

## Методы эффективного анализа данных для решения плохо формализованных задач распознавания

Л. А. Лютикова

Институт прикладной математики и автоматизации –  
филиал Кабардино-Балкарского научного центра Российской академии наук  
360000, Россия, г. Нальчик, ул. Шортанова, 89 А

**Аннотация.** В работе проводится сравнительный анализ результатов решений нейронных сетей и логических алгоритмов. Предлагается метод, позволяющий улучшить результаты решений нейронных сетей. В качестве решателя рассматривается сигма-пи нейронная сеть, а данные представлены объектами и характеризующими эти объекты признаками. В работе показано, что существуют логические подходы, способные наиболее полно описать все возможные закономерности в исследуемой области. В то же время нейронные сети в качестве решений выдают только часть из возможных решений. Поэтому предлагается проанализировать и дополнить эти решения. Для этого в работе рассматривается возможность построения логического корректора, способного по структуре нейронной сети построить логическую функцию, а дальше реализовать ее в виде логических нейронных сетей. Такой подход позволяет выявить логические связи между объектами в исследуемых данных, знание логических закономерностей позволит провести формализацию и точнее понять природу анализируемой области.

**Ключевые слова:** логический классификатор, нейронная сеть,  $\Sigma\Pi$ -нейрон, решающие правила

Поступила 16.11.2022, одобрена после рецензирования 30.11.2022, принята к публикации 01.12.2022

**Для цитирования.** Лютикова Л. А. Методы эффективного анализа данных для решения плохо формализованных задач распознавания // Известия Кабардино-Балкарского научного центра РАН. 2022. № 6(110). С. 106–114. DOI: 10.35330/1991-6639-2022-6-110-106-114

MSC: 68T27

Original article

## Methods of effective data analysis for solving poorly formalized recognition tasks

L.A. Lyutikova

Institute of Applied Mathematics and Automation –  
branch of Kabardino-Balkarian Scientific Center of the Russian Academy of Sciences  
360000, Russia, Nalchik, 89 A Shortanov street

**Abstract.** The paper presents a comparative analysis of the results of neural network solutions and logical algorithms. A method is proposed to improve the results of neural network solutions. The work considers a sigma-pi neural network as a solver, and the data are represented by objects and features characterizing these objects. The paper shows that there are logical approaches that can most fully describe all possible patterns in the study area. At the same time, neural networks provide only a part of the possible solutions as solutions. Therefore, it is proposed to analyze and supplement these solutions. To do this, the paper considers the possibility of constructing a logical corrector capable of constructing a logical function according to the structure of a neural network, and then implementing it in the form of logical neural networks. This approach allows to identify logical relationships between objects in the data under study,

knowledge of logical patterns will allow to formalize and more accurately understand the nature of the analyzed area.

**Keywords:** logical classifier, neural network,  $\Sigma\Pi$ -neuron, decision rules

Submitted 16.11.2022,

approved after reviewing 30.11.2022,

accepted for publication 01.12.2022

**For citation.** Lyutikova L.A. Methods of effective data analysis for solving poorly formalized recognition tasks. *News of the Kabardino-Balkarian Scientific Center of RAS*. 2022. No. 6(110). Pp. 106–114. DOI: 10.35330/1991-6639-2022-6-110-106-114

## ВВЕДЕНИЕ

При проведении качественного анализа данных в настоящее время часто используют гибридные методы. Для успешного сочетания методов надо иметь четкое представление о правилах принятия решений или о тех областях данных, где эти методы наиболее эффективны. Многие активно используемые в настоящее время подходы не дают четкого представления о том, каким образом было принято решение. Это создает ряд проблем, начиная от доверия качеству результата и заканчивая возможностью коррекции используемого метода. К таким методам относятся очень популярные нейронные сети [1–3]. Исходя из самой структуры нейронной сети, например, сетей глубокого обучения, проследить цепочку рассуждений, приводящую к результату, не представляется возможным, но можно попробовать определить область данных, на которых обученная нейронная сеть будет особенно эффективна и где, возможно, будет ошибаться. Соответственно, разработать алгоритмы коррекции и получить более фундаментальные представления об обрабатываемых данных.

