UTILIZAREA REȚELELOR NEURONALE ÎN SIMULAREA CIRCUITELOR LOGICE COMBINAȚIONALE

Conf.univ.dr. Remus Joldeş
Conf.univ.dr. Ioan Ileană

1. INTRODUCERE

Calculatorul tradițional nu mai reuşeşte întotdeauna să facă față problemelor ce necesită calcul intensiv cum ar fi recunoaşterea formelor, controlul mişcării roboților, luarea deciziilor pe baza unor cantități mari de date cu zgomot etc., astfel încât s-au abordat şi alte metode de prelucrare a informațiilor, printre care prelucrarea distribuită [7].

Una dintre aceste direcții neconvenționale de prelucrare a informațiilor, care respectă multe din cerințele formulate în secțiunea precedentă, o constituie rețelele neuronale (calcul neuronal, conexionism). În contrast cu maşina Von Neumann care execută un program scris pe baza unui algoritm, rețelele neuronale învață prin exemple. Rezultatul învățării nu este un cod ci o reprezentare distribuită a informației. Reprezentarea distribuită şi calculul local caracteristic rețelelor neuronale micşorează complexitatea elementelor de calcul dar măreşte volumul conexiunilor dintre ele.

În ultimul deceniu , domeniul rețelelor neuronale a cunoscut o deosebită expansiune, practic existând aplicații în toate domeniile vieții sociale. Lucrarea de față abordează o categorie de aplicații referitoare la simularea, cu ajutorul rețelelor neuronale, a unor sisteme dinamice, în cazul de față, circuite logice combinaționale (CLC).

Secțiunea 2 prezintă pe scurt noțiuni legate de CLC, iar secțiunea 3 ilustrează un exemplu de simulare, pentru decodificatorul BCD- 7 segmente.

2. CIRCUITE LOGICE COMBINAȚIONALE

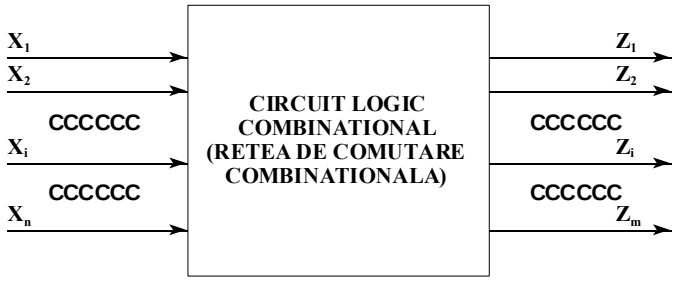
Un circuit logic combinațional (CLC) se defineşte ca un circuit electronic cu n intrări, pe care le vom nota cu $X_1, X_2, \ldots, X_n$, şi m ieşiri, notate cu $Z_1, Z_2, \ldots, Z_m$, pentru care ieşirile pot fi exprimate în funcție de intrări prin intermediul unui model matematic de forma:

$$Z_1 = f_1(X_1, X_2, \ldots, X_n)$$
$$Z_2 = f_2(X_1, X_2, \ldots, X_n)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$Z_I = f_i(X_1, X_2, \ldots, X_n)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$Z_m = f_m(X_1, X_2, \ldots, X_n)$$

Dacă notăm cu $X=\{ X_1, X_2, \ldots, X_n \}$ mulțimea variabilelor de intrare şi cu $Z=\{ Z_1, Z_2, \ldots, Z_m \}$ mulțimea variabilelor de ieşire, atunci un circuit logic combinațional se poate simplu descrie matematic prin tripletul:

$$CLC = (X, Z, F)$$

în care F este funcția de intrare-ieşire $F : X \rightarrow Z$ este independentă de timp.



În realizarea (sinteza) unui circuit CLC se începe în general, prin clasificarea condițiilor de funcționare, în conformitate cu cerințele impuse printr-un tabel de adevăr şi prin specificarea stărilor de operare şi respectiv neoperare. O condiție de operare ori de neoperare se referă la starea mărimilor de ieşire $Z_1, Z_2, \ldots, Z_m$ şi de intrare $X_1, X_2, \ldots, X_n$. Aatfel vectorul variabilelor de intrare poate lua valorile $\forall_1, \forall_2, \ldots, \forall_n$ ori valorile $\exists_1, \exists_2, \ldots, \exists_n$ care aparțin mulțimii {0, 1}.

