Journal of Theoretical and Applied Information Technology
15th June 2021. Vol.99. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# A TECHNIQUE FOR ANALYZING NEURAL NETWORKS IN TERMS OF TERNARY LOGIC

[1]IBRAGIM E SULEIMENOV, [2]AKHAT S BAKIROV, [2]DINARA K MATRASSULOVA
[1]Crimean Federal University named by V.I. Vernadsky, Simferopol, Republic of Crimea
[2]Almaty University of Power Engineering and Telecommunications, Almaty, Republic of Kazakhstan
E-mail: [1]esenych@yandex.ru, [2]axatmr@mail.ru, [2]dinara.kutlimuratovna@gmail.com

**ABSTRACT**

There is an extensive class of neural networks, the functioning of which can be described in terms of binary logic: a set of logical variables describing the state of the inputs is associated with a set of logical variables characterizing the state of the outputs. Such networks can be described in terms of logical functions, in particular, through the Zhegalkin polynomial. This imposes significant restrictions on the variability of the neuron weights. This fact is of significant interest from the point of view of overcoming the thesis about the logical opacity of neural networks, which is associated with the most common approaches to training neural networks, which are actually the results of computer experiments. Therefore, it can be considered that neuroscience is predominantly an empirical science, with the only difference that its foundations are not laboratory, but computer experiments. An important step towards overcoming the thesis about the logical opacity of neural networks is to establish restrictions on the variability of the weight coefficients, i.e. proof of the fact that in reality neurons can perform only a limited set of operations that can be reduced to logical ones. At the same time, there is no reason to assert that artificial neural networks must necessarily be built on the basis of the apparatus of binary logic. This paper shows that appliance of ternary logic in combination with a geometric interpretation of the operation of neural networks allows us to reveal the existence of more than strict restrictions on the variability of the weight coefficients of a neural network. An exhaustive description of a neuron with four inputs, which shows how the proposed approach can be extended to the analysis of neurons with an arbitrary number of inputs.

**Keywords:** *Artificial Intelligence, Artificial Neural Networks, Ternary Logic*

## 1. INTRODUCTION

Currently, one of the main tools for creating artificial intelligence systems are neural networks [1,2], which are widely used. Now there are a significant number of teaching methods, which differ significantly in terms of algorithmic basis and purpose [3,4]. However, there is a common distinctive feature characteristic of the vast majority of neural networks used in practice. It is assumed that the weight coefficients that determine the response of an individual formal neuron to a set of logical variables applied to its inputs can change continuously [5]. This fact makes it possible to use various methods of teaching neural networks, which are based on a step-by-step change in the values of the coefficients. In particular, neural networks have become widespread, the training of which are carried out through the use of continuous activation functions, for example, sigmoidal [6]. The only exception is the case of threshold activation functions, but in practice they are practically not used, since the existing methods of training neural networks are mainly focused on continuous activation functions [7]. Obviously, the continuity of the activation functions leads to the fact that the weight coefficients formed as a result of training can also take on values that change continuously.

This approach has proven itself in practice, however, in any attempt to overcome the thesis about the logical opacity of neural networks, which is often repeated in the literature, especially popular [8,9], one way or another, it will be necessary to raise again and again the question of the possibility of direct calculation of weight coefficients of the neural network performing the specified functions. We emphasize that even a partial solution to this issue, for example, limiting the search area for the desired values of the coefficients, is of significant interest, since the complexity of neural networks, the procedures for their training, de facto based on the empirical selection of coefficients, will become increasingly cumbersome. Currently, this issue is becoming undoubtedly relevant, since with the development of artificial intelligence systems [10], mainly using trained neural networks, they begin to perform a variety of functions (such as computer vision, autonomous control of complex systems, analysis and forecast, classification), while becoming more and more complex [11].

It is especially important to overcome the thesis about the logical opacity of neural networks from the questioning the essence of intelligence as such [12]. We emphasize that this question remains largely open, and it is this fact, in particular, that gives a rise to numerous discussions [13] as whether computing and expert systems can be attributed to artificial intelligence or not. As emphasized in [12, 14, 15], this issue will not be removed until the essence of intelligence as such is revealed, for which, among other things, it is necessary to use the methods of the theory of neural networks and the methods of philosophy in parallel, in other words - starting from the thesis about the convergence of natural science and humanitarian knowledge.

