

Algorithm for constructing logical neural networks based on logical various-valued functions

Dimitrichenko D. (dimdp@rambler.ru).

Institute of Applied Mathematics and Automation KBSC RAS (IAMA KBSC RAS), Nalchik, Russia.

ABSTRACT

The intelligent control system, as a set of production rules, is implemented in the form of an various-valued logical function. The combined use of mathematical logic and neural network methods gives the intelligent control system additional flexibility and the possibility of self-learning. In this paper we propose a method for representing various-valued logic function in a logical neural network. This logical neural network will keep the totality of cause-and-effect relationships identified using various-valued logic functions within a given specified area. These logic operations are implemented by special logic neural cells: conjunctors and disjunctors. The theorems given in this article justify the possibility of constructing such neural networks. The method of proof of these theorems contains an algorithm for constructing logical neural networks for a finite number of steps.

General problem statement

Let $W = \{w_1, \dots, w_m\}$ - be the set of the objects forming the training selection (TS). Various subjects to recognition can act as representatives of TS: graphic (or sound) images, situations that require certain responses of the operating system, diagnostic messages and so on. Descriptions of the objects comprising TS W represent n -component vectors where n is a number of the signs applied to characterize the analyzed objects, and j - coordinate of these vectors equals to j -sign, $j=1, \dots, n$. The number of such vectors is equal to m according to the number of the analyzed objects in TS. In the description of an object, the lack of information on value of this or that sign is admissible. Compliance of a set of objects and the signs characterizing them is presented in the following table 1.

Table 1.

x_1	x_2	...	x_n	W
$x_1(w_1)$	$x_2(w_1)$	...	$x_n(w_1)$	w_1
$x_1(w_2)$	$x_2(w_2)$	...	$x_n(w_2)$	w_2
...	...	...	...	...
$x_1(w_m)$	$x_2(w_m)$	...	$x_n(w_m)$	w_m

$$X_j = \{x_1(w_j), x_2(w_j), \dots, x_n(w_j)\}$$

- a vector of qualitative characters, where each element is the fixed sign of the object, is a part of TS.

$$W = \bigcup_{j=1}^m w_j$$

- a set of the characterized objects.

Each corresponding sign of $x_j(w_i)$ is coded by a various-valued predicate of atomicity of k_j , $j=1, \dots, n$.

$$f(x) = \bigwedge_{j=1}^n (\bigwedge_{i=1}^m x_j(w_i) \rightarrow w_j) = \bigwedge_{j=1}^n (\bigvee_{i=1}^m x_j(w_i) \vee \overline{w_j})$$

The specified type of function follows from the known logical identity: $a \rightarrow b = \overline{a} \vee b$, where a - is a conjunction of characteristics (signs) defining an object, and b - the predicate equal to unity, when w_j becomes equal to the corresponding defined object. We will call such predicates object predicates, and the disjunctors containing such predicates - production disjunctors.

The expediency of application of the various-valued predicates follows from the fact that the adequate description of various characteristics requires predicates of various atomicity. For example, two-unit predicate for coding of the fact of presence (or absence) of a certain property in the description of the considered objects is enough. The three-digit predicate is convenient when we speak about various extents of sign presence in an object: "No", "Partially", "Present".

The set of production rules of the following type stands as a basis for creation of various-valued logic functions $F(X, W)$ where the X -set of signs, and W -the objects of TS:

(Conjunction of the signs $_1 \rightarrow$ Object),

(Conjunction of the signs $\rightarrow$ Object),

...

(Conjunction of the signs $_m \rightarrow$ Object_m).

The set from m production rules by means of m -1 conjunction operation forms one logical expression which makes various-valued logic function of $F(X, W)$ [7]. Consecutive disclosure of m brackets in [7] operation of the generalized denial leads to creation of various-valued logic function of $F(X, W)$ where X - the n -component vector of logic variables coding the set of signs of objects: $X = (x_1, x_2, \dots, x_n)$, and $W = \{w_1, \dots, w_m\}$ - a set of the characterized objects of power m , that form the training selection.

As a result of this conversion function $F(X, W)$, takes a form of a disjunction of subclasses of objects, each of which represents conjunction of signs and objects on which these subclasses are formed.

Finding the value of the logic function of $F(X, W)$ from the set of characteristics of X^* as a final result we have a disjunction of those subclasses of objects from a set of W , values of variables of which match the values of variables in a vector of a request of X^* . Let's notice that at the resultant expression there will be also objects which satisfy to a request of X^* in number of signs, smaller, than n . Here, the more numbers of variables from a request of X^* corresponds object w^* , belonging to W , the bigger the number of subclasses with its participation will be in the final answer of W^* . Therefore, to identify the best object which is most satisfy the request of X^* it is enough to apply the procedure of frequent analysis to the received set of subclasses of W^* .

