciziei de prognoza, neuronul ③ se află pe poziția 1, neuronul ④ pe poziția 2, iar neuronul ⑤ pe poziția 3. Acești neuroni au ieșirile y_{11} , y_{12} , y_{13} , (în care primul indice se referă la primul strat ascuns, iar cel de al doilea la poziția neuronului în stratul ascuns din care face parte).

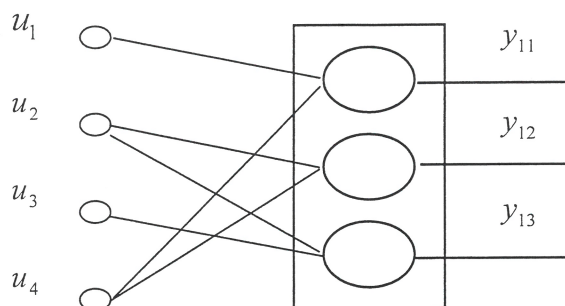


Figura 9. Primul strat ascuns al rețelei neuronale

S-a construit din acești 3 neuroni selectați o parte a RN, reprezentată în figura 9, din care se observă că un strat ascuns, format din acești trei neuroni, respectă criteriul reprezentării globale.

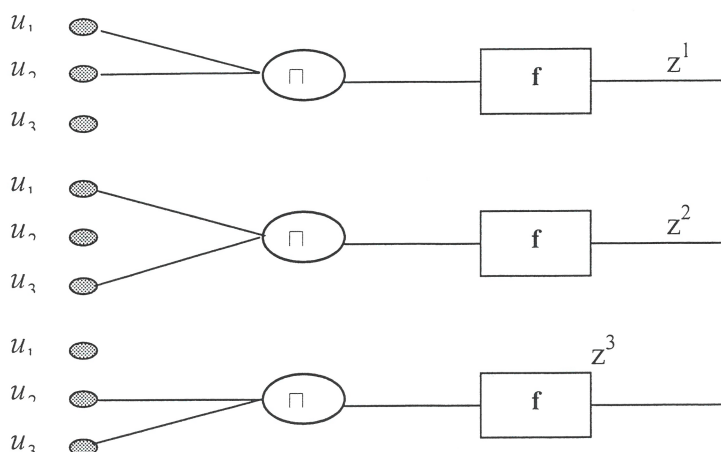


Figura 10. Modele care concursa pentru un loc în al 2-lea strat ascuns din RN

Ieșirile acestor 3 neuroni din *primul strat ascuns* constituie intrări pentru neuronii din stratul ascuns următor al RN. Folosind același procedeu, se poate obține o populație formată din alte trei modele prezentate în figura 10. Din această nouă populație de neuroni ale căror ieșiri au fost notate cu z se va selecta cel de *al doilea strat ascuns* al rețelei care modelează comportarea statică a procesului considerat. În urma aplicării algoritmului și modificării numărului de neuroni pe strat, s-a constatat experimental ca eroarea pe setul de verificare prezintă un minim în funcție de numărul de epoci de antrenament (de straturi).

Pentru stratul 2, au fost selectate numai două modele din 3, pe baza criteriului erorii pătratice celei mai mici (figura 11). Să presupunem că au fost selectați al doilea și al treilea dintre neuroni ca fiind primii clasificați în topul modelelor parțiale, aferente celui de al doilea strat. Ieșirile neuronilor din al doilea strat sunt intrări pentru *neuronul din stratul de ieșire* (al treilea strat) ca în figura 11. Din figura 11 se poate observa clar că aceasta *nu este total conectat* ca cel din figura 4 construit pe baza teoremei Kolmogorov. De remarcat că o asemenea procedură de optimizare a numărului de straturi poate fi aplicată și pentru determinarea numărului de neuroni pe strat.