Furthermore, the most general proof that very strict constraints can be imposed on the variability of the weight coefficients of a neural network follows directly from the application of the Zhegalkin polynomial to the description of the operation performed by the neurons of the network [16]. Indeed, if an individual neuron of the network associates a set of logical variables describing the state of its inputs with one logical variable that describes the state of its output, then the operation performed by this neuron can be represented in its most general form as

$$f(x_1, x_2, x_3, \dots, x_N) = \sum_{\sigma_1,\sigma_2,\sigma_3,\dots,\sigma_N} f(\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_N) x_1^{\sigma_1} x_2^{\sigma_2} x_3^{\sigma_3} \dots x_N^{\sigma_N} \quad (1)$$

where $N$ is the number of neuron inputs, the summation is carried out over all possible combinations of logical variables in the sequence $(\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_N)$, each of which can take the value of either a logical zero or a logical one, the designation $x_i^{\sigma_i}$ has the following meaning:

$$x_i^{\sigma_i} = \begin{cases} x_i, & \sigma_i = 1 \\ \bar{x}_i, & \sigma_i = 0 \end{cases} \quad (2)$$

where the symbol $\bar{x}_i$ denotes the operation of logical inversion

$$\bar{x}_i = 1 + x_i \quad (3)$$

Formula (1), in particular, shows that there are no more than $2^{2^N}$ possible options describe the functioning of a particular neuron of the type under consideration, and this number does not depend on the type of activation function. This also implies to the fact that there is a countable number of possible combinations of weight coefficients, which does not exceed $2^{2^N}$.

This conclusion, however, does not in itself have significant practical significance, since we are talking about combinatorically large numbers. Nevertheless, it is possible to narrow the variability of the neural network weight coefficients further, if we consider that the indicator $2^{2^N}$ corresponds to the number of logical operations at all, i.e. here, the specificity of operations performed by neurons is not considered, where the weighted sum of values describing the state of its inputs is calculated [2].

Based on the methods of projective geometry, such attempt was made in [17]. It was shown that while constructing a neural network, the weight coefficients that specify the operation of an individual neuron can be selected from a certain finite set of sequences. Specifically, for each j-th neuron of the network, a certain set of weight coefficients

$$W_j = (w_{j1}, w_{j2}, \dots, w_{jN}), \quad (4)$$

obtained in the process of training a neural network can be replaced with a set

$$W_j^0 = (w_{j1}^0, w_{j2}^0, \dots, w_{jN}^0), \quad (5)$$

where the values $w_{jN}^0$ take on certain discrete values.

However, the indication of a specific set, which completely exhausts the choice of sequences of the form (5), when using the methods of projective geometry, encounters certain computational difficulties. Moreover, these methods are not very clear.

In this work, an attempt is made to develop a visual method for constructing a set that exhausts admissible sequences of the form (5). It is shown that the construction of such a set corresponds to the transition to the description of considering type of neural networks in the language of ternary logic.

Our proposed approach creates the basis for a significant modernization of the basic approaches to the creation of artificial intelligence systems, at least those of them that are based on the use of neural networks. Indeed, at present, the overwhelming majority of results in this area have been obtained on a purely empirical basis using computer experiments. The true algorithm for the functioning of an empirically tuned neural network remains unknown, which makes many authors talk about their logical opacity. However, reducing the operations performed by the neural network to logical ones will allow, at least, to explicitly reveal these algorithms, i.e. establish the set of logical operations that the neural network performs. At the next stage, this approach will allow us to proceed to the construction of explicitly written algorithms for the functioning of neural networks.

## 2. ESTIMATION OF THE VARIABILITY OF WEIGHT COEFFICIENTS` SETS BASED ON THE NEURONS FUNCTIONING DESCRIPTION USING TERNARY LOGIC

Consider a basic expression describing the functioning of an individual neuron in a network with N inputs.

$$Y = \theta(w_1 X_1 + w_2 X_2 + \cdots + w_N X_N) \qquad (6)$$

From the point of view of approach being developed, in which logical operations performed by neurons are de facto considered, without loss of generality, we can assume that in this record the activation function is stepwise

$$\theta(x) = \begin{cases} -1, & x \leq 0 \\ +1, & x > 0 \end{cases} \qquad (7)$$

The proof of this statement will be given below.

