

Using a Boolean derivative to evaluate the significance of properties of recognized objects

L A Lyutikova^{1,*}

¹Institute of Applied Mathematics and Automation KBSC RAS (IAMA KBSC RAS), 89a, str. Shortanova, 360000, KBR, Nalchik, Russia

Abstract. This paper offers an approach for evaluating the significance of individual characteristics of recognized objects. The scope of this approach is not the subject area where objects and characteristics of these objects are specified, but a trained $\Sigma\Pi$ - neural network that works correctly on the specified subject area. In this paper, we propose a method for constructing a crucial function based on the weight characteristics of a correctly functioning $\Sigma\Pi$ - neuron. A logical derivative is used to evaluate the significance of object characteristics. This makes it possible to track how the decision function will change its value if one or more object characteristics change their value. This will allow us to draw a conclusion about the most important properties of the subject area under consideration.

1 Introduction

Today, neural networks are one of the most popular tools for solving poorly formalized tasks. Tasks for which there is no mathematical formulation and formal algorithmic solutions. These are tasks for the solution of which heuristics are required in order to find a more rational solution, rather than an exact mathematical one, by eliminating previously unsuitable solutions. The data and knowledge of this area are characterized by factors: incompleteness, unreliability, inaccuracy, ambiguity.

Despite the fact that neural networks do a good job with a great many of such tasks, the rules for making decisions are not clear to the user. Available structure and weight characteristics, which acquired a neural network as a result of training. And to identify the logical connections according to the characteristics of a correctly functioning neural network, this means to gain new knowledge about the subject area being studied.

By analogy with natural intelligence, which can be first taught by examples, then rules, methods, etc. are realized or created.

* Corresponding author: lylarisa@yandex.ru

2 Construction of the classifier function based on the structure of the $\Sigma\Pi$ -neuron

As is known, a $\Sigma\Pi$ -neuron (sigma-pi neuron) is a generalization of the classical model of a formal neuron with a linear function of summing $sp(x_1, \dots, x_n)$ input signals.

$\Sigma\Pi$ -neuron is represented by the following structure

$$sp(x_1, \dots, x_n) = \sum w_i \prod x_i$$

Where $\{w_1, w_2, \dots, w_k\}$ is the set of weights of a given $\Sigma\Pi$ -neuron that recognizes k elements of a given subject $Y = \{y_1, y_2, \dots, y_k\}$ area formed by the corresponding set of features $\{X_1, \dots, X_k\}$ [1].

Example. Let the following training set be given:

Table 1. Example.

x_1	x_2	x_3	y
0	0	1	a (2)
0	1	1	b(4)
1	1	0	c(6)

The set of attributes X is represented by the following values:

$$x = \{x_1 = (0,0,1), x_2 = (0,1,1), x_3 = (1,1,0)\}$$

and the set of objects $\{a, b, c\}$ can be transcoded for training in

$$a-2, b-4, c-6.$$

As a result of training according to the table, the $\Sigma\Pi$ -neuron will look like:

$$sp(x_1 x_2 x_3) = 2x_3 + 2x_2 x_3 + 4x_2 x_1$$

Any query (x_1, x_2, x_3) , which is presented in the table, will be identified with the corresponding object.

If the query does not coincide with the values of the variables that are in the training set, for example $(0,1,0)$, then the result may be incorrect or it may not exist at all.

$$sp(0,1,0) = 2 * 0 + 2 * 1 * 0 + 4 * 0 * 0 = 0 [7].$$

Although, it could be an object, b-4, or c-6, in cases where there are inaccuracies, noise, interference in the data.

To obtain more stable solutions, a trained neuron requires additional corrective methods.

3 Construction of the decisive function according to the structure of the $\Sigma\Pi$ -neuron

When constructing the decisive function, you may not know the training set; it is enough to know the value of weights and the structure of the neuron. The function is built on a tree, the construction algorithm of which is described in [1].

The number of levels is equal to the largest number of products of variables in each of the terms +1. In the example there will be 3.

At the bottom level are variables $\{x_1, x_2, \dots, x_n\}$. The weights of the first layer $\{w_1, w_2, \dots, w_r\}$, respectively, are objects $\{y_1, y_2, \dots, y_r\}$ on each subsequent layer $y_{k+1} = w_{k+1} + \sum y_i$, where i the indices of the corresponding objects whose variables are included as factors in the element by y_{k+1} .

Example.

$$sp(x_1x_2x_3) = 2x_3 + 2x_2x_3 + 4x_2x_1$$

We will restore the objects of the training sample and find generalizing logical rules (see Fig. 1).

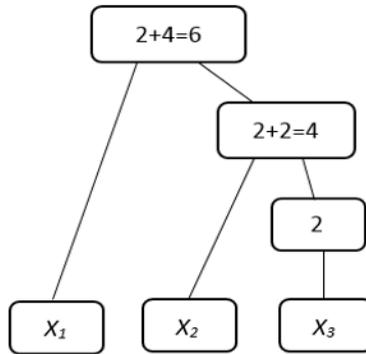


Fig. 1. Recovery of objects of the training sample.

From each vertex y_k to each variable x_i : $P(y_k) \& P(y_{k-1}) \& \dots \& P(y_i) \& x_i$

$$P(y_k) = 1, \text{ if } y = y_k \quad P(y_k) = 0, \text{ if } y \neq y_k$$

For this example, the minimum set of rules will look like:

$$F(x_1x_2x_3) = P(6)x_1 \vee P(6)P(4)x_2 \vee P(2)P(2)x_3$$

These rules are sufficient if only the presence of a characteristic is important for the data under consideration. But these rules are not enough if the value of the variable zero is also informative for decision making. And these rules are not enough, in the case of multi-valued coding. Therefore, there is a need to build additional trees, or imaginary paths, in the figure this is indicated by a dash-dot line. For example, it looks like in Fig. 2.

