MODELAREA NEURALĂ A SCHIMBĂTOARELOR DE CĂLDURĂ

Condur George

g_condur@yahoo.com

Universitatea Politehnica Bucureşti

Rezumat:Lucrarea prezintă un nou algoritm de tip evolutiv privind proiectarea structurii şi antrenarea reţelelor neurale. Ca studiu de caz, se dezvoltă aplicarea reţelelor neurale în determinarea experimentală a modelului matematic static al sistemelor industriale caracterizate prin schimb de căldură. Este prezentat un algoritm original de generare automată a programului experimentului activ la trei niveluri, destinat achiziţiei datelor experimentale pentru instruirea reţelei neurale utilizată ca model. Este aplicat un criteriu euristic de partiţionare a datelor din setul de instruire în: ***setul*** de date pentru ***antrenare*** a reţelei şi setul de date pentru verificarea modelului. Se aplică pentru instruire un algoritm hibrid, rezultat prin combinarea algoritmului backpropagation cu un algoritm evolutiv propus în [5]. Se prezintă exemple şi un studiu de caz.

Cuvinte cheie: reţele neurale, experiment activ, schimbător de căldură, backpropagation, algoritm evolutiv.

Abstract: The paper presents an evolutionary algorithm guy concerning the neural nets structure projection and entrainment. As study of the development adhibition neural nets in the industrial systems with shifts of heats. Is presented an original algorithm for generation a program of active experiment destined to experimental data acquisition. 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 proposed for training an hybrid algorithm of training result through the combinations of the **backpropagation** agorithm with the evolutional suggested algorithm of the author work [5].

Keywords: Neural nets, heat interchanges, active experiments, neural nets training, back propagation.

1. Introducere

În literatura de specialitate, sunt prezentate două categorii de metode de identificare a proceselor industriale continue în regim static: ***metode pasive*** (în care sunt achiziţionate date experimentale intrare – ieşire în regim de funcţionare normală a instalaţiei tehnologice fără a forţa aducerea sistemului în anumite regimuri impuse) şi ***metode active*** (în care mărimile de intrare sunt modificate forţat) [4]. Dintre metodele active pentru identificare statică, cea mai răspândită este ***metoda experimentului factorial*** .Întrucât experimentul este activ, mărimile de intrare sunt modificate forţat astfel încât măsurările mărimilor de intrare şi ieşire se pot efectua după un program prestabilit. Pentru determinarea modelului static, se aplică de cele mai multe ori metode experimentale, aceasta în cazurile în care modelul procesului este destinat conducerii automate, iar complexitatea procesului cercetat face dificilă determinarea pe cale analitică a modelului matematic $y = F(u_1, u_2, \ldots, u_r)$. Acest model caracterizează comportarea procesului în regim staţionar şi exprimă dependenţa între ieşirea y şi intrările u ale modelului procesului. Imaginea grafică a acestei funcţii în spaţiul r+1 dimensional este denumită şi *suprafaţă de* ***răspuns***. Aceste metode experimentale implică explorarea suprafeţei de răspuns prin modificarea variabilelor de intrare şi măsurarea mărimii de ieşire y în diverse puncte (de pe această suprafaţă) corespunzătoare fiecărei combinaţii de valori ale mărimilor de intrare u. Modificarea variabilelor de intrare este limitată de considerente de ordin tehnologic, astfel încât nu se poate explora întreaga suprafaţa de răspuns, ci numai o porţiune a acesteia în jurul ***punctului nominal de funcţionare*** al instalaţiei tehnologice.

Metoda experimentului factorial în varianta propusă de Box şi Wilson [4], cuprinde următoarele etape:

- determinarea unui model matematic liniar al procesului în zona regimului tehnologic iniţial folosind date din proces obţinute printr-un experiment special organizat numit „experiment la 2 niveluri";
- deplasarea regimului tehnologic spre zona de extrem a suprafeţei de răspuns, zona în care trebuie să se afle regimul nominal;
- efectuarea unui "experiment la 3 niveluri"în zona de extrem, pentru determinarea modelului matematic adecvat acestei zone neliniare a suprafeţei de răspuns care conţine şi punctul aferent regimului nominal al procesului.

În lucrare, se are în vedere modelarea proceselor neliniare regresionale de tipul:

$$y = \varphi^T(k)\theta_* \tag{1}$$

în care y este ieşirea procesului, φ este vectorul regresorilor, iar θ_* vectorul parametrilor [4] .Scopul prezentei lucrări este să rezolve problemele pe care le ridică utilizarea unei reţele neurale feedforward ca model static în locul modelului (1), pentru un schimbător de căldură din reţeaua orăşenească de termoficare. Lucrarea conţine rezultatele cercetării ştiinţifice, desfăşurate de autor în perioada doctoranturii, privind rezolvarea armatoarelor probleme:

- achiziția datelor experimentale intrare – ieşire printr-un experiment activ ortogonal, destinate instruirii rețelei neurale ca model al procesului;
- propunerea unui criteriu de partiționare a datelor experimentale de instruire în două seturi de date(setul de antrenare a rețelei şi setul de verificare);
- elaborarea algoritmului de proiectare şi testarea unei structuri a rețelei neurale, adoptată ca model.

În continuare, este prezentat un algoritm original de generare automată a programului experimentului activ la trei niveluri, destinat achiziției datelor experimentale pentru instruirea rețelei neurale. Se propune un criteriu euristic de partiționare a datelor din setul de instruire în: setul de date pentru antrenarea rețelei şi setul de date pentru verificarea modelului neural, obținut după antrenarea rețelei. Se aplică pentru instruire un algoritm de instruire hibrid, rezultat prin combinarea algoritmului backpropagation cu un algoritm evolutiv. Se prezintă exemple şi un studiu de caz.

2. Planificarea experimentului activ pentru obținerea datelor de instruire a rețelei neurale (RN) ca model al procesului tehnologic

Se consideră un proces tehnologic caracterizat printr-o comportare statică neliniară, care poate fi descrisă de o funcție cu structură necunoscută:

$$y = F(x_1 \ldots . x_n) \tag{2}$$

în care $x_1 \ldots . x_n$ sunt intrările procesului, iar y ieşirea acestuia.