For the convenience of further use, we will assume that all variables appearing in formula (6) and similar ones can only take discrete values -1 and +1.

$$Y_i, X_i \in (-1, +1) \qquad (8)$$

The choice of these values is dictated only by geometric analogies, which will be used in the future. Such a choice of variables does not in any way affect the generality of its consideration, since it is always possible to establish a correspondence between the above values and logical variables in accordance with the formula

$$\begin{cases} -1 \rightarrow \text{logical } 0 \\ +1 \rightarrow \text{logical } 1 \end{cases} \qquad (9)$$

The values ± 1 can also be considered as values that a ternary logic variable can take or as elements of the Galois field $GF(3)$, on which the following addition and multiplication operations are defined.

$$1 + 1 = (-1);\ (\pm 1) + 0 = (\pm 1);\ (-1) + (-1) = 1 \qquad (10)$$

$$(\pm 1) \cdot 1 = (\pm 1);\ (\pm 1) \cdot 0 = 0;\ (-1) \cdot (-1) = 1 \qquad (11)$$

Formula (10) is actually the rules for adding integers modulo 3.

Note that formula (6) is often used in the form

$$Y = \theta(w_1 X_1 + w_2 X_2 + \cdots + w_N X_N + X_0) \qquad (12)$$

However, if a neuron with an arbitrary number of inputs is considered, the value $X_0$ can be excluded from consideration without loss of generality, i.e., put it equal to zero.

Formula (6) admits a transparent geometric interpretation [12]. Namely, consider the hyperplane

$$w_1 X_1 + w_2 X_2 + \cdots + w_N X_N = 0 \qquad (13)$$

This hyperplane divides the $N$-dimensional space $\mathbb{R}^N$i into two half-spaces. If a point with coordinates $(X_1, X_2, \ldots, X_N)$ falls into one of these half-spaces, then function (6) takes on the value $-1$, if in the other, then $+1$. Moreover, since $X_0 = 0$, then plane (13) passes through the origin.

In fact, we are talking about the fact that the hyperplane (13) cuts the hypercube with edge 2 into two equal parts. Accordingly, the variability of the sequences of weight coefficients, as noted in [17], is determined by the topology of the mutual arrangement of those vertices that remain in the same half-space. There is a fairly wide range of changes in the specific values of the weighting factors, at which the above topology will remain unchanged, i.e., from the point of view of neuron functioning, two planes dividing the set of hypercube vertices into the same subsets can be considered as insignificantly different. We emphasize that this applies only to hyperplanes in general position, i.e., to those that do not pass through the vertices of the hypercube. If this condition is not satisfied, then a small change in the coordinates of the hyperplane leads to qualitative differences between the subsets indicated above.

The hyperplane equation can be specified in various ways. In particular, if such a plane passes through the origin, then it can be given by an equation of the form

$$det\begin{pmatrix} x_1 & x_2 & \cdots & x_N \\ a_1^2 & a_2^2 & \cdots & a_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{N-1} & a_2^{N-1} & \cdots & a_N^{N-1} \end{pmatrix} = 0 \quad (14)$$

where $(a_1^j, a_2^j, \dots, a_N^j)$ are the coordinates of $N-1$ fixed points in the space $\mathbb{R}^N$ and $(x_1, x_2, \dots, x_N)$ are the current coordinates included in the hyperplane equation.

Indeed, if the points with coordinates $(a_1^j, a_2^j, \dots, a_N^j)$ and $(x_1, x_2, \dots, x_N)$ lie in the same hyperplane, this means that the equations

$$\begin{cases} w_1 a_1^j + w_2 a_2^j + \cdots + w_N a_N^j = 0 \\ w_1 x_1 + w_2 x_2 + \cdots + w_N x_n = 0 \end{cases}, j = 1,2, \dots, N-1 \quad (15)$$

This system of equations can be considered as equations for the coordinates of the straight line $(w_1, w_2, \dots, w_N)$. This system is homogeneous; therefore, it has a solution only if its determinant is zero, which is expressed by formula (14). We also emphasize that system (15) includes $N-1$ sets of coordinates corresponding to $N-1$ points in the space $\mathbb{R}^N$. This corresponds to the fact that one and only one plane can be drawn through $N$, which coordinates do not correspond to linearly independent vectors. So, in three-dimensional space through three points that do not lie on one straight line, one and only one plane can be drawn.

