METODE DE ESTIMARE A ERORILOR ÎN APROXIMAREA FUNCȚIILOR

Liviana Tudor

LTudor@upg-ploiesti.ro

Departamentul Informatică, Tehnologia Informației, Matematică şi Fizică

Adrian Moise

AMoise@upg-ploiesti.ro

Departamentul Automatică, Calculatoare şi Electronică

Universitatea Petrol-Gaze din Ploieşti

Rezumat: Articolul prezintă două metode de estimare a erorilor în aproximarea funcțiilor, folosind abordarea numerică şi cea neuronală. Metoda numerică de aproximare a unei funcții foloseşte interpolarea cu polinoame Lagrange şi descrie eroarea absolută de interpolare. Abordarea neuronală demonstrează proprietatea de aproximare universală a rețelelor feed-forward şi prezintă modul de propagare a erorilor neuronale în timpul procesului de învățare. Rezultatele experimentale evidențiază superioritatea sistemelor inteligente neuronale în rezolvarea problemelor de aproximare a unei funcții.

Cuvinte cheie: erori, aproximarea funcțiilor, metode numerice, polinom Lagrange, aproximație neuronală, algoritm Backpropagation, rețea neuronală feed-forward.

Abstract: This paper presents two methods to estimating errors in functions approximation, using the numerical and neural approach. The numerical method in function approximation uses Lagrange polynomials interpolation and describes the absolute interpolation error. The neural approach proves the property of a feed-forward artificial network to be a universal approximator and presents the neural errors propagation during the training process. The experimental results highlight the advantages of the intelligent neural systems in solving problems of function approximations.

Keywords: errors, function approximation, numerical methods, Lagrange polynomial, neural approximation, Backpropagation algorithm, feed-forward neural network.

1. Introducere

Acest articol abordează tema aproximării funcțiilor folosind metode numerice şi neuronale şi analizează comparativ câteva tehnici de estimare, propagare şi minimizare a erorilor.

Erorile generate în rezolvarea problemelor matematice pot proveni din diverse surse precum: formularea problemei, metoda folosită, valorile inițiale ale variabilelor sau metodele de rotunjire şi trunchiere. Pentru simplificarea unei probleme complexe, se poate alege o metodă de rezolvare aproximativă care poate determina apariția unor erori. Incertitudinile legate de aproximarea valorilor pot influența valorile erorilor propagate.

Erorile de rotunjire şi de trunchiere ale problemelor care necesită aproximări numerice sunt generate de reprezentarea numerelor cu un număr dat de cifre semnificative exacte şi pot depinde de modul de codificare internă a datelor în memoria calculatorului şi de numărul de iterații necesare pentru aproximarea soluției cu eroare minimă. Propagarea erorilor de rotunjire este determinată de creşterea numărului de operații iterative ale problemei. Reducerea erorilor de trunchiere reprezintă obiectivul principal al metodelor numerice de calcul.

În general, erorile întâlnite în aproximările numerice pot fi de tip absolut şi relativ. Pentru exprimarea erorilor de reprezentare a numerelor aproximative, vom considera un număr $y \in R$ şi $y' \in R$ o aproximație a lui y. Eroarea absolută (respectiv relativă) a aproximației y', notată E_{abs} (respectiv E_{rel}), verifică relația [1]:

$$|y - y'| \leq E_{abs} \tag{1}$$

$$E_{rel} = \frac{E_{abs}}{|y'|},\ y' \neq 0$$

În probleme de calcul numeric se determină valoarea aproximativă y' folosind valoarea minimă a erorii E_{abs} care satisface relația (1) sau a erorii E_{rel} din relația (2). Astfel, numărul y poate fi reprezentat aproximativ sub forma:

$$y = y' \pm E_{abs} = y' \pm E_{rel} \cdot |y'| = y' \cdot (1 \pm E_{rel}) \tag{2}$$