Funcția necunoscută $F(x_1 \ldots . x_n)$ admite dezvoltarea în serie Taylor în jurul punctului $\{x_{10} \ldots . x_{n0}\}$ din spațiul intrărilor astfel că, trunchiind seria Taylor, se poate adopta pentru proces un model polinomial. Se consideră un punct de coordonate $\{x_{10} \ldots . x_{n0}\}$ ca fiind caracteristic pentru regimul „nominal" al procesului. În cursul funcționării normale a instalației tehnologice, mărimile exogene ale procesului $x_i \quad (i = 1 \ldots n)$ nu depăşesc valorile limită admise:

$$x_i = x_{i0} \pm \Delta_i ; \quad \Delta_i > 0 \tag{3}$$

în care $2\Delta_i > 0$ reprezintă banda de variație admisă prin prescripții tehnologice şi care, de obicei, este în jur de (10-20) % din valorile de regim nominal ale intrărilor $x_i \quad (i = 1 \ldots n)$.

Banda în care mărimile x_i vor lua valori este deci cunoscută şi limitată de valorile minimă x_{im}, respectiv, maximă x_{iM}.

$$\begin{aligned} x_{im} &= x_{i0} - \Delta_i \\ x_{iM} &= x_{i0} + \Delta_i \end{aligned} \tag{4}$$

în care x_{i0} reprezintă valorile de regim nominal (inițial) ale intrărilor procesului.

Pentru achiziția datelor de instruire am considerat un experiment activ în care intrările sunt „forțate" să ia valori impuse în domeniul admisibil (4)de valori ale intrărilor procesului pe parcursul experimentului special organizat. Acest experiment în care intrările pot lua doar două valori definite de relația (4) se numeşte „experiment la 2 niveluri". În cazul experimentului la 2 niveluri, se pot efectua un număr N= 2^n de măsurători distincte în punctele cu coordonatele descrise de (4) în zona de regimuri tehnologice admisibile pentru procesul tehnologic respectiv. Experimentul la trei niveluri vizează cazul în care variabilele de intrare pot lua fiecare numai trei valori (în centru şi la capetele intervalului cunoscut):

$$x_i \in \{x_{i0}, \quad x_{im}, \quad x_{iM}\} \tag{5}$$

În acest caz, se pot efectua maximum 3^n măsurători intrare-ieşire. Pentru exemplificare, considerăm cazul n = 2, care va admite $3^2 = 9$ măsurători experimentale în punctele reprezentate în figura 1, din planul variabilelor x_1, x_2. În cazul experimentului la 2 niveluri, reprezentarea grafică se reduce la cele patru puncte din colturile dreptunghiului din figura 1.

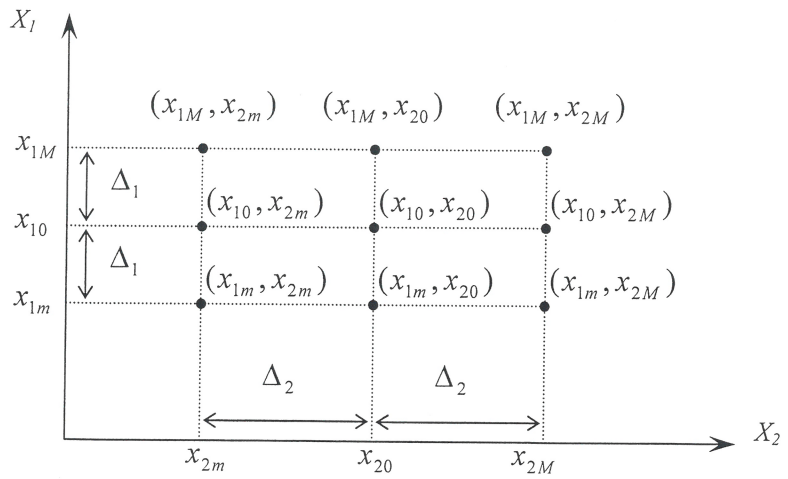


Figura 1. Punctele din planul variabilelor de intrare în care se fac măsurători

Astfel, în cazul experimentului la 3 niveluri, matricea X a datelor de intrare (valorile aplicate la intrarea procesului necunoscut) va fi următoarea:

x_1	x_{10}	x_{1m}	x_{1M}	x_{10}	x_{10}	x_{1m}	x_{1m}	x_{1M}	x_{1M}
x_2	x_{20}	x_{20}	x_{20}	x_{2m}	x_{2M}	x_{2m}	x_{2M}	x_{2m}	x_{2M}

(6)

Acest procedeu de planificare a experimentului ne permite ca, pentru modelul neural, să utilizăm date centrate şi normate, fapt care asigură, într-o oarecare măsură, standardizarea matricei datelor de intrare pentru numărul de variabile de intrare dat.

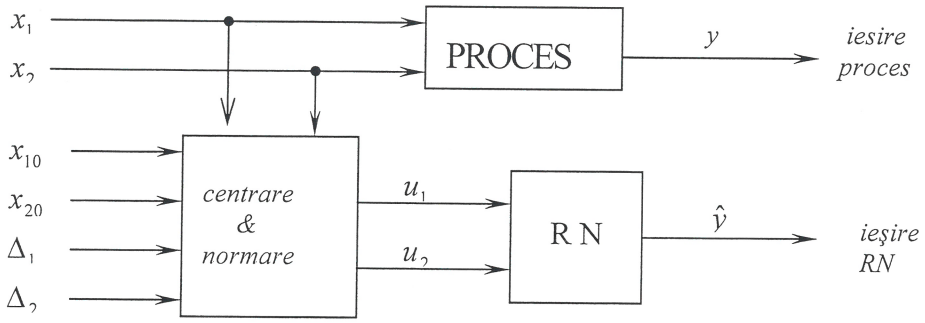


Figura 2. Blocul de prelucrare preliminară a datelor de intrare

Centrarea datelor se va face prin x_{i0}, iar normarea prin banda Δ_i. Noile variabile de intrare (pentru modelul neural) vor fi:

$$u_i = \frac{x_i - x_{i0}}{\Delta_i} \in \{-1, 0, 1\} \tag{7}$$

În aceste condiţii, se obţin din relaţiile (6) şi (7) vectorii de date:

u_1 = [0, -1,+1,0,0 ,-1,-1,+1,+1] şi u_2= [0, 0,0,-1,+1 ,-1,+1,-1,+1] (8)

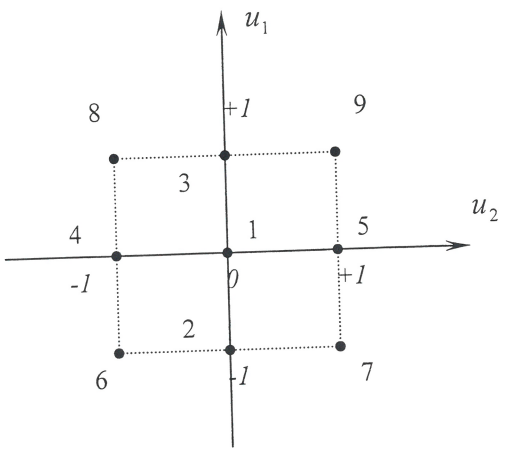
respectiv noua matrice U a datelor de intrare în reţeaua neurală este:

0	-1	+1	0	0	-1	-1	+1	+1
0	0	0	-1	+1	-1	+1	-1	+1

În schema bloc a sistemului din figura 2, pentru achiziţia şi prelucrarea datelor în vederea aplicării lor la intrarea modelului neural RN al procesului, este prevăzut un bloc plasat între proces şi reţeaua neurală denumit

blocul de prelucrare preliminară, cu rol în centrarea şi normarea intrărilor xi, măsurate direct din proces.