Expanding determinant (14) along the top row, we find that the coordinates of the straight line are related to the coordinates of the point $(x_1, x_2, \dots, x_N)$ by a relation of the form (13), and there is

$$w_i = A_{1i}, i = 1,2, \dots, N \quad (16)$$

where $A_{1i}$ are the algebraic complements of the elements of the first row of the determinant (14), in particular,

$$A_{11} = det\begin{pmatrix} a_1^2 & \cdots & a_N^2 \\ \vdots & \ddots & \vdots \\ a_1^{N-1} & \cdots & a_N^{N-1} \end{pmatrix} \quad (17)$$

You can select points that define a hyperplane in different ways. In particular, all these points can be chosen on the edges of a hypercube with edge 2, all vertices of which have coordinates

$$Q_{\vec{\imath}}^j = (b_1^j, b_2^j, \dots, b_N^j); b_i^j = \pm 1; \; j = 1,2, \dots, N-1 \quad (18)$$

where multi-index $\vec{\imath}$ and specifies a specific vertex (specific sequence of characters).

Moreover, as it follows, among other things, from the results of [17], planes in general position can be specified by points lying exactly in the center of each of the edges. We emphasize that for the purposes pursued, it is sufficient to consider only planes in general position, i.e., those that, at small displacements of the reference points, will ensure the division of hypercube`s vertices set into the same subsets.

The coordinates of such points are specified by sequences of the form (18) with the difference that one and only one of the coordinates takes on a value equal to zero. The selection of a specific edge can be displayed by the following entry

$$Q_{\vec{\imath},k}^j = (b_1^j, b_2^j, \dots, b_N^j); b_{i \neq k}^j = \pm 1; \; b_k^j = 0; \; j = 1,2, \dots, N-1 \quad (19)$$

Let us show that such a choice of points defining the coordinates of the hyperplanes characterizing the functioning of neurons allows us to estimate the variability of the neural networks` weight coefficients. This will prove the stated statement that much more serious restrictions are imposed on their variability than it follows from the direct application of the Zhegalkin polynomial [16].

Consider expressions for algebraic complements (17), considering that the rows in matrix (14), starting with the second, satisfy condition (19). We also consider that the determinants of the form (17) have a direct geometric interpretation. This is the volume of a parallelepiped, the edges of which are specified by vectors, the coordinates of which correspond to the rows of the determinant of the form (17) in the $N$ -dimensional space. All these vectors are located inside a hypercube with edge 2 and center at the origin. Therefore, for the volume of a given body, the estimate is valid

$$|V_N| \lesssim 2^{N+1} \quad (20)$$

Considering the fact that the number of coordinates of a hyperplane in an N-dimensional space is exactly N, for the variability of the sequences of weight coefficients of a neuron with N inputs the following estimate is valid

$$q \lesssim 2^{(N+1)N} \quad (21)$$

This assessment follows from the following considerations. The volume in expression (20) is taken modulo, since the calculation of the determinant of the form (17) gives the oriented volume, i.e. the value of an individual weighting factor varies within

$$-2^{N+1} \lesssim w_j \lesssim 2^{N+1} \quad (22)$$

At the same time, the values of the weight coefficients calculated through algebraic additions of the form (17), when choosing the reference points at the center of the edges of the 2-hypercube, are certainly integers, i.e. it follows from (22) that the number of possible options is indeed determined by the number $2^{N+1}$. Estimate (21) is obtained if we consider that in order to define the entire plane, $N$ points are needed, i.e. q is the variability of the sequence as a whole, but not of the individual coefficient.

Note that the obtained estimate (21) is indeed much more stringent than the one that follows directly from the use of the Zhegalkin polynomial, given by the exponent $2^{2^N}$. In reality, this calculation is also overestimated, which can be demonstrated by specific examples of using the proposed approach to the geometric classification of neurons, which is actually based on ternary logic. The term "ternary logic" is used in the following sense: all possible sequences of the weighting coefficients of an individual neuron in the network are marked with a set of sequences in which only three values -1, 0 and +1 appear. This corresponds to the use of ternary logic [18, 19], which will be used below to prove the possibility of extending the developed approach to neural networks with arbitrary activation functions with a limited transition region.

## 3. NEURONS WITH FOUR INPUTS: GEOMETRIC CLASSIFICATION

Let's consider the application of the proposed approach to the description of a neuron with four inputs. As noted in [17], the situation is qualitatively different depending on whether the neuron has an even or odd number of inputs, and a neuron with three inputs was analyzed in detail in the cited work.

The classification of neurons by the nature of the weighting coefficients` sequences can indeed be given in geometric language, which is, not surprisingly, the most illustrative for the case of neurons with three and four inputs.