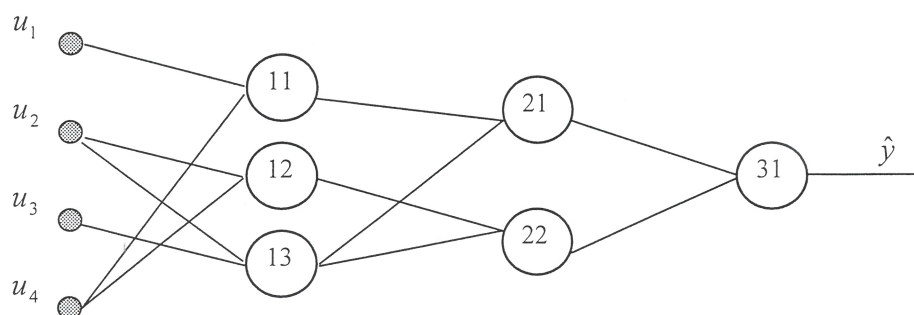


Figura 11. Rețea de predicție proiectată prin algoritmul evolutiv

Oprirea antrenamentului înainte ca eroarea de verificare să înceapă să crească este întâlnită în literatura de specialitate [4] sub denumirea de „*criteriul de compromis între eroarea de deviație și cea de dispersie*”. Pentru aceasta, este necesară existența celor două seturi distincte de date pentru *antrenamentul rețelei* și pentru *verificarea* modelului neural.

8. Concluzii

Aplicarea teoremei Kolmogorov în proiectarea structurii rețelei neuronale pentru modelarea unor semnale în vederea predicției, furnizează structuri prohibitive total conectate ale rețelei neuronale cu un singur strat ascuns. În cazul structurilor proiectate pe baza teoremei Kolmogorov numărul de neuroni depinde de numărul n de intrări ale rețelei. Dacă numărul de intrări $n > 3$ se recomandă metoda evolutivă de proiectare care presupune în executia următorilor 8 pași ai algoritmului hibrid rezultat din combinarea algoritmului evolutiv cu algoritmul backpropagation [4]. În concluzie, aplicarea algoritmului hibrid de obținere a unei structuri optimale a modelului neural, presupune parcurgerea următorilor opt pași de calcul:

Pasul 1. Se formează mulțimea modelelor parțiale cu $m < n$ intrări în care $m=2$ (sau $m=3$), caz în care numărul de modele este $M = C_n^2$, respectiv $M = C_n^3$.

Pasul 2. Folosind setul de instruire, format din circa jumătate din datele inițiale ale seriei de timp, se realizează calculul parametrilor datelor parțiale.

Pasul 3. Se calculează eroarea pătratică medie pe setul de verificare pentru toate cele C_n^2 modele parțiale și se ordonează modelele după valoarea erorii pătratice medii.

Pasul 4. Se realizează trierea a cel mult n neuroni din cei C_n^2 care sunt promovați în strat, după criteriul valorii erorii celei mai mici și criteriul reprezentării globale.

Pasul 5. Se adaugă rețelei stratul de ieșire cu un singur neuron, reprezentând o RN cu un singur strat ascuns și se face instruirea cu algoritmul BKP, inițializat cu valorile algoritmului la pasul 1 și apoi se calculează E_1^2 pe setul de verificare.

Pasul 6. Se trece la proiectarea unui nou strat ascuns repetând pașii 1-5 anteriori. În noul strat, intrările sunt ieșirile neuronilor selectați în stratul ascuns anterior.

Pasul 7. Se adaugă RN stratul de ieșire, rezultând o rețea cu două straturi ascunse, care se antrenează cu BKP , inițializând cu valorile obținute la pașii anteriori 1-6 și calculând în final E_2^2 .

Dacă $E_2^2 > E_1^2$ este propusă rețeaua cu un singur strat ascuns, construită la pasul 5. Altfel, $E_2^2 < E_1^2$ procesul de introducere a noi straturi ascunse continuând cu pasul 8.

Pasul 8. Se construiește un nou strat obținând o rețea cu trei straturi ascunse, un strat de intrare inactiv și un strat de ieșire cu un singur neuron. Pe setul de verificare, se obține E_3^2 . Dacă $E_3^2 < E_2^2$, este propusă rețeaua cu trei straturi ascunse. Astfel, se justifică o structură cu două straturi ascunse.

Bibliografie

1. **DUMITRAȘ, A.:** Proiectarea rețelelor neurale artificiale, Ed. Odeon, București, 1997.
2. **TERTIȘCO, M., GH. PETRACHE:** Identificarea proceselor, Ed. I.P.B, 1978.
3. **DUMITRACHE, I., N. CONSTANTIN, M, DRAGOICEA:** Rețele neurale. Identificarea și conducerea proceselor, Ed. Matrix Rom, 1999.
4. **ȘERBAN, S.:** Modelare și simulare, Ed. Printech Rom, 2007.