Aşadar, datele de antrenament ale reţelei sunt cele din matricea U dată de (8) şi sunt poziţionate conform graficului din figura 3.



a)

$$\{6,7,8,9\} \cup \{2,3,4,5\} \cup \{1\}$$

b)

Figura 3. Reprezentarea grafică a datelor experimentale la intrarea RN

Figura 3a ilustrează poziţionarea datelor în spaţiul variabilelor de intrare, iar 3b prezintă partiţionarea datelor în ordinea dispersiei diverselor grupuri de date rezultate după partiţionare în ordinea dispersiei lor.

3. Criteriul dispersional de partiţionare a setului de instruire

Setul datelor de instruire a modelului neural trebuie partiţionat în două entităţi: setul de antrenare şi setul de test sau de verificare al RN după antrenare. Criteriul intuitiv adoptat pentru partiţionare derivă din necesitatea regularizării modelului neural. Regularizarea are ca scop creşterea capacităţii de generalizare a modelului, în sensul că acesta trebuie să se comporte corect şi pe alte date de intrare care nu au făcut parte din setul de instruire. Pentru aceasta, există recomandarea euristică (intuitivă) ca din cele N date, o parte sunt repartizate în setul de antrenare. Aceste date trebuie să fie cele care sunt cel mai depărtate de centrul mulţimii de date (coordonatele căruia sunt mediile calculate ale intrărilor măsurate). Din punct de vedere al dispersării lor, datele din (8), respectiv din figura 3b se împart în trei submulţimi şi anume:

dispersie maximă	**dispersie medie**	**dispersie minimă (nulă)**
$\{6,7,8,9\}$	$\{2,3,4,5\}$	$\{1\}$

Regula intuitivă de selecţie a datelor în setul de antrenament indică pentru antrenament datele cu dispersie mare (adică acoperitoare din punctul de vedere al domeniului maxim de variaţie, iar aproximarea în cazul numărului mic de date să se facă în favoarea setului de antrenament, în ceea ce priveşte cardinalitatea.

O demonstraţie analitică ori o teoremă care să demonstreze regula propusă nu există. Explicaţiile legate de folosirea acestui principiu intuitiv ar fi următoarele:

- datele de intrare puternic dispersate provoacă o componentă utilă în semnalul de răspuns, mai favorabilă din punct de vedere al raportului semnal/zgomot;
- datele dispuse la periferia cluster-ului vor asigura acoperirea mai corectă a domeniului de utilizare în viitor a modelului, interpolând prin generalizare şi punctele din jurul centrului clusterului;
- în principiu, nu se doreşte o funcţie de aproximare $\hat{F}$, care să treacă prin toate punctele din setul de instruire şi nici de antrenament. Aceasta asigură reţelei neurale o capacitate mai mare de generalizare.

În continuare, se prezintă verificarea acestui criteriu pe datele (9) reprezentate în figura 3 în care setul de antrenament a fost format din datele cele mai depărtate de centru (adică cele din colturi), iar datele

pentru setul de verificare au fost alese cele mai apropiate de centru.

Calculul mediilor datelor de antrenament a RN (centrate şi normate):

$$M(u_1) = \frac{(-1)+(-1)+1+1}{4} = 0$$

$$M(u_2) = \frac{(-1)+1+(-1)+1}{4} = 0$$

Calculul dispersiilor datelor de antrenament a RN (centrate şi normate):

$$\sigma^2_{u_1}(A) = \frac{1}{N}\sum_{k=1}^{N}\left[u_1(k) - M(u_1(k))\right]^2 = \frac{1}{4}\left[(-1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2\right] = 1$$

$$\sigma^2_{u_2}(A) = \frac{1}{N}\sum_{k=1}^{N}\left[u_2(k) - M(u_2(k))\right]^2 = \frac{1}{4}\left[(-1)^2 + 1^2 + (-1)^2 + 1^2\right] = 1$$

Pentru ***setul de verificare***, destinat testării reţelei neurale dispersiile sunt :

$$\sigma^2_{u_1}(T) = \frac{1}{5}\left[0^2 + 0^2 + 1^2 + 0 + (-1)^2\right] = 0.4$$

$$\sigma^2_{u_2}(T) = \frac{1}{5}\left[0^2 + 1^2 + 0 + (-1)^2 + 0^2\right] = 0.4$$

Se observă că dispersiile datelor de verificare sunt inferioare celor de antrenare ceea ce înseamnă că partiţionarea făcută, practic prin inspectarea vizuală a datelor achiziţionate (figura 3), respectă întrutotul criteriul euristic de partiţionare introdus mai sus.

4. Proiectarea arhitecturii reţelei neurale ca model static

Pentru adoptarea unei arhitecturi iniţiale , pentru o reţea neurală de tip feedforward, majoritatea autorilor invocă în principal teorema Kolmogorov de aproximare şi „parsimony principle" enunţat de filosoful medieval William of Ockham în 1340 [2], de la care s-a plecat în cercetările declanşate exploziv în acest domeniu, în ultimii ani. Adaptat la problema reţelelor neurale, principiul lui Ockham şi teorema Kolmogorov pot fi reformulate astfel:

Principiul Okham,"Fiind date două reţele, dintre care una supradimensionată (ca număr de parametri), şi acelaşi set de date de instruire, reţeaua supradimensionată va greşi mai mult în viitor, mai exact, va generaliza mai slab, atunci când i se vor aplica la intrare forme care, deşi fac parte din clasele aferente problemei date, nu au făcut parte din setul de instruire a reţelei supradimensionate."

Teorema Kolmogorov se referă la aproximarea unei funcţii F(x) continue prin $\hat{F}(x) = F(w,x)$.

Fiind dată $F: \{0,1\}^n \rightarrow R^m$ ***pentru orice funcţie continuă y = F(x), aceasta poate fi implementată printr-o reţea neurală de tip feedforward cu n intrări şi m ieşiri.***

În figura 4, este prezentată schema Kolmogorov în cazul a n intrări şi m ieşiri ale reţelei.