Since we are considering a 4-hyperplane passing through the origin, cutting a four-dimensional cube into two identical parts, it can be visualized using a three-dimensional cube. It displays 8 vertices out of 16; the plane under consideration passes symmetrically relative to the other eight vertices. Otherwise, this three-dimensional cube is formed by those vertices of the 4-cube, for which the value of the fourth coordinate $x_4$ is equal to 1.

We will classify triples of points defining a 4-hyperplane according to the following scheme.

In Figure- 1 shows the top face of the considering cube, two points on it are marked, a starting point to construct hyperplanes belonging to different classes. Let us take as the "initial" position of the hyperplane where the upper face of the considering cube lies in it.

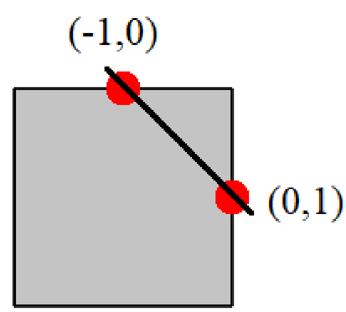


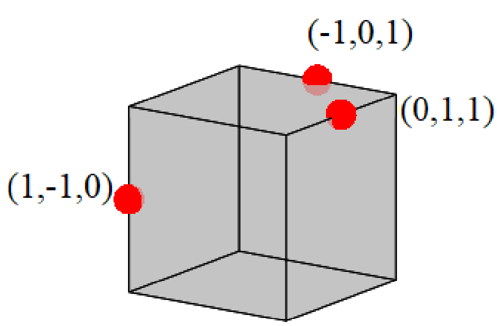

*Figure 1: The scheme of the points location, used to construct a geometric classification of neurons with four inputs, on the upper edge (Position 1).*

We will rotate this plane around an axis passing through the points shown in Fig. 1 until the plane passes through a point located in the center of one of the faces. The corresponding arrangement of the

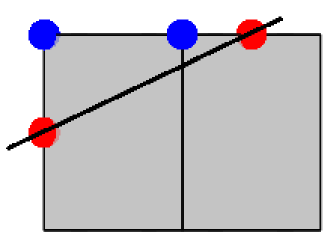
Journal of Theoretical and Applied Information Technology
15th June 2021. Vol.99. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

points defining the hyperplane is shown in Fig. 2a. Figure 2b shows the projection of the location of these points on the plane passing through the diagonal of the upper face and parallel to the vertical edges of the cube. Figure 2 emphasizes that in this position of the hyperplane, it cuts off three vertices of the three-dimensional cube corresponding to the value $x_4 = 1$, i.e., the scalar product of the radius vectors of these three points in the 4-space by the vector $\vec{w} = (w_1, w_2, w_3, w_4)$ is positive. Accordingly, the same 4-plane cuts off 5 vertices of the three-dimensional cube corresponding to the value $x_4 = -1$. Of course, provided that the above vector $\vec{w}$ changes sign, in one case 5 and 3 vertices of three-dimensional cubes corresponding to the values $x_4 = 1$ and $x_4 = -1$, respectively, become clipped. The value of the coordinate $x_4 = 1$ is not shown in Figure 2, since it is the same for all selected points




a)



b)

*Figure 2: The scheme of the points location on the edges of the cube, in which the hyperplane specified by them cuts off 3 or 5 points (at position 1 of the original two points, Fig. 1); three-dimensional construction (a) and construction in projection (b).*

With this choice of the points location, the 4-plane equation takes on a specific form

$$det\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ 0 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 \\ 1 & -1 & 0 & 1 \end{pmatrix} = 0 \qquad (23)$$

Accordingly, the coordinates of the vector $\vec{w}$ are given by the expressions

$$w_1 = \det\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ -1 & 0 & 1 \end{pmatrix} = 1 - 1 + 1 = 1 \qquad (24)$$

$$-w_2 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} = 0 + 2 - 1 = 1 \qquad (25)$$

$$w_3 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & -1 & 1 \end{pmatrix} = 0 + 2 + 1 = 3 \qquad (26)$$

$$-w_4 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix} = 0 + 1 + 1 = 2 \qquad (27)$$