O condiție de operare este de forma:

$$\nu(\forall_1, \forall_2, \ldots, \forall_n) = 1$$

iar o condiție de neoperare este exprimată de relația matematică:

$$\nu(\forall_1, \forall_2, \ldots, \forall_n) = 0.$$

Problema sintezei circuitului CLC este incompatibilă, dacă aceeaşi stare, apare în acelaşi timp, atât într-o condiție de operare cât şi într-una de neoperare, adică:

$$\nu(\forall_1, \forall_2, \ldots, \forall_n) = 1$$
$$\nu(\forall_1, \forall_2, \ldots, \forall_n) = 0.$$

Dacă însă nu intervin asemenea situații, avem de-a face cu o problemă compatilă, care poate fi determinată sau nedeterminată.

O problemă de acest tip este compatibil determinată dacă numărul condițiilor de operare şi neoperare este egal cu 2^n.

Sistemul este compatibil nedeterminat, având gradul de nedeterminare r, dacă numărul condițiilor de operare notat cu q, si numărul condițiilor de nedeterminare notat cu p, respectă relația:

$$q + p < 2^n$$

şi gradul de nedeterminare r se calculează cu relația:

$$r = 2^n - (q + p).$$

Cele r stări nedeterminate, pe care le notăm cu $8_1, 8_2, \ldots, 8_r$, pot primi valorile mulțimii {0, 1}. Atribuirea uneia din valorile 0 sau 1 depinde, în minimizarea clasică, de realizarea minimizării sau nu. Cu cele r soluții nedeterminate se pot obține 2^r soluții care satisfac condițiile problemei.

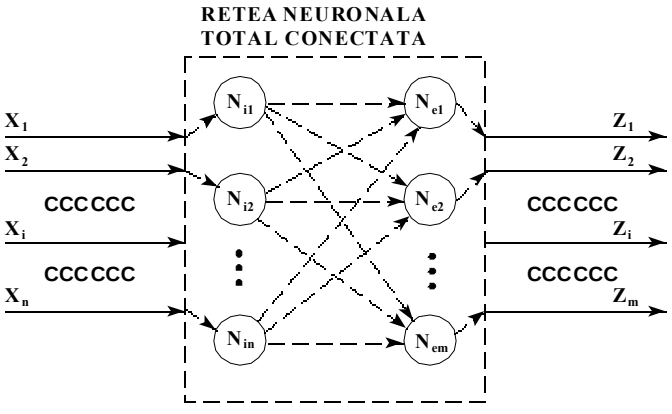
În sinteza unui CLC intervin următoarele etape:

- enunțul problemei;
- întocmirea tabelei de adevăr;
- minimizarea funcției de adevăr plecând de la forma analitică a acesteia sau de la tabelul de adevăr care o descrie;
- minimizarea corelată a funcțiilor de comutație;
- analiza schemei şi eliminarea hazardului;
- implementare hardware a funcțiilor logice cu elemente de comutație discretă.

O problemă delicată, care se ridică în cazul abordării clasice a sintezei circuitelor CLC, este faptul că nu întotdeauna se ajunge la o schemă absolut riguros optimă. Astfel, în cazul sintezei funcțiilor complexe cu număr mare de variabile de intrare (cele care au n>6), cu multe ieşiri şi stări nedeterminate, metodele algebrice respectiv topologice sunt foarte greu de aplicat.

3. SIMULAREA CLC CU AJUTORUL REȚELELOR NEURONALE FEED-FORWARD

Lucrarea de față propune o tratare specială a acestor CLC-uri prin utilizarea rețelelor neuronale. Soluția a fost sugerată de similitudinea funcțională existentă între un CLC şi o rețea neuronală cu n intrări şi m ieşiri. În fapt s-a utilizat o rețea neuronală cu două straturi, primul strat



având n neuroni de intrare şi al doilea m neuroni de ieşire. Pentru antrenarea rețelei neuronale se foloseşte setul de date de intrare şi setul de ieşire dat de tabelul de adevăr:

L_1	L_2	. . . .	L_i	. . . .	L_n	F
0	0	. . . .	0	. . . .	1	F_1
0	0	. . . .	0	. . . .	1	F_2
. . .	. . .	. . . .	. . .	. . . .	. . .	. . .
0	0	. . . .	1	. . . .	1	F_i
. . .	. . .	. . . .	. . .	. . . .	. . .	. . .
1	1	. . . .	1	. . . .	1	F_2n

în care există 2^n linii, din care rezultă 2^n valori pentru funcția logică ce va descrie funcționarea circuitului combinațional .