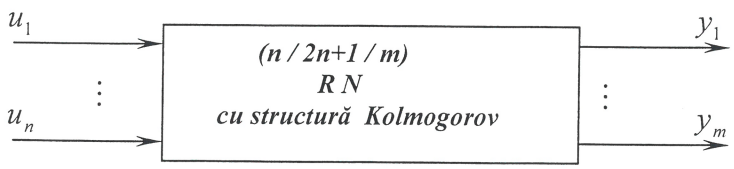


Figura 4. Reţea *(n / 2n+1 / m)de tip Kolmogorov*

Reţeaua neurală cu n intrări şi m ieşiri concepută ca structură pe baza teoremei Kolmogorov are un singur strat ascuns cu (2n+1) neuroni, iar stratul de ieşire conţine m neuroni. În figura 4, este prezentată schema bloc a unei astfel de reţele neurale, simbolizată sub forma: (n / 2n+1 / m). În exemplul din figura 5, reţeaua are structura (2 /5 /1) şi reprezentă reţeaua cu arhitectura Kolmogorov, pentru modelarea procesului din figura 2.

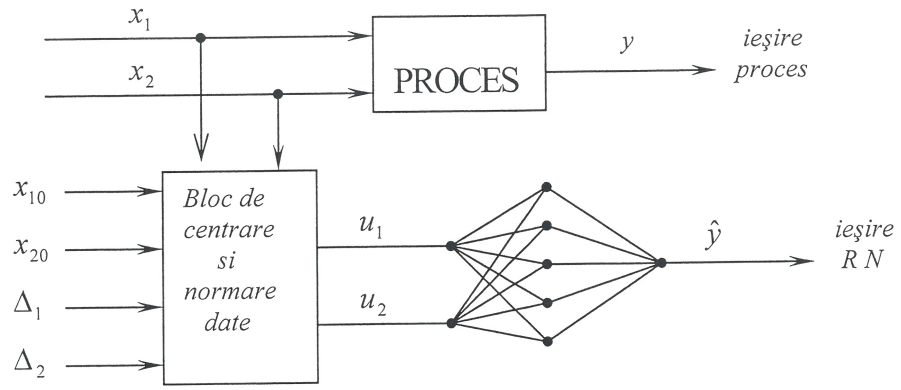


Figura 5. Rețea neurală cu arhitectura Kolmogorov

Este însă evident că teorema Kolmogorov dă răspuns doar la problema existenței unei structuri neurale pentru orice funcție nu și la problema de suficiență. Pentru cazurile particulare pot rezulta însă structuri mult mai simple. De exemplu, în cazul în care $F(u_1,u_2)$ descrie un plan, în acest caz particular rețeaua cu structura optimă din punct de vedere al numărului de conexiuni, neuroni, straturi etc. are în structura sa un singur neuron în timp ce pentru același caz din figura 2, structura modelului neural $\hat{F}(u_1,u_2)$ conform teoremei Kolmogorov de aproximare este mult mai complicată și este cea din figura 6.

Legenda figurii 6:

○ - neuroni pasivi din primul strat de reproducere a intrării la ieșire;
● - neuroni activi cu funcții de activare $(1+e^{-x})^{-1}$;
***i* – strat de intrare cu neuroni pasivi de reproducere;**
***a* – strat ascuns;**
***e* - strat de ieșire.**

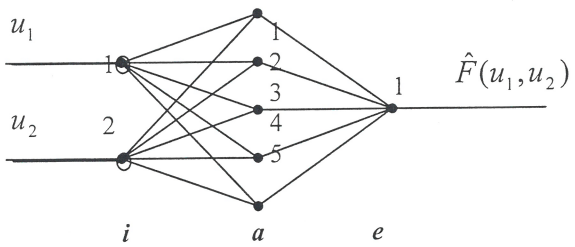


Figura 6. Structura Kolmogorov a modelului neural $\hat{F}(u_1,u_2)$

Parametrii rețelei neurale cu structură Kolmogorov de aproximare sunt următorii

pentru stratul ascuns

$$W_a = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{25} \\ b_1 & b_2 & b_3 & b_4 & b_5 \end{bmatrix}$$

ponderi pe conexiunile intrare-neuron din stratul *a*

praguri (activări) pentru neuronii din stratul *a*

pentru stratul de ieșire

$$W_e = \begin{bmatrix} w_{1e} \\ w_{2e} \\ w_{3e} \\ w_{4e} \\ w_{5e} \\ b_e \end{bmatrix}$$

ponderi pe conexiunile neuroni stratul *e* - neuroni stratul *a*

pragul (activarea) neuronului din stratul de iesire *e*

Datorită faptului că nu se cunoaște structura modelului, informația privitoare la structura modelului neural trebuie extrasă din setul de date de instruire. În cazul din figura 7, se poate presupune spre exemplu că pentru același proces neliniar folosim model neural cu structura Kolmogorov F1 sau un model mult simplificat F3 subdimensionat. Există însa un model cu structura optimala F2. Neavând de unde ști acest lucru probăm pe setul S

de date cele trei modele, prin calcularea abaterii pătratice medii APM pentru fiecare din ele:

$$\underset{i\in\{1,2,3\}}{APM_i} = \sum_{t=1}^{N}\left(y_t - \hat{F}_i\right)^2 \qquad (10)$$

Cele trei valori pentru abaterea pătratică pentru F1, F2, F3 vor fi dispuse ca în figura 7. Modelul care respectă criteriul abaterii pătratice minime este $\hat{F}_2$ şi are o structura optimă minimală ca număr de parametri.

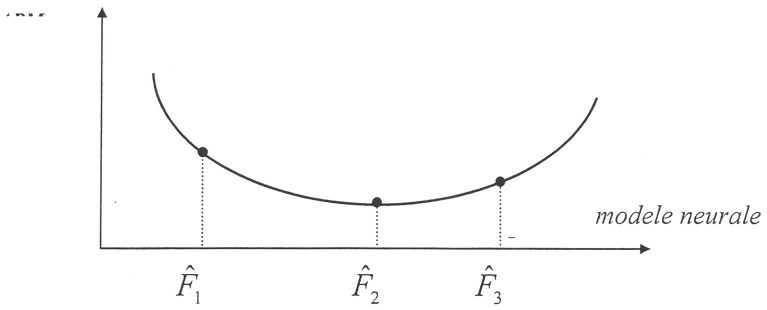


Figura 7. Criteriul APM de selecţie a modelului neural cu structura Kolmogorov

Probabil că adevărul este undeva la mijloc şi este necesară o procedură care să stabilească acest adevăr. Această problemă a proiectării reţelelor neurale nu şi-a găsit o procedură unică de rezolvare. Există o multitudine de metode. Întrucât comparaţia metodelor este dificil de realizat, deoarece, de regulă acestea nu au o bază teoretică clar definită şi sunt dificil de implementat (comparaţia se poate face eventual numai pe exemple), în aceasta secţiune a lucrării propunem o metodă pe care o susţinem prin exemplu.