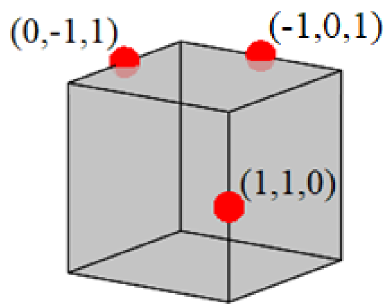
Or

$$\vec{w} = (1, -1, 3, -2) \qquad (28)$$

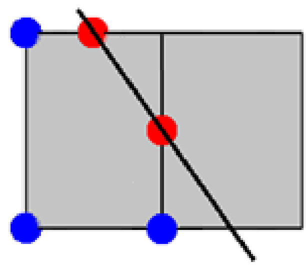
We emphasize that the obtained result is de facto related to the whole class of 4-planes. Namely, we change the sign of all elements of the second column of matrix (23), except for what is in the first row, i.e., consider the 4-plane given by the equation

$$det\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ 0 & -1 & 1 & 1 \\ -1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} = 0 \qquad (29)$$

The location of the points defining this 4-plane is shown in Fig. 3. The figure emphasizes that these points are obtained from the initial configuration (Fig. 2) by the operation of reflection relative to the coordinate plane $x_2 = 0$.




a)



b)

*Figure 3: The configuration of points located on the edges of the three-dimensional cube, obtained by the operation of reflection of the configuration shown in Fig. 2, reflection relative to the plane $x_2 = 0$); three-dimensional construction (a) and construction in projection (b).*

All determinants of the form (24) - (27), by ensuring the calculation of the coordinates of the 4-plane, change sign, since when passing to matrix (29) from matrix (23) all elements of one of their columns change sign. The only exception is the $w_2$, component, since the column that changed the sign in the corresponding determinants does not appear. Accordingly, vector (28) turns into vector

$$\vec{w} = (1,1,3,-2) \tag{30}$$

This reflection operation can be applied to other columns, to pairs of columns, or to all of them at once. All these operations correspond to certain symmetry operations applied to the triplet of points shown in Fig. 2, which leave their relative position unchanged.

Hence it follows that the chosen topology does indeed generate a whole class of 4-planes, which in general case are characterized by vectors

$$\vec{w} = (\pm1, \pm1, \pm3, \pm2) \tag{31}$$

where characters can be chosen in arbitrary combinations.

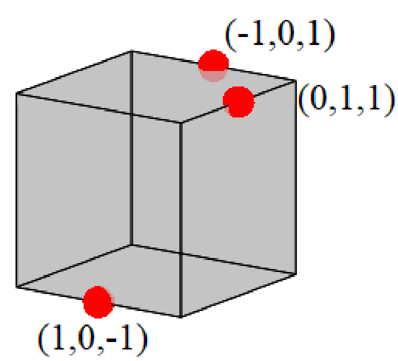
Following that approach, we can summarize the result obtained in [12], where it was shown that all non-degenerate neurons with three inputs de facto correspond to vectors

$$\vec{w} = (\pm1, \pm1, \pm1) \tag{32}$$

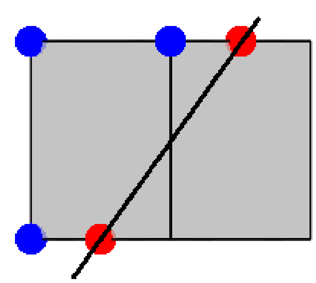
where characters can also be chosen in arbitrary combinations.

The same conclusion proves that it is really possible to propose a geometric classification of neurons based on the analysis of the topology of the mutual arrangement of points defining a hyperplane dividing a hypercube centered at the origin into two parts.

The next configuration of points defining a 4-plane can be reached further by rotating the plane passing through the points shown in Fig. 1, until the next intersection with a point located in the center of the edge of the three-dimensional cube (Fig. 4).




a)



b)

*Figure 4: The scheme of the location of points on the edges of the cube, in which the hyperplane specified by cuts off 4 points (at position 1 of the original two points, Fig. 1); three-dimensional construction (a) and construction in projection (b).*

The following equation of the 4-plane corresponds to such an arrangement of three reference points.

$$det\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ 0 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & 1 \end{pmatrix} = 0 \qquad (33)$$

In accordance with the above, the coordinates of this hyperplane are determined by the following expressions.

$$w_1 = \det\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & -1 & 1 \end{pmatrix} = 2 \qquad (34)$$

$$-w_2 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \end{pmatrix} = 2 \qquad (35)$$

$$w_3 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} = 2 \qquad (36)$$

$$-w_4 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & -1 \end{pmatrix} = 0 \qquad (37)$$