## МАТЕРИАЛЫ И МЕТОДЫ

Будем считать, что в исследуемой области связь признаков и объектов может быть представлена как связь по претендентам, для обработки данных будем использовать  $\Sigma\Pi$ -нейронную сеть и логический анализ данных.

Отсутствие данных допустимо.

Формальная постановка задачи распознавания образов [3–5].

Пусть  $X = \{x_1, x_2, \dots, x_n\}$ ,  $x_i \in \{0, 1, \dots, k_i - 1\}$ , где  $x_i$  – набор признаков. Будем считать, что каждый признак кодируется произвольным дискретным значением  $k_i \in [2, \dots, N]$ ,  $N \in \mathbb{Z}$ ,  $Y = \{y_1, y_2, \dots, y_m\}$  – множество объектов, каждый объект  $y_i$  характеризуется соответствующим набором признаков  $x_1(y_i), \dots, x_n(y_i)$ :  $y_i = f(x_1(y_i), \dots, x_n(y_i))$ .

$$\begin{pmatrix} x_1(y_1) & x_2(y_1) & \dots & x_n(y_1) \\ x_1(y_2) & x_2(y_2) & \dots & x_n(y_2) \\ \dots & \dots & \dots & \dots \\ x_1(y_m) & x_2(y_m) & \dots & x_n(y_m) \end{pmatrix} \rightarrow \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{pmatrix}$$

Нужно найти функцию  $Y = f(X)$ .

Очень часто при решении подобных задач приходится иметь дело с данными, основным источником которых является опыт, а не теория. Поэтому подобрать формальный аппарат для выявления закономерностей является непростой задачей. Будем считать, что формализмы, допускающие на заданном множестве какие угодно связи, кроме тех, которые противоречат заявленному опыту, являются наиболее полными для работы с данными. Построенную на таких рассуждениях функцию назовем полным классификатором данных.

В результате такого рассуждения можно построить булеву функцию от  $n + m$  переменных, значение которой будет равно 0, на тех наборах, где есть признаки объекта, а самого объекта нет, и единице во всех остальных случаях.

Если обучающая выборка состоит из  $k$  элементов, то для описания с помощью булевой функции  $F(x_1(y_i), \dots, x_n(y_i), P^\sigma(y_1), \dots, P^\sigma(y_n))$ , когда

$$P^\sigma(y_i) = \begin{cases} \overline{P(y_i)} & \text{при } \sigma = 0; \\ P(y_i) & \text{при } \sigma = 1. \end{cases}$$

Эта функция принимает значения «0» на множествах, когда данные, характеризующие объект, есть, а сам объект отрицается, т.е.  $(x_1(y_i) \dots x_n(y_i) P^\sigma(y_1), \dots, \overline{P(y_i)} \dots P^\sigma(y_n))$ . И «1» на всех остальных наборах.

Пример.

Пусть заданы следующие соотношения:

**Таблица 1**  
**Table 1**

$x1$	$x2$	$y$
0	1	a
1	1	b

Построим таблицу, определяющую функцию для переменных:  $x_1, x_2, P(y(a)), P(y(b))$ .

**Таблица 2**  
**Table 2**

$x1$	$x2$	$P(a)$	$P(b)$	$f(X, Y)$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

$$f(x_1, x_2, P(a), P(b)) = \overline{x_2} \vee P(a) \& \overline{x_1} \vee P(b) \& x_1$$

Это представление функции после преобразования ее конъюнктивной нормальной формы. Как видно из примера, переменная, которая имеет одно значение для двух объектов, являясь неинформативной, стоит как свободная переменная, а объекты идентифицируются по переменной, которая отличается [5–7].

Представления функции в виде формулы позволяют легче строить конъюнктивную нормальную форму, так как в столбце значений функции нулей будет существенно меньше. В результате мы получим

$$f(X) = \& \left( \bigvee_{i=1}^n \bar{x}_i \vee P(y_j) \right).$$

Учитывая, что

$$\bigvee_{i=1}^n \bar{x}_i(y_j) \vee P(y_j), j \in [1, \dots, m],$$

можно представить как

$$\&_{j=1}^m x_j(y_i, \cdot) \rightarrow P(y_i), i = 1, \dots, l; x_j(y_i) \in \{0, 1, \dots, k-1\},$$

где предикат  $P(y_i)$  принимает значение true, т.е.  $P(y_i) = 1$ , если  $y = y_i$  (объект, соответствующий заданным характеристикам), и  $P(y_i) = 0$ , если  $y \neq y_i$  [4]. Мы получаем, что эта функция есть конъюнкция правил продукции, которые выражают связь «признаки – объект» на заданном наборе данных [8–10].