5. Generarea automată a planului experimentului activ

În secţiunea precedentă, a fost prezentată procedura de obţinere a planului experimentului activ la 2 niveluri pentru două variabile, în care $u_i \in \{0, +1, -1\}$ care, practic, se compune din două părţi setul de antrenare şi setul de test al datelor de intrare.

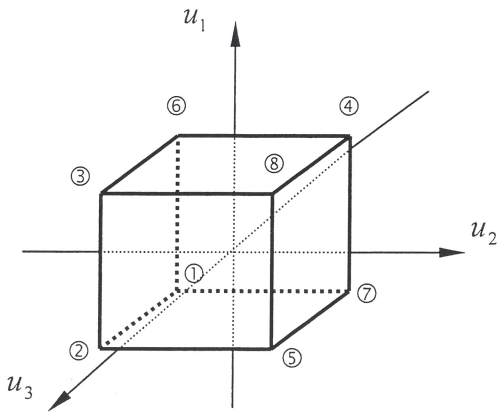


Figura 8. Poziţiile datelor de intrare în colţurile cubului

În cazul a trei variabile ,setul de antrenare conţine datele cele mai dispersate, poziţionate în colţurile unui hipercub cu centrul în origine (figura 8). În acest caz, datele sunt obţinute prin $2^3 = N_a$ măsurători, în care $u_i \in \{-1, +1\}$, $i \in \{1, 2, 3\}$. Acesta este un experiment factorial ortogonal la doua niveluri, al cărui plan este prezentat în următoarea matrice U_a (de dimensiune 3 x 8) a datelor de antrenament:

-1	-1	-1	+1	+1	-1	+1	+1
-1	-1	+1	+1	-1	+1	-1	+1
-1	+1	+1	-1	+1	-1	-1	+1

Se observă că datele pentru măsurătorile $k = 1, 2, 3, .., 7$ pot fi generate prin metoda vectorului $u^T = [-1, -1, -1]$ cu reacție de tip produs $u_1 * u_2 = u_r$ și deplasare prealabilă, ilustrată prin schema bloc din figura 9. Algoritmul descris în figura 9 generează toate combinațiile, cu excepția ultimei combinații (pentru $k = 2^n$, unde n este numărul de variabile exogene). Adică, N - 1 combinații pot fi obținute algoritmic și ultima (k = N) poate fi adăugată, deoarece este totdeauna aceeași:

$$u_i = +1 \quad (\forall)\, i\,, \text{ la } k = N \tag{11}$$

O inițializare corectă este oricare alta în afară de (11). De exemplu, se alege de obicei starea inițială:

$$u_i = -1 \quad (\forall)\, i\,, \text{ la } k = 1 \tag{12}$$

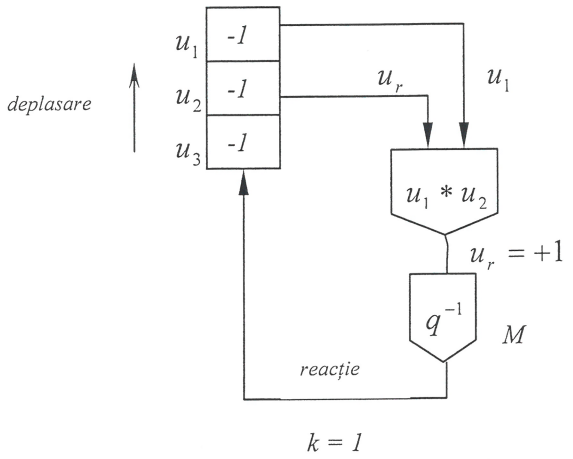


Figura 9. Algoritmul vectorului cu reacție și deplasare prealabilă

În continuare, din aceasta stare inițială o combinație ulterioară (pentru k = 2) se obține pe baza combinației anterioare, în trei pași:

Pasul 1 – se calculează și se memorează în M valoarea reacției:

$$M = u_1 * u_r$$

Pasul 2 – se deplasează informația din vector:

$$u_i = u_{i+1};\ i = 1, 2, .., n-1 \tag{13}$$

Pasul 3 – se încarcă M în elementul u_n al vectorului și se reia cu pasul 1.

Pentru obținerea corectă a celor N – 1 combinații, pentru diverși n = 2, 3, . . ,10 reacția u_r trebuie conectată la elementul din poziția r, care diferă în funcție de n, după cum este arătat în figura 10.

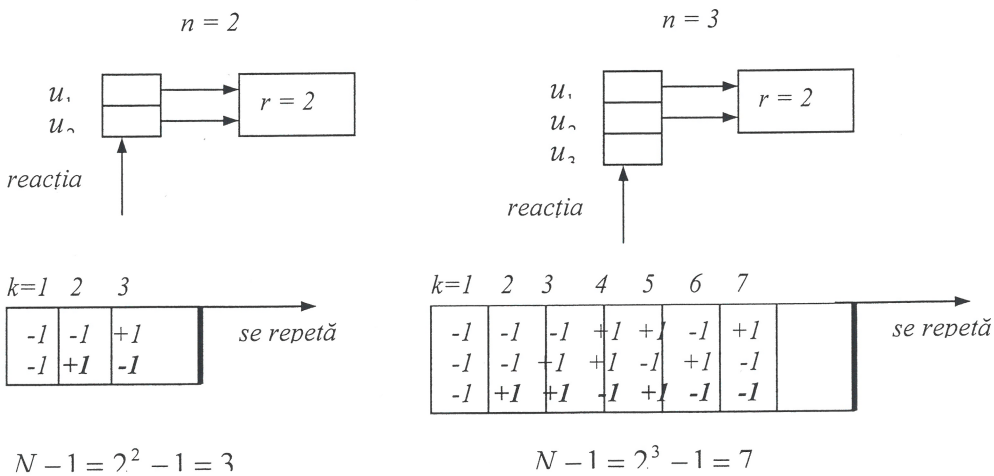


Figura 10. Modificarea reacției în funcție de n

Algoritmul trebuie oprit după N - 1 iterații, deoarece în continuare toate combinațiile se repetă ciclic după N – 1 iterații, unde $N = 2^n - 1$ .Reacțiile u_1 şi u_r sunt conectate la elementele vectorului din pozițiile 1, respectiv r, care se modifică în funcție de numărul de variabile. De exemplu, pentru n = 3 vom avea r = 2, ca în figura 10 Conectarea reacției r în funcție de n este dată în următorul tabel, conform [2], care conține date numai pentru vectori cu până la 10 elemente, adică procese cu până la 10 intrări.

n	2	3	4	5	6	7	8	9	10
r	2	2	4	3	6	5	2	5	8

Generarea planului experimentului activ 3^n s-a făcut prin intermediul unei secvențe de cod scrisă în limbajul Matlab. Datele de antrenament, respectiv cele de test sunt furnizate utilizatorului prin apelarea la prompterul Matlab a funcțiilor antrenare, respectiv testare.