From where

$$\vec{w} = (\pm 2, \pm 2, \pm 2, 0) \qquad (38)$$

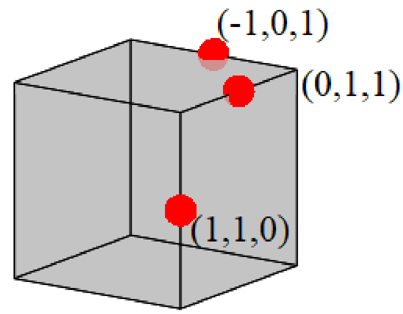
Equation (38) immediately considers that this configuration can be transformed using the operation of reflection relative to the coordinate planes. There is a well-defined class of 4-planes, the coordinates of which differ in the choice of a combination of signs in formula (38); it can be anything.

It is essential that the neuron corresponding to such a hyperplane is degenerated, i.e., the number of its entries is de facto three, not four. Indeed, substituting expression (38) into formula (6) describing the functioning of a neuron, we obtain

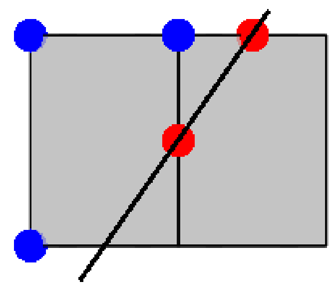
$$Y = \theta(w_1 X_1 + w_2 X_2 + w_3 X_3), \qquad (39)$$

which corresponds to a neuron with three inputs.

To illustrate the fact that the coordinates of the 4-plane do not depend on the choice of points on the edges of the cube (if these points lie in this plane), consider the arrangement of the defining points shown in Fig. 5. We emphasize that this example, despite the apparent obviousness, demonstrates a very remarkable fact: the coordinates of a hyperplane do not depend on the choice of points on the edges of the cube, if only the set of vertices cut off by this plane remains unchanged. Otherwise, we are de facto talking about the choice of the vertices` set cut off by the hyperplane, which specific edges are used to mark such a partition of the vertices set, is secondary.




a)



b)

*Fig. 5. An alternative of the points location on the edges of the cube, in which the hyperplane specified by them cuts off 4 points (at position 1 of the original two points, Fig. 1); three-dimensional construction (a) and construction in projection (b).*

The hyperplane defined by such an arrangement of points is described by the following equation

$$det\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ 0 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} = 0 \qquad (40)$$

that is, the coordinates of the 4-plane are given by the expressions

$$w_1 = \det\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} = 1 \qquad (41)$$

$$-w_2 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} = 1 \qquad (42)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$w_3 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} = 1 \qquad (43)$$

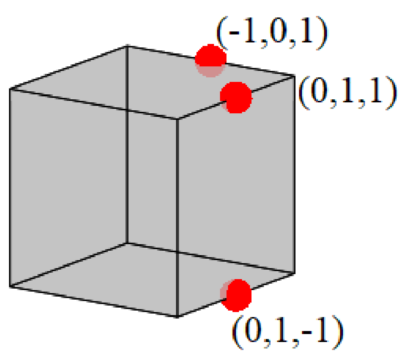
$$-w_4 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} = 0 \qquad (44)$$

As a result,

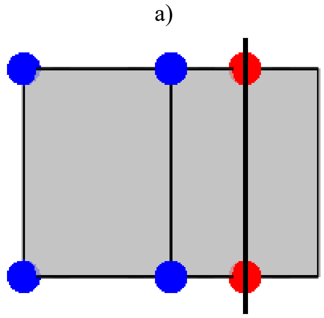
$$\vec{w} = (\pm 1, \pm 1, \pm 1, 0) \qquad (45)$$

The possibility of choosing an arbitrary combination of signs in formula (45) is determined by the same symmetry considerations as above. It is also seen that expression (45) coincides with expression (38) up to a constant factor, i.e., configuration Fig. 4 and Fig. 5, as one would expect, do indeed define the same plane.