Erorile de aproximare a unei funcții: Se consideră o funcție reală de o variabilă reală $f : R \rightarrow R$ şi se presupune că se doreşte calcularea valorii $y = f(x)$ dar se obține o valoare aproximativă y'. Eroarea forward absolută (respectiv relativă) de aproximare a funcției f se exprimă [2]:

$$E_{Fabs} = |y - y'| \tag{3}$$

$$E_{Frel} = \frac{E_{Fabs}}{y} = \frac{|y - y'|}{y}.$$

Analiza erorii de aproximare a funcției f din perspectiva erorilor backward ia în considerare valoarea de perturbație astfel: valoarea aproximativă y' este considerată soluție exactă a problemei modificate, $y' = f(x')$ unde x' este perturbația lui x. Eroarea backward absolută (respectiv relativă) de aproximare a funcției f se exprimă [2]:

$$E_{Babs} = |x - x'|, \text{ unde } y = f(x) \text{ şi } y' = f(x'), \tag{4}$$

$$E_{Brel} = \frac{E_{Babs}}{x} = \frac{|x - x'|}{x}.$$

Pentru problema aproximării funcției f se vor analiza comparativ metoda de interpolare polinomială şi metoda de aproximare neuronală folosind rețele neuronale artificiale de tip feedforward.

Articolul este structurat după cum urmează: Secțiunea *Aproximarea unei funcții prin interpolare* prezintă metoda de interpolare folosind polinomul lui Lagrange. Secțiunea *Aproximarea neuronală* descrie aproximarea unei funcții folosind o rețea neuronală artificială antrenată cu algoritmul Backpropagation şi modul de propagare a erorilor în timpul procesului de învățare neuronală. Secțiunea *Studiu de caz. Aproximarea unei funcții reale* descrie rezolvarea problemei pornind de la un set de valori, folosind polinomul Lagrange şi o rețea neuronală feedforward şi analizează comparativ erorile şi valorile aproximate.

2. Aproximarea unei funcții prin interpolare

Aproximarea unei funcții este necesară dacă nu se cunoaşte expresia analitică, dar se dau valorile funcției într-un număr finit de puncte dintr-un interval dat. De asemenea, aproximarea este utilă când expresia analitică a funcției este prea complicată şi calculele efectuate sunt dificile. Una dintre metodele numerice de aproximare a unei funcții prin interpolare foloseşte polinoamele Lagrange şi va fi schițată în continuare.

2.1 Interpolarea Lagrange

Se consideră o funcție de clasă C^1 de o variabilă reală $f : [a,b] \rightarrow R$ dată prin $n+1$ puncte de tabelare:

$$f(x_i) = y_i,\ i = 0,1,...,n \tag{5}$$

Aproximarea funcției f prin interpolare presupune găsirea unei funcții de interpolare g care să aproximeze funcția f pe intervalul $[a,b]$ și să satisfacă relația:

$$g(x_i) = y_i,\ i = 0,1,...,n \tag{6}$$

Dintre funcțiile care îndeplinesc condiția (6), trebuie aleasă funcția g care asigură o aproximare cu eroare minimă. Problema aproximării prin interpolare are soluție unică dacă funcția g se caută în spațiul polinoamelor generalizate [3]. Căutarea funcției g echivalează cu găsirea unui polinom de interpolare P_n care verifică relația (6). Se obțin astfel $n+1$ polinoame Lagrange de formă particulară, notate L_i, $i = 0,1,...,n$, cu gradul n și rădăcinile $x_0, x_1,..., x_{i-1}, x_{i+1},..., x_n$:

$$L_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j},\ i = 0,1,...,n \tag{7}$$

Forma Lagrange a polinomului de interpolare P_n asociat punctelor (x_i, y_i) $i = 0,1,...,n$ poate fi exprimată:

$$P_n(x) = \sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{8}$$

În continuare, vom prezenta procedura pseudocod pentru aproximarea valorii funcției f într-un punct $k \in [a,b]$ folosind interpolarea Lagrange:

```
procedure Lagrange (n, k)
array x(n+1), y(n+1)
Pn = 0
for i = 0, n
        t = y(i)
        for j = 0, n
                if j <> i then t = t * (k – x(j))/ (x(i) – x(j))
                endif
        endfor
        Pn = Pn + t
endfor
write ('f(k) ~ ',Pn)
return
end
```

Polinomul Lagrange P_n aproximează funcția f în punctul $k \in [a,b]$, cu o eroare absolută de interpolare: $e(k) = | f(k) - P_n(k) |$.

