

Using a logical Derivative to Determine the Information Content of Object Properties in Speech Recognition Tasks

Larisa Lyutikova^{a,b}

^a Institute of Applied Mathematics and Automation KBSC RAS, 360000, KBR, Nalchik, str Shortanova. 89A

^b Institute for Computer Science and Problems of Regional Management KBSC RAS, 360000, KBR, Nalchik, st. I. Armand, 37A.

Abstract

This paper offers an approach for determining the information content of object properties in recognition tasks. 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 a given subject area. In this paper, we propose a method for constructing a decision 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.

Keywords¹

Decisive function, Boolean derivative, analysis of the data; the algorithm $\Sigma\Pi$ -neuron; decision trees; corrective surgery.

1. Introduction

The research area of this work is related to such scientific direction as pattern recognition, the purpose of which is to classify objects into several categories or classes. The practicality of this method is obvious, since after classification, working with a certain class of information requires fewer resources than working with its full volume. In practice, solving problems related to pattern recognition is a complex theoretical and practical task. This is primarily because each specific case has its own specifics, which does not allow creating a universal algorithm for working with information.

Today, neural networks are one of the most popular tools for solving problems for which there is no acceptable mathematical model or exact algorithm, but there are many so-called heuristics, based on which they try to get more or less accurate patterns inherent in the studied subject area [4,5].

The data used to find patterns are not perfect, they are usually incomplete, have a lot of inaccuracies and distortions.

Despite the fact that neural networks do a good job with a great variety of such tasks, the rules of their decision-making are not clear to the user. Only the structure and weight characteristics that the neural network acquired because of training are available.

Identifying logical connections based on the characteristics of a correctly functioning neural network is an important task, since neural networks are built on heuristics, and their solutions can be ambiguous. It also takes many cycles to build object models, which entails long time costs, and training can lead to a dead end. The problem of retraining is acute.

To solve these problems, corrective methods are important, since they provide an opportunity to gain new knowledge about the patterns in the subject area under study. This will further help you

YRID-2020: International Workshop on Data Mining and Knowledge Engineering, October 15-16, 2020, Stavropol, Russia

EMAIL: lylarisa@yandex.ru (Larisa Lyutikova)

ORCID: 0000-0003-4941-7854 (Larisa Lyutikova)



© 2020 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

better understand the nature of the data being studied, and therefore develop algorithms that are more expressive

To date, the main methods of pattern recognition have been identified. There are four main methods: the method of comparison with the standard, the statistical method, the method using artificial neural networks (ins), and the structural method [2, 4]. However, these methods have a number of obvious disadvantages. The method of comparison with the standard is accompanied by distortions of the samples under consideration, which requires taking into account a considerable number of cases of small deviations from the standard. Statistical methods face a number of difficulties in solving practical problems, since samples from each class are often not representative enough, and probability distribution density functions are difficult to construct. Neural networks are characterized by long-term training based on the consideration of many examples. Structural methods are sensitive to distortions of recognized images and require a complex procedure for constructing a set of features.

This situation leads to the need to combine these methods in order to compensate for the shortcomings of some approaches with the advantages of others, which is actively used in solving practical problems.

2. The purpose and objectives of the study

This paper offers a combined approach to solving problems of pattern recognition, which makes it possible to correct the operation of the neural network. In cases of an already trained neural network, even if the subject area is not known, based only on the weight characteristics of the neuron, get logical connections in the data under study, build a logical solver function for a given area, and thus go to structural methods of solving. Use the Boolean derivative to identify the most important characteristics for each object and for the subject area as a whole. This means getting a more correct picture of the area under study, and therefore the ability to work with it

For example, if we are talking about pattern recognition, this approach makes it possible to build a logical classifier function that can correct the operation of this network using only the weights of the neural network. Further investigation of this function will reveal the most important features for each object. You can also show how changing the values of certain attributes results in changes in objects.

3. Research methods

Such authors describe methods of teaching Sigma-PI neuron as A.V. Timofeev, Z. M. Shibzukhov. A.V. Timofeev proposed training the network in one pass.

The construction of the classifier function will be based on the study of the weight values of the $\Sigma\Pi$ -neural network.

As a solver (a function or method that divides the subject area into classes), we will consider the $\Sigma\Pi$ - neuron (Sigma-PI neuron), which is a more expressive generalization of classical neural networks, since it corresponds more formally to natural neurons. It can be represented by the following linear function of the input signals $sp(x_1, \dots, x_n)$ Or: $sp(x_1, \dots, x_n) = \sum w_i \prod x_i$, $\{w_1, w_2, \dots, w_k\}$ - the coefficients, which becomes a neuron in the learning process.