Вообще для того, чтобы разделить объекты в данном примере, достаточно одного дизъюнкта, такого как  $f(x_1, x_2, P(a), P(b)) = P(a) \& \bar{x}_1$  или  $f(x_1, x_2, P(a), P(b)) = P(b) \& x_2$ . Но мы потеряем часть значений и в случае некорректных или зашумленных данных не сможем распознать наиболее вероятный из представленных объектов. Например, набор  $(1, 0)$ .

Построенная функция является полным классификатором для заданных данных. Подробно ее свойства рассмотрены в работе [4].

Конечно, для больших данных такой подход будет несколько громоздким. Поэтому интересно посмотреть, какая часть этой функции является решением нейронных сетей на заданном множестве данных [5].

В качестве примера будем рассматривать  $\Sigma\Pi$ -нейрон.

$\Sigma\Pi$ -нейрон представляет алгебраическую модель нейрона [6–10], которая обладает лучшими способностями по аппроксимации зависимостей.

Один  $\Sigma\Pi$ -нейрон в своем классическом варианте имеет полилинейную функцию суммарного сигнала

$$sp(X) = out(\theta + \sum w_k \prod_{i=1}^n x_i),$$

где  $\theta$  – константа,  $out$  – функция вывода [12–14].

При обучении объекты кодируются натуральными числами. Для приведенного примера если  $a=1, b=2$ , то  $sp = x_2 + x_1 x_2$ .

Тогда в случае запроса  $(1, 0)$  нейрон выдаст решение «0», логическая функция « $b$ ».

Если посмотреть, как решения нейронной сети будут выглядеть в столбце значений функции таблицы 2, то это будут единицы только в тех наборах, где есть данные, обозначенные в таблице, и есть соответствующий в таблице объект. Он может быть с другими объектами в любом сочетании.

**Таблица 3**  
**Table 3**

$x1$	$x2$	$P(a),$	$P(b)$	$f(X, Y)$
.....	....	.....	....	0

0	1	1	0	1
0	1	1	1	1
.....	....	...	..	0
1	1	0	1	1
1	1	1	1	1

Можно увидеть, что у функции 14 возможных решений, у нейронной сети – 4. Если наши решения замкнуты относительно представленных данных, то возможности логического анализа более широкие.

Для нейронной сети может быть построен логический корректор, один из способов его построения описан в работе [10]. После его применения к  $sp = x_2 + x_1x_2$  получаем следующую функцию:  $f_1(x_1, x_2, P(a), P(b)) = P(a) \& P(b) \& x_2 \vee P(a) \bar{x}_1 \vee P(b) \& x_1$ . Эта функция добавит количество решений в исходной таблице, но полностью ее не восстановит. На запрос (1,0) ответит правильно. Потому что  $f(x_1, x_2, P(a), P(b)) = \bar{x}_2 \vee P(a) \& \bar{x}_1 \vee P(b) \& x_1$  и  $f_1(x_1, x_2, P(a), P(b)) = P(a) \& P(b) \& x_2 \vee P(a) \bar{x}_1 \vee P(b) \& x_1$  – разные функции, но их таблицы истинности пересекаются на наборах, где определены объекты.

Таблица 4  
Table 4

$x1$	$x2$	$P(a),$	$P(b)$	$f(X, Y)$	$\Sigma\Pi$ -нейрон	Корректор
0	0	0	0	1	0	0
0	0	0	1	1	0	0
0	0	1	0	1	0	1
0	0	1	1	1	0	1
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	0	0
1	1	0	1	1	1	1
1	1	1	0	0	0	0
1	1	1	1	1	1	1

Как видно из таблицы, объектная часть функции может быть восстановлена, не удастся восстановить только ту часть, где в качестве дизъюнкта выступают переменные.