Introduction

The solution of a task of intellectual control of the mobile robot is impossible without the corresponding information support realized by an information system. The information from various sensors characterizes current state of the equipment of the mobile robot therefore it is used in the system of automatic control as the feedback. The feedback signals allow to automatically correct the operating programs to ensure the stability in work of a production system in general. The information from various (both external and internal) sensors is also used for self-checking and self-diagnostics of mobile robot conditions.

At the same time, optimal solutions to control the mobile robot have to be developed. Input information is given to the operating system on the basis of which the optimal solution is developed. The set of such solutions can be written down in the form of production rules: either in advance formulated by experts, or received later in the course of adaptation performing procedures [9, 11].

Logical neural networks

In the work [2] circuitry approach for creation of logical neural network was offered. Relationships of the cause relationships between the sets of entrance signals that get on an entrance layer of neural network by aggregate incoming signals and by made decisions are established for this purpose (fig. 1). In this case the intellectual system of decision-making is presented in the form of the following set of implicative statements:

Aggregate of the signals $_1 \rightarrow$ Solution $_1$,

Aggregate of the signals $_2 \rightarrow$ Solution $_2$,

...

Aggregate of the signals $_m \rightarrow$ Solution $_m$.

Let's notice that generally one and the same solution can follow from different sets of input signals.

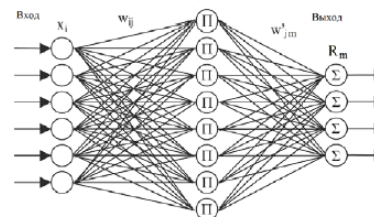


Fig. 1. Logical neural network

At the same time it is supposed that calculations in each of m implicative expressions are made in parallel according to features of neural network functioning. As output result the set of accurately (or it is indistinct) weighed solutions is formed. While constructing the logical neural network the corresponding production rules were used. Each set of entrance signals is considered as conjunction of the corresponding logic variables describing the initial situations acting as the training selection for decision making system. For the creation of logical neural network two types of neurons are used: conjunctors and disjunctors. The weight of input signals of these neurons are selected so that at accurate values of input signals the results of functioning of these neurons correspond the values of logical actions of conjunction and disjunction.

The plotting of the various-valued logical neural network

We see that one and the same system of production rules (implicative statements) is a basis as for construction of the various-valued of logical functions $F(X, W)$, so for creation of the logical neural networks trained for creation of intellectual control systems. The situations characterized by indications of external and internal sensors act as an entrance vector of X here, and the optimal solutions corresponding to these situations act as subjects to recognition.

The following theorem is proved:

Theorem 1. Any various-valued logical function $F(X, W)$ is representable in the form of logical neural network, the set of logical communications in which is defined by the structure of production disjunctors.

Thus, various-valued logical function $F(x, W)$ it is representable in the form of the three-layer neural network, where:

1. The entrance layer are the predicates x_j , of the k_j value, each $j=1, \dots, n$.

2. The set of objects or output signals of $W = \{w_1, \dots, w_m\}$ as w_m is the output layer that makes final calculations.

3. Production disjunctors in which conjunction operations and disjunctions replaced by neurons conjunctors and disjunctors act as the intermediate, ulterior layer that make calculations.

4. Such sets of input signals of logical neural network to which no output signals are compared meet free knowledge of various-valued logic function. In an explicit form free knowledge of logical neural network does not exist. However, free knowledge can be connected to a special circuit, and at rather high potential of a signal on this line the procedure of a finish-teaching of the current logical neural network is started [4].

5. The neurons that send the output signal on those objects which enter the corresponding subclasses are compared to production disjunctors of logic function.

In case of accurate values of variables any conjunctor corresponding to a production disjunctor of various-valued logic function of $F(X, W)$, transmits a signal of single weight to each of the objects entering this disjunctors (subclass).

If to give values of logic variables in compliance with some request of X^* on an input of the neural network built this way then at the output the set of the objects satisfying to this request which weight in accuracy match the results of work of the procedure of frequency analysis applied to find the objects which are most fully answer the request of X^* while finding the various-valued logic function of $F(X, W)$: $W^* = F(X^*, W)$.

Obviously, the results of the proof of Theorem 1, as well as the method of its proof and the algorithm for constructing a logical neural network, can be completely transferred to the case of an extended logical neural network, making some changes to them.

The transition to fuzzy logic is provided by the use of various-valued predicates and flexibility conjunctors and disjunctors. Additional expressiveness of the artificial system of recognition or decision-making gives the possibility of using fuzzy logic in the conditions of noisy or fuzzy source data.

ACKNOWLEDGMENTS & REFERENCES

The reported study was funded by RFBR according to the research project №18-01-00050-a