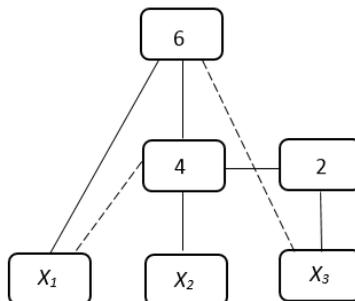


Fig. 2. Object part of a logical function.

A dashed line indicates the relationship with the variable's negation.

Then the decision function for our example will look like this:

$$F(x_1x_2x_3) = P(6)x_1\bar{x}_3 \vee P(6)P(4)x_2 \vee P(4)P(2)x_3\bar{x}_1$$

That is, the most important features for the source data will use the logical derivative.

4 Some properties of operations of logical differentiation of boolean functions

Logical differential and integral calculus are the directions of modern discrete mathematics and find their application in the problems of dynamic analysis and synthesis of discrete digital structures. The basic concept of logical differential calculus is the derivative of a Boolean function, the idea of which in the form of a Boolean difference was obtained back in [8-10].

By some of its properties, the Boolean derivative is an analogue of the derivative in classical differential calculus.

Definition 1. The first-order derivative $\frac{\partial f}{\partial x_i}$ of the Boolean function $f(x_1, \dots, x_n)$ with respect to the variable x_i is the sum modulo 2 of the corresponding residual functions:

$$\frac{\partial f}{\partial x_i} = f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \oplus f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$$

Definition 2. The weight of the derivative $P(\frac{\partial f}{\partial x_i})$ of a Boolean function is the number of constituents ("1") of this derivative.

Statement 1 The greater the weight of the derivative, the greater the function $f(x_1, \dots, x_n)$ depends on the variable x_i .

Definition 3. A mixed derivative of the k-th order of a Boolean function $f(x_1, \dots, x_n)$ is an expression of the form:

$$\frac{\partial^k f}{\partial(x_1 \dots x_k)} = \frac{\partial}{\partial x_k} \left(\frac{\partial^{k-1} f}{\partial x_1 \dots \partial x_{k-1}} \right)$$

In this case, the order of the fixed variable does not matter. The k-th derivative determines the conditions under which this function changes its value while changing the values of x_1, \dots, x_k .

For our example:

$$\frac{f(X)}{dx_1} = P(6)\bar{x}_3 \vee P(6)P(4)x_2 \oplus P(6)P(4)x_2 \vee P(4)P(2)x_3 = P(4)P(2)x_3 \vee P(6)\bar{x}_3$$

The derived derivative can classify objects by the variable x_3 .

Derivative of the variable x_2

$$\begin{aligned} \frac{f(X)}{dx_2} &= P(6)\bar{x}_3x_1 \vee P(2)P(4)\bar{x}_1x_3 \oplus P(2)P(4)\bar{x}_1x_3 \vee P(6)\bar{x}_3x_1 \vee P(4)P(6) \\ &= P(6)P(4)\bar{x}_3\bar{x}_1 \vee P(6)P(4)x_1x_3 \end{aligned}$$

This result gives conflicting data on only two objects, and makes their classification impossible. Therefore, it can be argued that the variable x_2 reflects the most important properties for the data under study. And the variables x_3 and x_1 are dependent, i.e. they are ensemble variables. $x_1 = \bar{x}_3$.

5 Conclusion

Building a logical function based on the weight characteristics of a neural network gives an idea of the hidden rules of functioning of this neural network, and makes it possible to correct the result in cases when the neural network is wrong. For example, if there is interference in the data.

Analysis of the constructed decision function using logical derivative methods allows us to formalize the process of finding the coefficients of importance for the characteristics of object properties. And also to find the ensemble characteristics. This is particularly important when data is distorted due to information noise, or for other reasons.

As a result, the quality of automated solutions to intellectual problems, their reliability, and ensuring the accuracy of achieving the correct solution are significantly improved by using the most effective systems for analyzing source data and developing more accurate methods for processing them.

Acknowledgements

The reported study was funded by RFBR according to the research project № 19-01-00648-a.

References

1. Lyutikova L 2018 *Procedia Computer Science* **145** 312
2. Graves A, Wayne G, Reynolds M, Harley T, Danihelka I, Grabska Barwinska A, Gómez Colmenarejo S, Grefenstette E, Ramalho T, Agapiou J et al 2016 *Nature* **538(7626)** 471
3. Naimi A I, Balzer L B 2018 *European Journal of Epidemiology* **33** 459
4. Yang Fan, Yang Zhilin, Cohen W W 2017 *Advances in Neural Information Processing Systems* 2320
5. Flach P 2012 *Machine Learning: The Art and Science of Algorithms that Make Sense of Data* (Cambridge University Press) p 396
6. Rahman Akhlaqur, Tasnim Sumaira 2014 *International Journal of Computer Trends and Technology* **10(1)** 31
7. Dyukova E V, Zhuravlev Yu I, Prokofiev P A 2015 *Machine learning and data analysis* **1(11)** 1555
8. Zhuravlev Yu I 1978 *Problems of Cybernetics* **33** 5
9. Lyutikova L A, Shmatova E V 2016 *Information Technologies* **22(4)** 292