Посмотрим более иллюстративный пример 2:

Таблица 5  
Table 5

$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	0	1	0	a(2)

0	1	0	1	$b(4)$
0	1	1	1	$c(8)$
1	1	1	0	$d(16)$

$$sp = 2x_3 + 4x_2x_4 + 2x_4x_2x_3 + 6x_1x_2x_3$$

Для любых данных из таблицы нейрон выдаст верное решение, а если данные будут несколько изменены, например (1111), (0110), то результаты будут сомнительными. В первом случае получим 14, такого объекта нет в наших данных, во втором получим объект под номером 2, хотя вполне мог бы быть и объект под номером 8. Нейронная сеть вообще предлагает единственное решение, и в этом случае логические методы более корректны, потому что они предлагают все возможные решения.

Полный логический классификатор будет выглядеть несколько громоздко:

$$f(x_1, x_2) = x_1x_2 \vee x_1\bar{x}_3 \vee \bar{x}_4\bar{x}_3 \vee x_1x_4 \vee \bar{x}_2\bar{x}_3 \vee b\bar{x}_3 \vee cx_3x_4 \vee dx_1 \vee bcx_4 \vee acdx_3 \vee a\bar{x}_2.$$

После применения метода построения логической функции по структуре  $\Sigma\Pi$ -нейрона

**Таблица 6**  
**Table 6**

$0_1$	$1_1$	$0_2$	$1_2$	$0_3$	$1_3$	$0_4$	$1_4$
$a$		$a$			$a$	$a$	
$b$			$b$	$b$		$b$	
$ab$			$b$	$b$		$c$	
$c$						$bc$	
$abc$			$c$		$c$	$d$	
			$bc$		$ac$	$ad$	
	$d$		$d$		$d$		
			$bcd$		$acd$		

получим функцию  $f_1(x_1, x_2) = dx_1 \vee a\bar{x}_2 \vee b\bar{x}_3 \vee cx_3x_4 \vee bcx_4 \vee b\bar{x}_3 \vee acdx_3$ .

После запроса (1111) получим ответ  $d \vee c$ , после запроса (0110) получим  $acd$ .

Эти ответы соответствуют данным в таблице и улучшают работу нейронной сети.

Однако функция  $f_1(x_1, x_2)$  хоть и расширяет количество решений, но не является полным классификатором данных. Для получения полного классификатора может быть предложен следующий алгоритм.

Пусть  $\{w_1, w_2, \dots, w_k\}$  – множество весов нейрона. Если  $w_i \in Y$ , среди множества распознаваемых объектов есть элемент, пронумерованный  $w_i$ . Тогда все соответствующие этому весу переменные записываем через дизъюнкцию с отрицанием, а те переменные, которых нет в качестве сомножителей этого веса, записываем без отрицания и через дизъюнкцию записываем сам объект.

Например,  $sp = 2x_3 + 4x_2x_4 + 2x_4x_2x_3 + 6x_1x_2x_3$ ;  $w_1 = 2$ . Этому весу соответствует объект  $a$ , тогда записываем следующее правило:  $x_1 \vee x_2 \vee \bar{x}_3 \vee x_4 \vee a$ .

Если очередная весовая величина не соответствует объекту в данных, то нужно сложить приводящие веса, которые содержат переменные, входящие в качестве сомножителей в данное слагаемое.

Таким образом можно получить все правила в исходных данных, а поставив между ними знак конъюнкции, получить функцию – полный классификатор. Для нашего примера это будет

$$f(X, Y) = (x_1 \vee x_2 \vee \bar{x}_3 \vee x_4 \vee a) \& (x_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4 \vee b) \& (x_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4 \vee c) \& (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee x_4 \vee d).$$

Независимо от того, какой подход будет применен для коррекции работы нейронной сети, мы получим функцию, у которой есть дизъюнкции, определяющие конкретные объекты или классы объектов. При определенных запросах такая функция может иметь несколько решений. Один из подходов, который поможет выбрать наиболее подходящее решение, – это логические нейронные сети [15–18].