Apelul funcției testare (număr_variabile) va genera matricea care va conține datele de test, caracterizate printr-o dispersie mai mică.

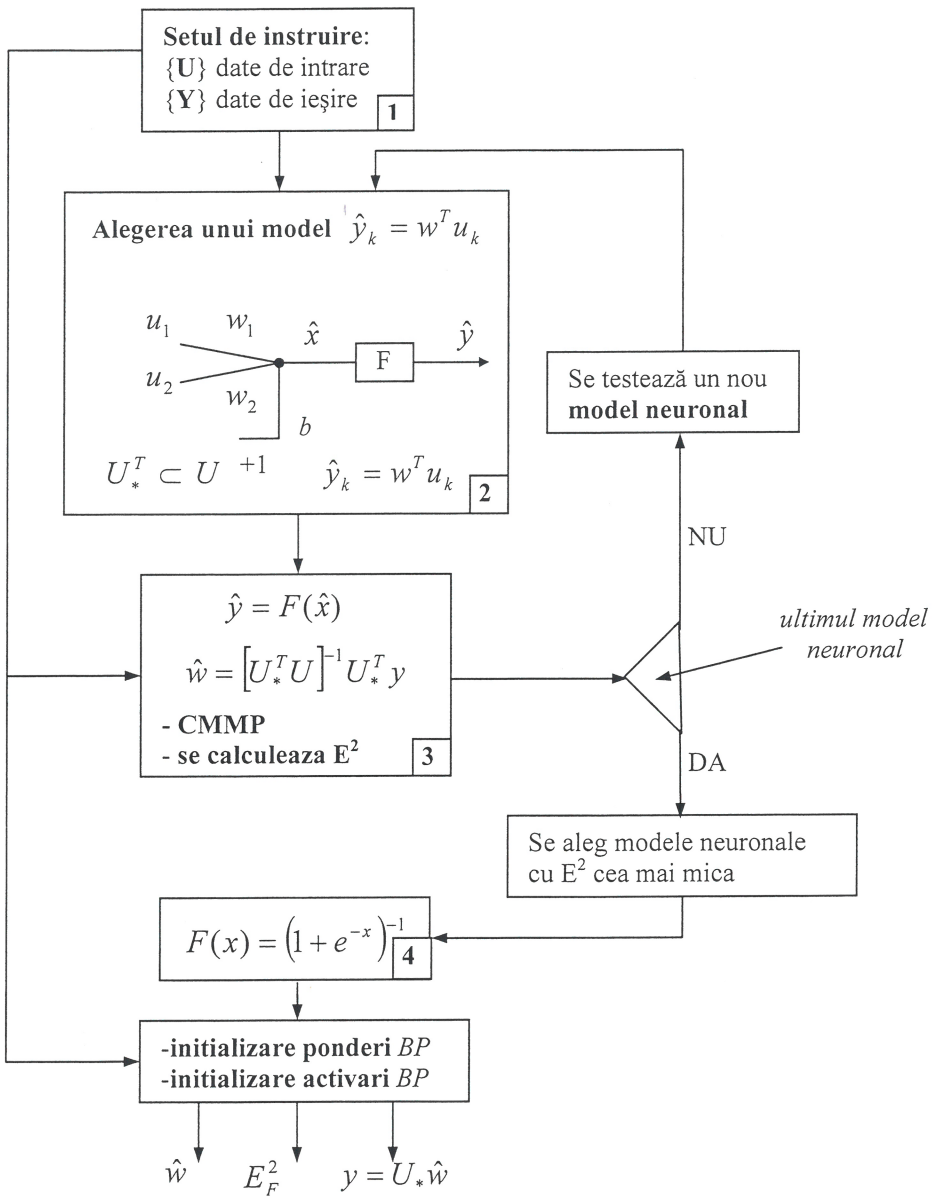


Figura 11. Inițializarea algoritmului BP prin folosirea criteriului MMP

Toate aceste seturi de date furnizate de către cele două funcții, vor fi folosite ulterior la antrenarea rețelei neurale care modelează procesul analizat.

Este evident faptul că se preferă o rețea neurală, care are o structură optimală din punct de vedere al numărului de straturi, numărului de neuroni în strat, numărului de conexiuni între straturi, etc. O asemenea rețea neurală are proprietăți de generalizare mai bune, un cost de implementare şi antrenare mai redus si, totodată, scade probabilitatea de estimare greşită a unor ponderi. Singurul neajuns al unor asemenea structuri neredundante este scăderea toleranței la defecte, ruperea unei conexiuni sau "moartea unui neuron" influențează rezultatele, ca în cazul rețelei neurale total conectată care este robustă la rupere.

Exista numeroase metode de proiectare a structurii rețelei neurale pe baza datelor din setul de instruire [5]. În prezenta lucrare, este utilizată metoda bazată pe principiul selecției artificiale a structurilor biologice, care face parte din categoria metodelor evolutive[6].

Informația disponibilă pentru proiectarea modelului neuronal al unui sistem neliniar este compusă din mulțimea perechilor de date intrare-ieşire folosite şi pentru instruire. Metoda folosită în acest scop se bazează pe faptul că datele de instruire conțin informații şi despre structura modelului (în speță a modelului neuronal), şi aceste informații pot fi extrase din date, prin metoda evolutivă. Scopul acestei secțiuni este prezentarea acestuia într-o variantă hibridă CMMP-back propagation, de instruire şi construire a rețelei neurale adecvată datelor, optimală din punct de vedere al numărului de straturi şi neuroni pe strat, în sensul abaterii pătratice minime. Ideile expuse aici sunt elaborate pentru identificarea sistemelor pe seturi mici de date şi descrise în lucrare, aceste idei fiind adoptate la cazul rețelelor neurale.

În figura 11, sunt prezentate schematic etapele de prelucrare a datelor în faza de antrenare, pentru inițializarea algoritmului back-propagation (BKP), în cadrul unuia din paşii algoritmului evolutiv [2]. Inițializarea ponderilor şi a activărilor neuronilor din modelul neural se va face prin utilizarea metodei celor mai mici pătrate.