2.2 Eroarea de interpolare

Dacă funcția f este de clasă C^{n+1}, eroarea absolută de interpolare cu care polinomul P_n aproximează funcția f se calculează:

$$e(x) = | f(x) - P_n(x) |,\ x \in [a,b]. \tag{9}$$

Eroarea maximă de interpolare depinde de derivata de ordin $n+1$ (numărul de puncte de tabelare):

$$|e(x)| \le \frac{\sup_{t\in[a,b]} |f^{(n+1)}(t)|}{(n+1)!} \prod_{j=0}^{n} |x - x_j| \tag{10}$$

În continuare, vom prezenta o tehnică neuronală de aproximare a unei funcții, metoda de estimare şi propagare a erorii de aproximare.

3. Aproximarea neuronală

Proprietatea rețelelor neuronale artificiale (RNA), cu cel puțin două straturi, de aproximare a funcțiilor are o importanță deosebită pentru aplicațiile practice [4], [6]. Rezultatul privitor la aproximarea unei funcții [7] menționează că orice funcție f de clasă C^1 poate fi aproximată suficient de bine pe o mulțime compactă cu ajutorul unei rețele neuronale cu două straturi ale cărei ponderi sunt determinate în mod corespunzător.

3.1 Aproximarea unei funcții folosind rețele neuronale feed-forward

Fie $f:[a,b] \to R$ o astfel de funcție de clasă C^1 . Dacă se dă o mulțime compactă $S \subset [a,b]$ şi un număr pozitiv ε_N, există o rețea neuronală de tipul feed-forward cu două straturi astfel încât:

$$f(x) = W^T \sigma(V^T x) + \varepsilon \tag{11}$$

unde $\|\varepsilon\| < \varepsilon_N$ pentru orice $x \in S$ şi pentru un număr L (suficient de mare) de neuroni în stratul ascuns. V este matricea ponderilor dintre stratul de intrare şi stratul ascuns iar W este matricea ponderilor dintre stratul ascuns şi stratul de ieşire. Valoarea ε (în general, depinde de x) se numeşte eroarea de aproximare a funcției şi descreşte pe măsură ce dimensiunea L a stratului ascuns creşte. Se spune că, pe mulțimea compactă S, $f(x)$ este în interiorul gamei funcționale ε_N a rețelei neuronale. Pentru scopurile acestui articol, se va considera mulțimea

$$S = \{x_0, x_1, ..., x_n\} \tag{12}$$

Rezultate relativ recente [7] arată cât de mare trebuie să fie L pentru a se obține o anumită precizie ε_N şi o mulțime compactă dată S; pe măsură ce S devine mai mare, numărul L creşte în mod corespunzător.

Problema în determinarea ponderilor astfel încât rețeaua să aproximeze o funcție dată f suficient de bine nu este uşor de rezolvat. Vom arăta cum se pot obține aceste ponderi cu ajutorul algoritmului Backpropagation de propagare inversă a erorii.

