UTILIZATION OF NEURAL NETWORKS IN THE SIMULATION OF COMBINATIONAL LOGICAL CIRCUITS

by
Remus Joldeş, Ioan Ileană, Corina Rotar

Abstract: The paper presents an application of feed-forward neural networks: the simulation of combinational logical circuits (CLC). In the first part a presentation of main problems of CLC design is displayed, followed, in the second part by an example of simulating such circuits by artificial neural networks.

1. INTRODUCTION

The traditional calculator doesn`t always manage to face the problems that require intensive calculus such as pattern recognition, robots` movement control, decisions` taking based on a large quantity of data with noise etc. As a result, other methods of information processing were approached and distributed processing is one of them.

One of these non-conventional direction of information processing, that meets many of the above requirements is represented by neural networks (neural calculus). As opposed to Von Neumann machine that executes a written program based on an algorithm, neural networks learn from examples. The result of the learning is not a code but a distributed representation of the information. The distributed representation and local calculus characteristic to neural networks reduce the complexity of the calculus elements but increase the volume of the connexions between them.

In the last decade, the domain of neural networks expanded, existing practically applications in all social life domains. The present paper approaches a category of applications with reference to the simulation, with the help of neural networks, of some dynamic systems, in this situation – combinational logical circuits (CLC). Section 2 presents notions about CLC and section 3 illustrates an example of simulation, for the decoding BCD- 7 segments.

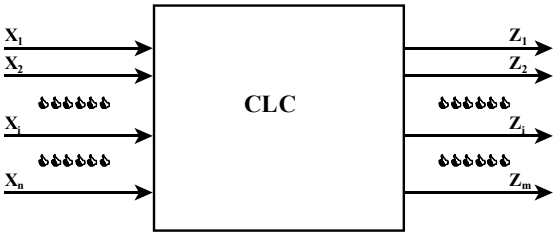
2. COMBINATIONAL LOGICAL CIRCUITS

A combinational logic circuit (CLC) is an electronic circuit with n inputs, noted by $X_1, X_2, \ldots, X_n$, and m outputs, noted by $Z_1, Z_2, \ldots, Z_m$, for which the outputs could be expressed according to inputs using a mathematical model such as:

$$Z_1 = f_1(X_1, X_2, \ldots, X_n)$$
$$Z_2 = f_2(X_1, X_2, \ldots, X_n)$$

$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$Z_I = f_i(X_1, X_2, \ldots, X_n)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$Z_m = f_m(X_1, X_2, \ldots, X_n)$$

If we note by $X=\{ X_1, X_2, \ldots, X_n \}$ the set of input variables and by $Z=\{ Z_1, Z_2, \ldots, Z_m \}$ the set of the output variables, then a combinational logical circuit could be easily described mathematically by the triplet CLC = (X, Z, F), in which the input-output function $F : X \rightarrow Z$ is independent of time.



In the synthesis of a circuit CLC generally one starts by classifying the functioning conditions according to the requirements imposed by a table of truth and the specification of the operation and non-operation state. A condition of operation or non-operation is referring to the state of the sizes of output $Z_1, Z_2, \ldots, Z_m$ and input $X_1, X_2, \ldots, X_n$. Thus, the vector of the input variables could have the values $\forall_1, \forall_2, \ldots, \forall_n$ or $\exists_1, \exists_2, \ldots, \exists_n$ that belong to the multitude $\{0, 1\}$.

A condition of operation is as follows:

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

and a condition of non-operation is expressed by the mathematical relation:

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

The problem of CLC circuit synthesis is incompatible if the same state appears at the same time, both in an operation condition and in a non-operation one:

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

But if such situations do not appear we are dealing with a compatible problem that could be determined or undetermined. A problem of such type is compatibly determined if the number of operation and non-operation situations is equal to 2^n.

The system is compatibly undetermined, having the degree of determination r, if the number of the operation conditions noted by q, and the number of non-determination conditions noted by p, respects the relation $q + p < 2^n$, and the degree of non-determination r is being computed with the relation:

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

The r undetermined states, noted by $8_1, 8_2, \ldots, 8_r$, could receive the values of the set $\{0, 1\}$. The assignment of one of values 0 or 1 depends, in the classical minimization, on the realization or not of the minimization. With these r undetermined solutions 2^r solutions could be obtained that could satisfy the conditions of the

problem. In the synthesis of a CLC the following steps are involved:

- Problem utterance; Formulation of the truth table;
- Minimization of the truth function starting from its analytical form or the truth table that describes it;
- Correlated minimization of the commutation functions; Scheme analysis and hazard elimination;
- Hardware implementation of the logical functions with elements of discrete commutation.

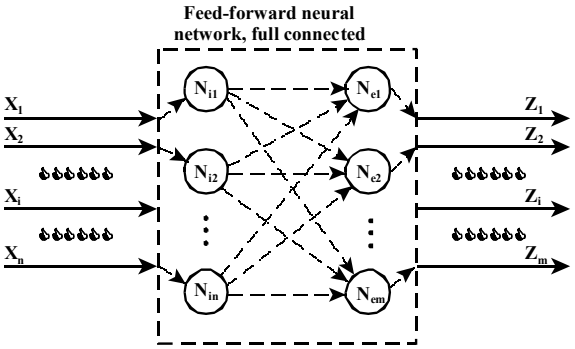
A delicate problem in the case of the classical approach of CLC circuits` synthesis is that not always an absolute optimum rigorous scheme is achieved. Thus, in the case of synthesis of the complex functions with a high number of input variables (those with n>6), with many outputs and undetermined states, the algebraic and topological methods are very difficult to be applied.

3. CLC SIMULATION WITH THE HELP OF FEED-FORWARD NEURAL NETWORKS

This paper proposes a special treatment of these CLC-s by means of the utilization of neural networks. The solution was suggested by the functional similitude existing between a CLC and a neural network with n inputs and m outputs. In fact, a neural network of three layers was used, the first layer having n input neurons and the third m output neurons. For the neural network driving the set of input data is used and the set of output given by the table of truth:

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

The circuit whose functioning was simulated was a BCD to 7 segments



decoder (7446). In the course of the search of an optimal configuration of the form 4:9:7 was found. The results of training with the 16th configurations of input are illustrated in table bellow. Index n shows the real output of the network.

In	a_n	a	b_n	b	c_n	c	d_n	d	e_n	e	f_n	f	g_n	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. CONCLUSIONS

As a conclusion, the methodology of combinational circuits simulation, with the help of neural networks, requires the following steps:

Step 1. Formulation of the truth table for the chosen combinational circuit;

Step 2. Selection of a neural network with three layers which has n neurons on the first layer, a number of neurons in the hidden layer that will be find and m neurons in the output layer;

Step 3. Training the neural network using the truth table;

Step 4. Verification of the correct functioning of the network;

Step 5. Network hardware implementation.

Bibliography

1. G. M. Ştefan, I. V. Drăghici, T. Mureşan, E. Barbu – *Digital integrate circuits*, Ed. Didactică şi Pedagogică, Bucharest, 1983.
2. D. Dumitrescu, H. Costin – *Neural networks. Theory and applications.* Ed. Teora, Bucharest, 1996.
3. R. Joldeş – *Neural Networks. Basic notions. Implementations.*, Seria Didactica, "1 Decembrie 1918"University, Alba Iulia, 1995.

Authors: Remus Joldeş, rjoldes@uab.ro.
Ioan Ileană, iileana@lmm.uab.ro.
Corina Rotar, crotar@lmm.uab.ro.
1 Decembrie 1918" University of Alba Iulia, N. Iorga 11-13, Alba Iulia,

2500, Romania.