If the subject is a $\Sigma\Pi$ - neuron, it recognizes k elements from the specified subject area $Y = \{y_1, y_2, \dots, y_k\}$ generated by the corresponding set of attributes $\{X_1, \dots, X_k\}$ [1].

EXAMPLE. Suppose the area for learning a Sigma PI neuron is represented by the following set of features and objects:

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)

Attributes are characterized by corresponding codes and are represented by a set with values:

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

The object is $\{a, b, c\}$ which are encoded accordingly

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

Encoding objects with numerical values is necessary for fast training of this neural network.

After training, which is performed in one pass, the $\Sigma\Pi$ - neuron will have the form:

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

Any query (x_1, x_2, x_3) , that is represented in the table will be identified with its corresponding object..

If the query does not match the values of the variables that are in the training sample, for example $(0,1,0)$, the result may not be correct or it may not be at all.

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

The network doesn't recognize any elements. Because an object with exactly the same characteristics is not in the data. However, it could be an object, b4, or C-6, if there are inaccuracies, noise, or interference in the data.

Therefore, to obtain a possible class of solutions, the trained neuron requires additional corrective approaches.

4. Construction of a crucial logical function based on the structure of a $\Sigma\Pi$ -neuron and identification of significant features

When constructing a decision function, you don't need to know the training sample. It is enough to know the value of the weights and the structure of the $\Sigma\Pi$ - neuron. The function is constructed using a tree whose construction algorithm 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 of them.

Arrange the variables $\{x_1, x_2, \dots, x_n\}$ to the lower levels of the tree. The second layer is the coefficients before the terms with one variable; the third is the coefficients before the terms with two variables, and so on. The value of each node will be considered as $y_{k+1} = w_{k+1} + \sum y_i$, where are the indexes of the corresponding objects whose variables are included as factors in the element on y_{k+1} .

EXAMPLE. Let us build a tree for defining the main properties of objects.

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

Because of building a tree from the training sample, you can identify the basic rules for the relationship between objects and their characteristics (see figure 1).

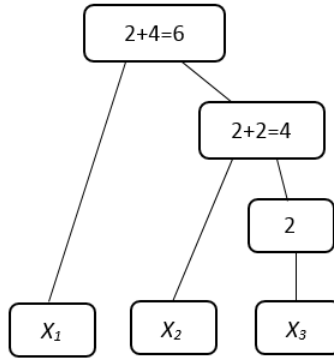


Figure 1: Decision Tree for a given subject area

The connection object y_k with each property $x_i : P(y_k) \& P(y_{k-1}) \& \dots \& P(y_i) \& x_i$

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

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

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

These rules are sufficient if only the presence of the attribute is important for the data under consideration.

However, these rules are not enough if the value of the variable zero is also informative for making decisions. In addition, these rules are not enough in the case of multi-value encoding.

Therefore, there is a need to build additional trees, or imaginary paths in the figure this is indicated with a dash-dotted line.

For example, it looks like in figure 2.

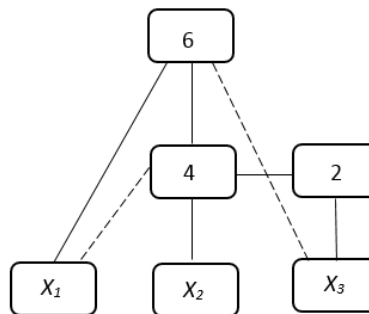


Figure 2: Object part of a logical function

If an item doesn't have any attribute, the path to it will be marked with a dashed line.

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

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

To identify significant variables in this function, i.e. the most important features for the source data, we will use the logical derivative.

Logical differentiation in this case implies differentiation of Boolean functions, which in some sense is analogous to classical differentiation [7, 8]. In the semantic meaning, the Boolean derivative can show the degree of dependence of a function on a given variable, and indicate how justified the expectations of changing the values of the function are in cases of changing the value of the variable.

Definition 1. The Boolean derivative $\frac{\partial f}{\partial x_i}$ from a Boolean function $f(x_1, \dots, x_n)$ by the variable x_i

we will call 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 of a Boolean function $P(\frac{\partial f}{\partial x_i})$ the number ("1") in the column of values of the derivative is called

Statement 1. The weight of the derivative for a given variable shows how much the function $f(x_1, \dots, x_n)$ depends on the variable x_i in comparison with other variables.