Paşii algoritmului BackPropagation (de antrenare a rețelei feed-forward prin propagarea inversă a erorii) pot fi descrişi [5], [9], [11]:

1. inițializarea ponderilor rețelei şi a setului de antrenare. Se foloseşte, în general, inițializarea aleatoare a ponderilor rețelei cu valori mici;

2. aplicarea unui vector de intrare;

3. propagarea directă a semnalului de intrare şi calculul ieşirilor rețelei;

4. calcul eroare prin compararea ieşirilor rețelei cu ieşirile țintă (dorite);

5. propagarea erorii înapoi prin rețea (începând de la stratul de ieşire, spre straturile ascunse), pentru corecția ponderilor;

6. dacă actualizările ponderilor sunt nesemnificative (eroarea este mai mică decât un prag) sau dacă s-au terminat seturile de antrenare, se opreşte algoritmul; altfel se reia de la pasul 2.

De multe ori este dificil să se aleagă funcțiile de activare astfel ca ele să reprezinte o bază (set de antrenare). Această problemă se poate rezolva prin selectarea în mod aleator a matricei V. S-a arătat în [7] că, pentru aceste rețele, funcția $\phi(x) = \sigma(V^T x)$ reprezintă o bază, astfel că rețeaua are proprietatea de aproximator universal.

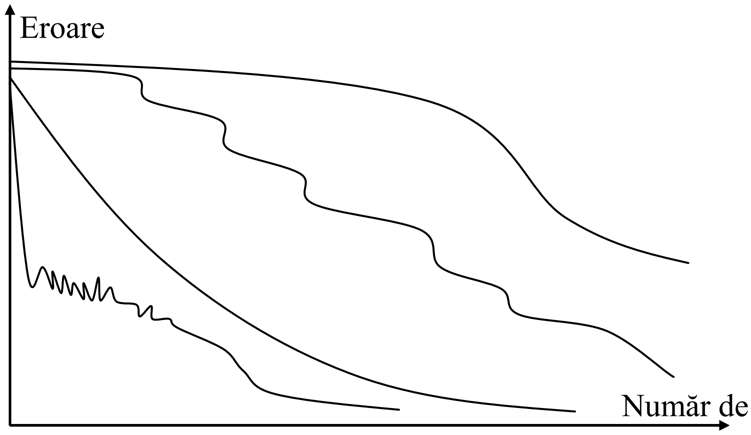

Figura 1. Diferite traiectorii ale măsurii erorii

3.2 Propagarea erorilor neuronale

Măsura erorii şi traiectoria erorii se folosesc în general pentru a conduce şi a evalua antrenarea. Există diverse măsuri ale erorii [10], iar traiectorii tipice sunt prezentate în Figura 1, care reprezintă un scenariu optimist în care eroarea descreşte.

Pentru a formaliza răspunsul în cazul unei rețele neuronale de tip feed-forward, se defineşte vectorul eroare de ieşire pentru a p -a pereche de forme din setul de antrenare H prin diferența dintre răspunsul dorit (vectorul țintă) $\underline{t}^p$ şi răspunsul curent $\underline{o}^p$ obținut prin aplicarea intrării p din setul de antrenare:

$$\underline{e}^p = \underline{t}^p - \underline{o}^p \,. \tag{13}$$

O măsură a erorii de ieşire pe baza celui de-al p -lea eşantion de antrenare se notează cu E^p şi se defineşte prin

$$E^p = \frac{1}{2}(\underline{e}^p)^T \cdot \underline{e}^p = \frac{1}{2}\left\|\underline{e}^p\right\|^2 . \tag{14}$$

Alegerea normei erorii, deşi aparent superficială, este totuşi consistentă cu alte numeroase formalisme din literatură [8]. Totuşi, este posibil să se utilizeze şi alte norme, inclusiv norme ponderate cum este $\left\|\underline{e}\right\|_R$, unde R este o matrice pozitiv definită, sau norme precum $\max\{e_i\}$, unde e_i este al i -lea element din $\underline{e}$.