6. Principiul metodei de inițializare a BKP prin CMMP, în cazul identificării statice a unui schimbător de căldură

S-a efectuat un experiment activ ortogonal pe un schimbător de căldură. Acest schimbător de căldură este caracterizat prin următorii parametri tehnologici de intrare: x_1 - debitul de agent termic; x_2 - temperatura apei menajere la intrare în schimbător; x_3 - presiunea; x_4 - temperatura la ieşire. Ieşirea y, este randamentul care exprimă cantitatea de căldură utila transmisă, pe tona de agent 24 de ore. Intrările au fost menținute constante în următoarele 8 regimuri.

k	$u_1(k)$	$u_2(k)$	$u_3(k)$	$u_4(k)$	y_k
1	-1	-1	-1	+1	103
2	-1	-1	+1	+1	97
3	-1	+1	-1	-1	93
4	-1	+1	+1	-1	87
5	+1	-1	-1	-1	103
6	+1	-1	+1	-1	97
7	+1	+1	-1	+1	113
8	+1	+1	+1	+1	107
Σ	0	0	0	0	800

(14)

Fiecare din cele opt regimuri a durat 24 de ore. Datele centrate şi normate sunt prezentate în (14).Pentru primul strat ascuns pot fi construite în total $C_4^2 = \frac{4!}{2! \cdot 2!} = 6$ modele parțiale.

Conform algoritmului din figura 11, rezultă la pasul 3 (pentru primul model neuronal parțial propus)avem următorul sistem:

$$\begin{cases} w_{11}^0(-1)+w_{12}^0(-1)+b_1^0=103 \\ w_{11}^0(-1)+w_{12}^0(-1)+b_1^0=97 \\ w_{11}^0(-1)+w_{12}^0(+1)+b_1^0=93 \\ w_{11}^0(-1)+w_{12}^0(+1)+b_1^0=87 \\ w_{11}^0(+1)+w_{12}^0(-1)+b_1^0=103 \\ w_{11}^0(+1)+w_{12}^0(-1)+b_1^0=97 \\ w_{11}^0(+1)+w_{12}^0(+1)+b_1^0=113 \\ w_{11}^0(+1)+w_{12}^0(+1)+b_1^0=107 \end{cases} \quad \text{sau} \quad y=\Phi\begin{bmatrix} w_{11}^0 \\ w_{12}^0 \\ b_1^0 \end{bmatrix} \tag{15}$$

în care w_{11}^0, w_{12}^0, b_1^0 sunt inițializări pentru algoritmul back propagation(BKP), iar Φ și y sunt

$$\Phi=\begin{bmatrix} -1 & -1 & +1 \\ -1 & -1 & +1 \\ -1 & +1 & +1 \\ -1 & +1 & +1 \\ +1 & -1 & +1 \\ +1 & -1 & +1 \\ +1 & +1 & +1 \\ +1 & +1 & +1 \end{bmatrix} \quad \text{și} \quad y=\begin{bmatrix} 103 \\ 97 \\ 93 \\ 87 \\ 103 \\ 97 \\ 113 \\ 107 \end{bmatrix}.$$

Deoarece matricea Φ nu este pătratică, prin înmulțire la stânga cu Φ^T ecuația matriceală (15) devine:

$$\Phi^T y=\Phi^T\Phi\begin{bmatrix} w_{11}^0 \\ w_{12}^0 \\ b_1^0 \end{bmatrix} \tag{16}$$

în care

$$\Phi^T y=\begin{bmatrix} -103-97-93-87+103+97+113+107 \\ -103-97+93+87-103-97+113+107 \\ 103+97+93+87+103+97+113+107 \end{bmatrix}=\begin{bmatrix} 40 \\ 0 \\ 800 \end{bmatrix}$$

Observație: Elementul zero de pe poziția a doua a vectorului, indică faptul că y nu depinde de u_2, respectiv lipsește conexiunea w_{12}^0.

Membrul drept al ecuației (16) este de forma

$$\Phi^T\Phi=8\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}=8I_3$$

în care I este matricea unitate de ordinul 3 și deci:

$$\left[\Phi^T\Phi\right]^{-1}=\frac{1}{8}I$$

Din ecuația (16), rezultă inițializările pentru algoritmul BKP:

$$\begin{bmatrix} w_{11}^0 \\ w_{12}^0 \\ b_1^0 \end{bmatrix}=\left[\Phi^T\Phi\right]^{-1}\Phi^T y=\frac{1}{8}\begin{bmatrix} 40 \\ 0 \\ 800 \end{bmatrix}$$

$w_{11}^0=5, w_{12}^0=0, b_1^0=100$.

Vectorul valorilor obținute la ieșirea primului model neuronal parțial este

$$\hat{y}_{1k} = \Phi\left[w_{11}^0 \quad w_{12}^0 \quad b_1^0\right]^T = [95 \quad 95 \quad 95 \quad 95 \quad 105 \quad 105 \quad 105 \quad 105]^T$$

iar suma erorilor pătratice va fi

$$E_1^2 = \sum_{k=1}^{8}(y_k - \hat{y}_{1k})^2 = (103-95)^2 + (97-95)^2 + \cdots + (107-105)^2 = 2.72 \cdot 10^2$$

Se procedează similar pentru următoarele 5 modele neuronale parțiale, apoi se vor ordona modelele în ordinea crescătoare a erorii pătratice obținute, după care vor fi selectați primii cei mai buni 3 neuroni. Ținând însă cont și de conexiunile acestora cu stratul de intrare (modelele parțiale alese trebuie să acopere întreg stratul de intrare)rezultă rețeaua din figura 12. Pentru a putea aplica algoritmul BKP pornind de la ponderile calculate prin metoda celor mai mici pătrate, pentru structura din figura 12, trebuie calculate în prealabil inițializările pentru stratul de ieșire prin CMMP.