Definition 3. Expression of the form:

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

it is called the mixed derivative of the k -th order of the corresponding variables.

However, the order of the fixed variable does not matter.

A mixed Boolean derivative with respect to k -variables indicates the conditions under which a function changes its value while simultaneously changing the values of x_1, \dots, x_k .

If it is necessary to determine the most important characteristics of the objects under consideration in a given subject area after the logical function solver is constructed, you can consider the Boolean derivatives for each variable and thus select the most significant properties.

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 resulting derivative can classify objects by the variable x_3 .

Derivative of the variable x_2

$$\frac{f(X)}{dx_3} = P(6)\bar{x}_3x_1 \vee P(2)P(4)\bar{x}_1x_3 \oplus P(2)P(4)\bar{x}_1x_3 \vee P(6)\bar{x}_3x_1 \vee P(4)P(6) = P(6)P(4)\bar{x}_3\bar{x}_1 \vee P(6)P(4)x_1x_3,$$

This result gives conflicting data about only two objects, and makes it impossible to classify them. 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

$$x_1 = \bar{x}_3.$$

Because of the analysis, we can say that knowing the weight values of the neural network, we can build a logical function that reflects the patterns in the data. These patterns may not be obvious when using conventional domain analysis methods. The study of the obtained logical function can reveal the most significant features for each object, and for the considered data as a whole.

5. Comparisons with well-known attention methods in neural networks

The successfully implemented Attention technology is a way to tell the network what to pay more attention to, that is, to report the probability of a particular outcome depending on the state of neurons and incoming data. The Attention layer implemented in Keras itself identifies factors based on the training sample that reduce network error. Identification of important factors is performed through the method of error back propagation, just as it is done for convolutional networks. When we train a network on data, the importance becomes a function of the probability of an outcome depending on the data received by the network.

In this method, it is proposed to identify logical connections between the objects of the training sample, without the training sample itself, which was used to form the analyzed neural network. In contrast to the Attention method, the proposed approach identifies logical rather than statistical patterns in the data. Allows you to avoid repeating the error back propagation procedure many times. Knowledge of logical patterns, as opposed to statistical ones, will allow you to formalize and more accurately understand the nature of the analyzed data. This approach is relevant in cases when we do not have a training sample with large amounts of data (rare events that are difficult to reproduce), or when processing large amounts of data is faced with the necessary capacity.

6. Conclusion

A large number of methods are used to create systems capable of obtaining knowledge from data. In this paper, we consider an approach that allows, knowing only the value of the neural network weights, to find patterns in the data and build a logical function that reflects these patterns.

The proposed analysis of the sensitivity of this function by the methods of the logical derivative allows one to formalize the process of finding the coefficients of importance for the characteristics of the properties of objects. This is important when the data is incomplete, fuzzy, or distorted due to information noise.

All this leads to the development of methods for more accurate solution of intellectual problems.

This work was supported by the RFBR grant No. 19-01-00648-a

7. References

- [1] Lyutikova L. Sigma-Pi Neural Networks: Error Correction Methods", published in *Procedia Computer Science*, Volume 145, 2018, Pages 312-318
- [2] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka GrabskaBarwinska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 2016. № 538(7626). Pp.471–476.
- [3] Ashley I. Naimi, Laura B. Balzer Stacked generalization: an introduction to super learning // *European Journal of Epidemiology* (2018) 33:459–464.
- [4] Fan Yang Zhilin Yang William W. Cohen Differentiable Learning of Logical Rules for Knowledge Base Reasoning // *Advances in Neural Information Processing Systems*. Volume 2017-December, 2017, Pages 2320-2329.
- [5] Peter Flach *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, 2012. — 396 p. — ISBN: 978-1107096394.
- [6] Rahman Akhlaqur & Tasnim Sumaira. Ensemble Classifiers and Their Applications: A Review // *International Journal of Computer Trends and Technology*. (2014). Vol. 10. No1. Pp.31 – 35.
- [7] Dyukova Ye.V., Zhuravlev YU.I., Prokof'ev P.A. Metody povysheniya effektivnosti logicheskikh korrektorov // *Mashinnoye obucheniye i analiz dannykh*. 2015. T. 1. № 11. S. 1555-1583.
- [8] Lyutikova L. A., Shmatova E. V. Analysis and synthesis of pattern recognition algorithms using variable-valued logic // *"Information Technologies"*. Volume 22. No. 4. 2016.S. 292-297.