Pentru a determina sensibilitatea erorii în algoritmii de antrenare Delta Rule (DR) şi Generalized Delta Rule (GDR) se foloseşte regula de derivare a funcțiilor compuse pentru a calcula derivata lui E^p în raport cu w_{ji} :

$$\frac{\partial E^p}{\partial w_{ji}} = \frac{\partial E^p}{\partial o_j^p} \cdot \frac{\partial o_j^p}{\partial w_{ji}} \tag{15}$$

unde, din ecuațiile (14) şi (15),

$$E^p = \frac{1}{2}\left\| \underline{t}^p - \underline{o}^p \right\|^2 = \frac{1}{2}\sum_j (t_j^p - o_j^p)^2 \tag{16}$$

Este posibil să se dezvolte o formulare alternativă prin luarea în considerare a erorii globale (pentru toate epocile):

$$E = \sum_p E^p \,. \tag{17}$$

Pentru neuroni de tip Weighted Linear Input Combination (WLIC), activarea neuronului artificial pentru unitatea j este dată de suma ponderată a intrărilor în unitatea j:

$$net_j = \sum_i w_{ji} \cdot i_i \tag{18}$$

sau, în cazul în care se include şi o intrare de polarizare,

$$net_j = \sum_i w_{ji} \cdot i_i + bias_j \,. \tag{19}$$

i_i este a i-a intrare în unitatea j iar f se alege o funcţie sigmoid, adică:

$$f(net_j) = \frac{1}{1+e^{-net_j}} \,. \tag{20}$$

Membrul drept din ecuaţia (16) reprezintă schimbarea incrementală în E^p datorată schimbării incrementale în ponderea w_{ji} sau, sensibilitatea erorii E^p la w_{ji}. $\partial E^p / \partial o_j^p$, reprezintă efectul asupra lui E^p datorat celei de-a j-a ieşire, în timp ce $\partial o_j^p / \partial w_{ji}$, măsoară schimbarea în o_j^p ca funcţie de ∂w_{ji}. Sensibilitatea este reprezentată în Figura 2.

Din ecuaţia (17), se poate obţine uşor $\partial E^p / \partial o_j^p$ pentru unităţile din stratul de ieşire:

$$\frac{\partial E^p}{\partial o_j^p} = -(t_j^p - o_j^p) = -e_j^p \tag{21}$$

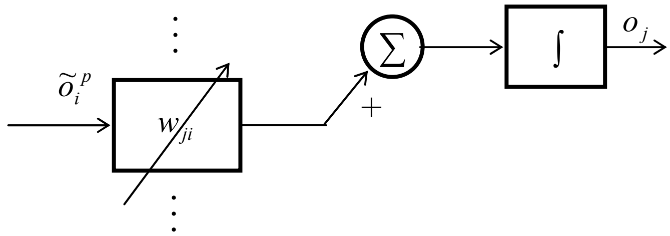

Figura 2. Sensibilitatea lui o_j la ponderea w_{ji}

Să presupunem că $f(net_j) = k \cdot net_j$. Pentru simplitate, vom considera $k = 1$. Se obţine $o_j^p = \sum_i w_{ji} \cdot i_i^p$ şi $\frac{\partial o_j^p}{\partial w_{ji}} = i_i^p$. Aşadar, ecuaţia (19) devine

$$\frac{\partial E^p}{\partial w_{ji}} = (-e_j^p) \cdot (i_i^p) \tag{22}$$

care este o relaţie conformă cu strategia de variaţie a ponderilor de tip produs. Pentru a verifica faptul că această strategie minimizează eroarea E observăm că, din ecuația (18) se obține

$$\frac{\partial E}{\partial w_{ji}} = \sum_p \frac{\partial E^p}{\partial w_{ji}}. \tag{23}$$

4. Studiu de caz. Aproximarea unei funcții reale

Se va prezenta modul de estimare a erorilor şi a aproximărilor valorilor unei funcții a cărei expresie nu se cunoaşte. Pentru studiul experimental, se va alege un set de puncte de tabelare furnizate grafic (cu mouse-ul pe suprafața grafică). Aproximarea se realizează în mediul de programare Matlab folosind metoda de interpolare Lagrange şi tehnica neuronală.