Входами у таких нейронных сетей будут переменные  $X = \{x_1, x_2, \dots, x_n\}$ . На выходе будет множество объектов. Слоем, производящим вычисления, будут дизъюнкты, в которых операции конъюнкции и дизъюнкции заменены нейронами конъюнктурами и дизъюнктами.

Дизъюнктам, которые содержат информацию об объектах, однозначно сопоставляются нейроны, подающие выходной сигнал на те объекты, которые входят в соответствующие подклассы.

В результате запроса конъюнктур передает сигнал единичного веса на каждый из входящих в этот дизъюнкт (подкласс) объектов [19–20].

Таким образом, для качественного анализа данных можно воспользоваться гибридом нейронных сетей – процедурой перевода структуры нейрона в логическую функцию и корректором в виде логической нейронной сети для полученной функции.

### Выводы

Логический подход к анализу результатов работы нейронных сетей предоставляет уникальную возможность анализа и коррекции получаемых решений. Как было показано, решения нейронной сети – это только часть тех возможностей, которые присутствуют в данных. Для выявления всех возможных, логически непротиворечивых решений необходимо дополнять обученную нейронную сеть корректорами, которые позволят установить логические связи между распознаваемыми объектами. Это существенно повысит качество получаемых решений.

В рассматриваемом методе предлагается выявить логические связи между объектами обучающей выборки, это можно сделать даже без обучающей выборки, анализируя структуру сети. В отличие от метода Attention предлагаемый подход выявляет логические, а не статистические закономерности в данных. Знание логических закономерностей в отличие от статистических позволит провести формализацию и точнее понять природу анализируемых данных.

### REFERENCES

1. Zhuravlev Yu.I. On an algebraic approach to solving problems of recognition or classification. *Problemy kibernetiki* [Problems of cybernetics]. 1978. Vol. 33. Pp. 5–68. (In Russian)

Журавлёв Ю. И. Об алгебраическом подходе к решению задач распознавания или классификации // Проблемы кибернетики. 1978. Т. 33. С. 5–68.

2. Shibzukhov Z.M. Correct aggregation operations on algorithms. *Raspoznavaniye obrazov i analiz izobrazheniy* [Pattern recognition and image analysis]. 2014. Vol. 24. No. 3. Pp. 377–382. (In Russian)

Шибзухов З. М. Корректные агрегирующие операции над алгоритмами // Распознавание образов и анализ изображений. 2014. Т. 24. № 3. С. 377–382.

3. Naimi Ashley I., Balzer Laura B. Multilevel generalization: an introduction to super learning *European Journal of Epidemiology*, 2018. Vol. 33. Pp. 459–464.

4. Lyutikova L.A. Construction of a Logical-Algebraic Corrector to Increase the Adaptive Properties of the  $\Sigma\Pi$ -Neuron. *Journal of Mathematical Sciences*. 2021. Vol. 253. Pp. 539–546.

5. Dimitrichenko D.P., Zhilov R.A. The Method of Construction of Logical Neural Networks on the Basis of Variable-Valued Logical Functions. *Journal of Mathematical Sciences*. 2021. Vol. 253. Pp. 500–505.

6. Haoxiang Wang, Smith S. Big data analysis and perturbation using a data mining algorithm. *Journal of Soft Computing Paradigm*, 2021. Vol. 3(01). Pp. 19–28.

7. Vijesh Joe MrK, Raj Jennifer S. User Recommendation System Dependent on Location-Based Orientation Context. *Journal of Trends in Computer Science and Smart Technology*. 2021. Vol. 3(01). Pp. 14–23.

8. Grabisch M., Marichal J-L, Pap E. Aggregation functions. *Cambridge University Press*, 2009. Vol. 127.

9. Calvo T., Belyakov G. Aggregating functions based on penalties. *Fuzzy sets and systems*, 2010. Vol. 161. No. 10. Pp. 1420–1436. DOI: 10.1016/j.fss.2009.05.012.

10. Mesiar R., Komornikova M., Kolesarova A., Calvo T. Fuzzy aggregation functions: a revision. *Sets and their extensions: representation, aggregation and models*. Springer-Verlag, Berlin, 2008.