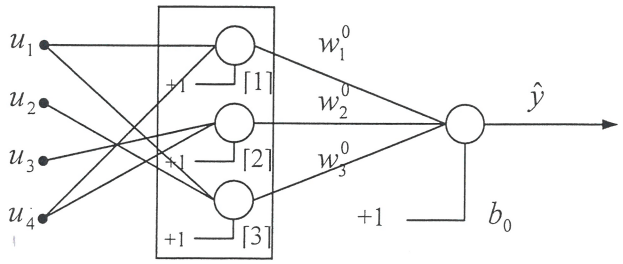


Figura 12 - Calculul ponderilor neuronului de ieșire pentru RN cu un singur strat ascuns

Intrările neuronului de ieșire cu ponderile w_1^0, w_2^0, w_3^0 și pragul b^0, sunt furnizate de ieșirile neuronilor din stratul ascuns, acestea fiind calculate anterior. Pe aceste date și y se calculează

$$\begin{bmatrix} w_1^0 \\ w_2^0 \\ w_3^0 \\ b^0 \end{bmatrix} = \left[\Phi_*^T \Phi_*\right]^{-1} \Phi_*^T y \qquad (17)$$

în care Φ_*^T este matricea de dimensiune 8x4, construită pe baza datelor (11).Pentru aplicarea metodei celor mai mici pătrate folosim același procedeu ca mai sus (liniarizăm funcția de activare F(x)=x). În acest mod, erorile pătratice obținute prin selectarea neuronilor din stratul ascuns, se redistribuie pe noile ponderi din stratul de ieșire. Având toate inițializările ponderilor găsite, se poate trece la aplicarea algoritmului BKP pentru structura rețelei neurale din figura 12.

7. Aplicarea algoritmului BKP pentru rețele neurale parțial conectate cu un singur strat ascuns

Pentru aplicarea algoritmului BP sunt necesare doua seturi de date, setul de antrenare, respectiv setul de verificare. Pentru antrenarea rețelei se folosesc datele (11) pe care s-a aplicat algoritmul evolutiv în vederea configurării primului strat ascuns, iar pentru testare, deoarece dispunem de puține date, vom genera setul de test pe baza celui de antrenare.

Un procedeu posibil este generarea datelor prin interpolare liniară:

$$u_i^*(k) = \frac{u_i(k) + u_i(k+1)}{2}$$

$$y^*(k) = \frac{y(k) + y(k+1)}{2}$$

În urma acestei metode de generare, va rezulta următorul set de verificare:

k	u_1^*	u_2^*	u_3^*	u_4^*	y^*
1	-1	-1	0	0	100
2	0	0	0	0	95
3	0	+1	0	0	90
4	0	0	0	-1	90
5	+1	-1	0	-1	100
6	+1	0	0	0	105
7	+1	+1	0	+1	110

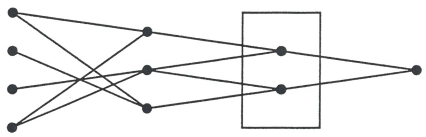
(18)

Dispunând de cele două seturi de date **(5.4)** şi **(5.8)** se aplică BP folosind ponderile şi activările calculate prin CMMP. După antrenarea reţelei din figura 5.5, rezultă la oprire o eroare pătratică E_I^2, noile valori ale ponderile vor fi:

$$\begin{bmatrix} w'_{11} & w'_{12} & w'_{13} & w'_{14} \\ w'_{21} & w'_{22} & w'_{23} & w'_{24} \\ b'_1 & b'_2 & b'_3 & b'_4 \end{bmatrix} \quad \text{stratul ascuns} \qquad \begin{bmatrix} w'_1 \\ w'_2 \\ w'_3 \\ b' \end{bmatrix} \quad \text{stratul de ieşire}$$

8. Continuarea căutării numărului optim de straturi în sensul erorii pătratice minime

În continuare, se procedează la testarea unei reţele cu două straturi ascunse, prezentată în figura 13, înghețând ponderile şi pragurile w' calculate la pasul precedent.



strat I strat II

Figura 13. Reţea neurală cu două straturi ascunse

În al doilea strat vom avea cel puţin doi neuroni, pe baza indicaţiilor furnizate de metoda evolutivă, care recomanda alegerea „celor mai bune modele neurale parţiale" în sensul erorii pătratice minime, dar totodată trebuie ca acestea să acopere toate intrările în strat.

Valorile ponderilor şi cele ale pragurilor se calculează tot prin metoda celor mai mici pătrate, intrările neuronilor din al doilea strat ascuns vor fi de aceasta dată ieşirile neuronilor din stratul ascuns precedent (ale cărui ponderi sunt „înghețate"). După estimarea iniţializărilor pentru stratul II, se trece la aplicarea algoritmului BP pe reţeaua din figura 13, obţinându-se pentru aceasta eroarea E_{II}^2 valabila pentru reţeaua cu doua straturi ascunse. Decizia de adoptare a uneia dintre cele două structuri de reţele neurale prezentate în figurile 12 şi 13 se va face pe baza comparării erorilor E_I^2, E_{II}^2. Dacă $E_I^2 < E_{II}^2$ se va alege structura prezentată în figura 12, altfel se va alege structura din figura 13. Pentru cazurile mai complexe, cu un număr mai mare de intrări (n>>4) pot rezulta reţele neurale cu două, trei straturi ascunse, procesul de căutare a structurii optime în sensul erorii pătratice minime conţinând, până când eroarea calculată pe ultimul strat N este mai mare decât cea provenită de la stratul precedent N-1. În acest caz se alege structura de reţea care are în componenta ei N-1 straturi.

9. Concluzii

Procedurile propuse pentru proiectarea optimală a structurii reţelelor neurale pentru modelare statică sunt în buna măsură originale, oferă o eşalonare a etapelor de proiectare, urmărind permanent un criteriu de optim. Condiţia impusă în ceea ce priveşte diferenţierea seturilor de antrenament şi de test în vederea aplicării algoritmului back propagation, oferă reţelei neurale o capacitate mai mare de generalizare

(recunoaşterea ulterioară de către reţea, a unor forme care nu fac parte din setul de instruire).Bibliografia folosită a fost în principal utilă pentru a constata că sunt unele probleme în instruirea pe seturi mici de date, optimizarea structurii reţelei neurale şi altele, nu sunt explicit tratate în literatura de specialitate. În majoritatea aplicaţiilor, autorii fixează pentru procesul de antrenare un număr maxim de epoci sau limitează antrenamentul până când eroarea pătratică ajunge la o valoare impusă. Asemenea procedee nu pot fi aplicate în practica industrială seturile de antrenament fiind mici.

Bibliografie

1. **DUMITRAŞ, A.**: Proiectarea reţelelor neurale artificiale, Ed. Odeon, Bucureşti, 1997.
2. **G. TODEREAN, M. COSTEIU**: Reţele neurale artificiale, Ed. Albastra, 1995.
3. **TERTIŞCO, M., GH. PETRACHE**: Identificarea proceselor, Ed. I.P.B, 1978.
4. **DUMITRESCU, D., H. COSTIN**: Reţele neurale, Ed. Teora, Bucureşti, 1996.
5. **DUMITRACHE, I., N. CONSTANTIN, M, DRAGOICEA**: Reţele neurale. Identificarea şi conducerea proceselor, Ed. Matrix Rom, 1999.