4.1 Metodologia de calcul

Algoritmul Lagrange foloseşte ca date de intrare n puncte de tabelare: (x_i, y_i), $i = 1,...,n$ şi valoarea k în care se aproximează funcția f. Algoritmul de interpolare Lagrange aproximează o singură valoare a funcției pentru valoarea k.

Pentru eficiența comparării celor două metode şi acuratețea rezultatelor, se va folosi o diviziune a intervalului în care se află valorile vectorului x, cu pasul $\frac{\max(x) - \min(x)}{100}$. În punctele diviziunii, se va aproxima funcția folosind algoritmul de interpolare Lagrange. Eroarea de aproximare prin metoda Lagrange va fi calculată ca o medie a erorilor absolute obținute la fiecare pas al diviziunii. Se va lua în considerare pentru comparare şi eroarea medie pătratică.

Pentru aproximarea neuronală, se foloseşte o rețea feed-forward cu arhitectura din Figura 3. Rețeaua foloseşte 2 straturi funcționale de neuroni (primul strat are 4 neuroni, iar al doilea strat are 1 neuron). Intrarea rețelei este vectorul $x = (x_1,...,x_n)$, $i = 1,...,n$, cu punctele în care se cunosc valorile funcției $f(x_i) = y_i$, $i = 1,...,n$.

Eroarea rețelei neuronale se calculează ca o eroare medie pătratică a diferențelor între valorile exacte şi cele aproximate la simularea funcționării rețelei.

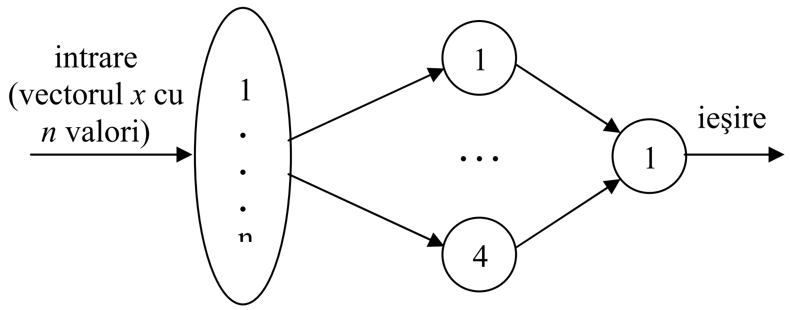

Figura 3. Rețea feed-forward cu 2 straturi funcționale

4.2 Comparații

Pentru exemplificare, se consideră o diviziune a intervalului [0, 1] şi punctele de tabelare:

x = [0.1187, 0.1601, 0.1901, 0.2339, 0.2892, 0.3191, 0.3698, 0.4136, 0.4597, 0.5127, 0.5726, 0.6094, 0.6256, 0.6394, 0.6671, 0.6855, 0.7177, 0.7385, 0.7638, 0.7892, 0.8306, 0.8606, 0.8790, 0.9136, 0.9528] şi y = [0.0892, 0.1944, 0.2266, 0.2558, 0.2763, 0.2851, 0.2763, 0.2705, 0.2529, 0.2383, 0.2412, 0.2763, 0.3289, 0.3670, 0.4137, 0.4576, 0.5190, 0.5658, 0.5921, 0.6126, 0.6155, 0.6155, 0.6067, 0.5921, 0.5863].

La diverse teste experimentale, se observă că metoda Lagrange generează o aproximare cu variații foarte mari ale valorilor punctelor diviziunii, obținându-se o eroare medie err_Lagrange1 = 3.6522e+003 şi o eroare medie pătratică err_Lagrange2 = 3.9720e+008. În exemplul de mai jos (Figura 4a şi 4b), rețeaua neuronală realizează o aproximare a funcției cu eroarea medie pătratică: err_RNA = 0.0181.