11. Yang F., Yang Zh., Cohen W.W. Differentiable learning of logical rules for reasoning in the knowledge base. *Advances in the field of neural information processing systems*, 2017. Pp. 2320–2329.

12. Flach P. *Mashinnoye obucheniye: nauka i iskusstvo postroyeniya algoritmov, kotoryye izvlekayut znaniya iz dannykh* [Machine learning: The science and art of building algorithms that extract knowledge from data]. Moscow: DMK Press, 2015. 400 p. (In Russian)

Флах П. Машинное обучение: наука и искусство построения алгоритмов, которые извлекают знания из данных. Москва: ДМК Пресс, 2015. 400 с.

13. Akhlakur R., Sumaira T. Ensemble classifiers and their applications: a review. *International Journal of Computer Trends and Technologies*. 2014. Vol. 10. Pp. 31–35.

14. Dyukova E.V., Zhuravlev Yu.I., Prokofiev P.A. Methods for improving the efficiency of logical correctors. *Mashinnoye obucheniye i analiz dannykh* [Machine Learning and Data Analysis]. 2015. Vol. 1. No. 11. Pp. 1555–1583. (In Russian)

Дюкова Е. В., Журавлев Ю. И., Прокофьев П. А. Методы повышения эффективности логических корректоров // Машинное обучение и анализ данных. 2015. Т. 1. № 11. С. 1555–1583.

15. Lyutikova L.A., Shmatova E.V. Algorithm for constructing logical operations to identify patterns in data. *E3S Web of Conferences, Moscow, november 25–27, 2020*. Moscow, 2020. Vol. 224. Art. 01009. DOI: 10.1051/e3sconf/202022401009.

16. Lyutikova L.A., Shmatova E.V. Analysis and synthesis of pattern recognition algorithms using variable logic. *Informatsionnyye tekhnologii* [Information technologies]. 2016. Vol. 22. No. 4. Pp. 292–297. (In Russian)

Лютикова Л. А., Шматова Е. В. Анализ и синтез алгоритмов распознавания образов с использованием переменной логики // Информационные технологии. Т. 22. №. 4. 2016. С. 292–297.

17. Burges C.J. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*. 1998. Vol. 2. No. 2. Pp. 121–167.

18. Aladjev V. Computer Algebra System Maple: A New Software Library. *ICCS 2003. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg*, 2003. Vol. 2657.

19. Prepare S.A., Cook S.A., Rekhov R.A. Relative efficiency of systems of proof of statements. *Journal of Symbolic Logic*. 1979. Vol. 44. No. 1.



20. Duda R., Hart P. Pattern classification and scene analysis. *Wiley New York*, 1973. Vol. 3. Pp. 731–739.

21. Shibzukhov Z.M. On a constructive method for synthesizing families of majority-correct algorithms. *Materialy VII Mezhdunarodnoy konferentsii po raspoznavaniyu obrazov i analizu izobrazheniy* [Proceedings of the VII International Conference on Pattern Recognition and Image Analysis]. 2004. Vol. 1. Pp. 113–115. (In Russian)

*Шибзухов З. М. О конструктивном методе синтеза семейств мажоритарно правильных алгоритмов / Материалы VII Международной конференции по распознаванию образов и анализу изображений. 2004. Т. 1. С. 113–115.*

### **Информация об авторе**

**Лютикова Лариса Адольфовна**, канд. физ.-мат. наук, зав. отделом нейроинформатики и машинного обучения, Институт прикладной математики и автоматизации – филиал Кабардино-Балкарского научного центра РАН;

360000, Россия, г. Нальчик, ул. Шортанова, 89 А;

lylarisa@yandex.ru, ORCID: <https://orcid.org/0000-0003-4941-7854>

### **Information about the author**

**Lyutikova Larisa Adolfovna**, Candidate of Physical and Mathematics Sciences, Head of the Dep., Neural Networks and Machine Learning, of the Institute of Applied Mathematics and Automation – branch of the Kabardino-Balkarian Scientific Center of the Russian Academy of Sciences;

360000, Russia, Nalchik, 89 A Shortanov street;

lylarisa@yandex.ru, ORCID: <https://orcid.org/0000-0003-4941-7854>