Circuitul a cărui funcționare s-a simulat a fost un decodificator BCD-7 segmente (7446). În cursul antrenării s-a găsit o configurație optimală de forma 4:9:7. Rezultatele antrenării cu cele 16 configurații de intrare sunt ilustrate în tabelul 1. Indicele n indică ieşirea reală a rețelei.

Tabelul 1

In	an	a	bn	b	cn	c	dn	d	en	e	fn	f	gn	g
1)	0,006	0,000	0,000	0,000	0,004	0,000	0,000	0,000	0,000	0,000	0,003	0,000	0,992	1,000
2)	0,992	1,000	0,000	0,000	0,000	0,000	0,999	1,000	0,999	1,000	0,997	1,000	0,999	1,000
3)	0,001	0,000	0,003	0,000	0,999	1,000	0,000	0,000	0,000	0,000	0,999	1,000	0,004	0,000
4)	0,003	0,000	0,002	0,000	0,994	1,000	0,000	0,000	0,998	1,000	1,000	1,000	0,001	0,000
5)	0,993	1,000	0,000	0,000	0,002	0,000	0,998	1,000	0,997	1,000	0,000	0,000	0,001	0,000
6)	0,000	0,000	0,994	1,000	0,006	0,000	0,002	0,000	0,999	1,000	0,000	0,000	0,000	0,000
7)	0,999	1,000	0,995	1,000	0,000	0,000	0,002	0,000	0,002	0,000	0,005	0,000	0,000	0,000
8)	0,007	0,000	0,008	0,000	0,001	0,000	0,995	1,000	0,996	1,000	0,995	1,000	0,995	1,000
9)	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,005	0,000
10)	0,004	0,000	0,003	0,000	0,000	0,000	0,995	1,000	0,999	1,000	0,001	0,000	0,003	0,000
11)	0,997	1,000	0,999	1,000	0,999	1,000	0,000	0,000	0,000	0,000	0,994	1,000	0,000	0,000
12)	0,995	1,000	0,999	1,000	0,004	0,000	0,001	0,000	0,997	1,000	0,999	1,000	0,000	0,000
13)	1,000	1,000	0,004	0,000	0,994	1,000	0,999	1,000	0,998	1,000	0,000	0,000	0,003	0,000
14)	0,005	0,000	0,999	1,000	0,994	1,000	0,005	0,000	0,999	1,000	0,004	0,000	0,002	0,000
15)	0,998	1,000	1,000	1,000	0,999	1,000	0,000	0,000	0,002	0,000	0,000	0,000	0,000	0,000
16)	0,994	1,000	0,992	1,000	0,998	1,000	0,996	1,000	0,997	1,000	0,999	1,000	0,995	1,000

4. CONCLUZII

În concluzie, metodologia de simulare a circuitelor combinaționale, cu ajutorul rețelelor neuronale, necesită respectarea următorilor paşi:

Pasul 1. Întocmirea tabelului de adevăr care descrie evoluția circuitului combinațional ales;

Pasul 2. Alegerea unei rețele neuronale cu trei straturi care să posede **n** neuroni pe primul strat, **n/2** neuroni în stratul ascuns şi un neuron în stratul de ieşire, unde **n** este numărul variabilelor de intrare;

Pasul 3. Antrenarea rețelei neuronale folosind ca date de intrare şi de ieşire fiecare linie a tabelului de adevăr;

Pasul 4. Verificarea corectei funcționări a rețelei, inclusiv a toleranței acesteia la eventualele zgomote;

Pasul 5. Implementarea hardware a rețelei care va juca rolul de circuit logic combinațional proiectat.

Bibliografie

1. G. M. Ştefan, I. V. Drăghici, T. Mureşan, E. Barbu – *Circuite integrate digitale*, Ed. Didactică şi Pedagogică, Bucureşti, 1983.
2. D. Dumitrescu, H. Costin – *Reţele neuronale. Teorie şi aplicaţii*, Ed. Teora, Bucureşti, 1996.
3. R. Joldeş – *Reţele neuronale. Noţiuni de bază . Implementări*, Seria Didactica, Universitatea "1 Decembrie 1918", Alba Iulia, 1995.