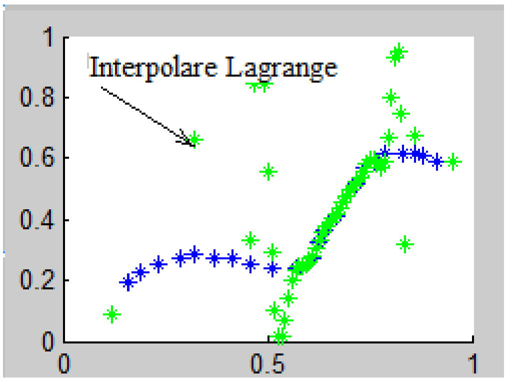

Figura 4a. Aproximare prin interpolare

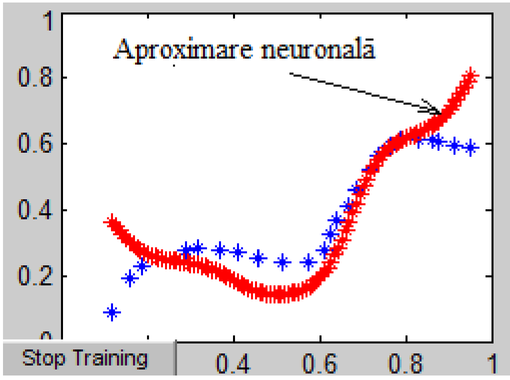

Figura 4b. Aproximare neuronală

Metoda neuronală furnizează rezultate dependente de eficiența învățării rețelei, iar eroarea medie pătratică obținută este mai mică decât cea generată de metoda numerică.

5. Concluzii

Acest articol prezintă comparativ două metode de aproximare a unei funcții reale: metoda numerică de aproximare prin interpolare Lagrange şi metoda de aproximare neuronală folosind rețele neuronale feed-forward antrenate cu algoritmul Backpropagation. Studiul metodelor de estimare şi propagare a erorilor de aproximare a unei funcții evidențiază complexitatea şi complementaritatea tehnicilor numerice şi neuronale. Tehnicile numerice furnizează uneori rezultate exacte, alteori rezultate cu erori mari, pe când metodele neuronale nu generează rezultate exacte, însă sunt instrumente foarte eficiente de aproximare numerică.

BIBLIOGRAFIE

1. **BEU, T.:** Calcul numeric în Turbo Pascal, Editura Micro-Informatica, Cluj-Napoca, 1992, 203 p.
2. **BUNECI, MĂDĂLINA ROXANA:** Metode Numerice - aspecte teoretice şi practice, Editura Academica Brâncuşi, Târgu-Jiu, 2009, 284 p.
3. **CIUPRINA, GABRIELA:** Suport didactic pentru disciplina Algoritmi numerici, Interpolarea funcțiilor, Universitatea "Politehnica" Bucureşti, 2012
4. **CYBENKO, G.:** Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals and Systems, vol. 2, no. 4, 1989
5. **DUMITRESCU D.:** Costin Hariton, Rețele neuronale. Teorie şi aplicații, 1996, Editura Teora
6. **HORNIK, K.; STINCHOMBE, M.:** Multilayer feedforward networks are universal approximators, Neural Networks, vol. 2, 1989.
7. **LEWIS, F. L.; JAGANNATHAN, S.; YESILDIREK, A.:** Neural Network Control of Robot Manipulators and Nonlinear Systems, Taylor and Francis, 1999.

8. **MOISE, A.:** Rețele neuronale pentru conducerea roboților, MatrixRom, Bucureşti, 2012, 217p.
9. **MOISE, A.:** Rețele neuronale pentru recunoaşterea formelor, MatrixRom, Bucureşti, 2005, 310 p.
10. **SCHALKOFF, R. J.:** Artificial Neural Networks, McGraw-Hill International Ed., New York, 1997.
11. **TUDOR, NICOLETA LIVIANA:** Rețele neuronale artificiale. Aplicații Matlab**,** Editura MATRIX ROM Bucureşti, 2012, 195 p.