The next class of 4-planes can also be found in the same way as above. The corresponding configuration of three points defining a 4-plane is shown in Fig. 6.




a)



b)

*Figure 6: The scheme of the points location on the edges of the cube, in which the hyperplane specified by them cuts off 6 (or 2) points (at position 1 of the original two points, Fig. 1); three-dimensional construction (a) and construction in projection (b).*

The equation of the 4-plane corresponding to such an arrangement of the setting points, as shown in Fig. 6, has the form

$$det\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ 0 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 \\ 0 & 1 & -1 & 1 \end{pmatrix} = 0 \qquad (46)$$

Accordingly, the coordinates of the straight line are given by the expressions

$$w_1 = \det\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & -1 & 1 \end{pmatrix} = 2 \qquad (47)$$

$$-w_2 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 1 & 1 \\ 0 & -1 & 1 \end{pmatrix} = 2 \qquad (48)$$

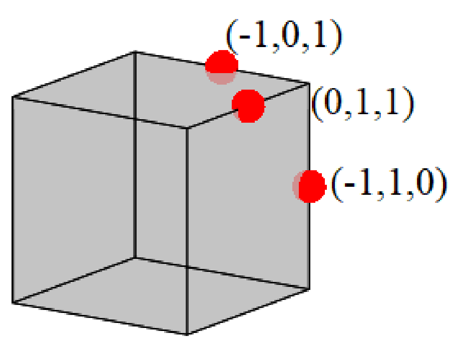
$$w_3 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} = 0 \qquad (49)$$

$$-w_4 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix} = -2 \qquad (50)$$

$$\vec{w} = (\pm 2, \pm 2, 0, \pm 2) \qquad (51)$$

It can be seen that a neuron with weight coefficients corresponding to the configuration in Fig. 6 is also degenerated, i.e., it de facto has not four, but three entrances.

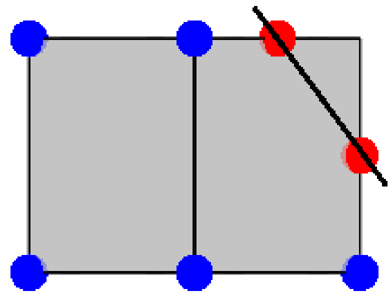
The last possible configuration of the considered type, obtained according to the same scheme as above, is shown in Fig. 7.




a)

Journal of Theoretical and Applied Information Technology
15th June 2021. Vol.99. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



b)

*Figure 7: The scheme of the points location on the edges of the cube, in which the hyperplane specified by them cuts off 7 (or 1) points (at position 1 of the original two points, Fig. 1); three-dimensional construction (a) and construction in projection (b).*

The equation of the 4-plane corresponding to such an arrangement of the setting points, as shown in Fig. 7, has the form

$$det\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ 0 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 \\ -1 & 1 & 0 & 1 \end{pmatrix} = 0 \qquad (52)$$

The coordinates of the 4-plane are given by expressions completely similar to those written above

$$w_1 = \det\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} = 1 \qquad (53)$$

$$-w_2 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & 0 & 1 \end{pmatrix} = 1 \qquad (54)$$

$$w_3 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & 1 & 1 \end{pmatrix} = -1 \qquad (55)$$

$$-w_4 = \det\begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{pmatrix} = -2 \qquad (56)$$

The vector defining the 4-plane, respectively, is given by the expression, where symmetry considerations are also considered, as above.

$$\vec{w} = (\pm 1, \pm 1, \pm 1, \pm 2) \qquad (57)$$

Thus, among all the classes of 4-planes considered, there are only two non-degenerate ones. This fact can be interpreted as follows. Arrangements of reference points, such as shown in Fig. 6, assume that the 4-plane is parallel to one of the edges of both the 3-cube and the 4-cube. Therefore, it is parallel to all those edges that are parallel to this particular edge. This means that when you move the point which coordinates are substituted into formula (6) along any of these edges, the result given by the specified formula will not change. Otherwise, the result of the calculation by this formula does not depend on one of the coordinates of a point on a 4-cube (or a cube of greater dimension with a different number of inputs). From a computational point of view, this input is "disabled", which corresponds to a degenerate neuron.

Let us show that these two non-degenerate combinations, in essence, exhaust the description of all non-degenerate neurons with four inputs.

The geometric classification presented above is based on two points selected on the face of a 4-cube that meets the condition

$$x_3 = x_4 = 1 \qquad (58)$$

Exactly in the same way, you can carry out the same constructions, but starting from other faces

$$x_{i_1} = x_{i_2} = 1 \qquad (59)$$

Passing to other faces leads to the same expressions for determinants (23), (33), etc. with the difference where the position of the $x_i$ variables in the top line should be swapped. It is easy to see that this will only lead to a change in the order of the values $\pm 1$, $\pm 2$, and $\pm 3$ in the expressions for the vectors $\vec{w}$, in particular, in expression (57). Note also that the faces identified by a condition of the form (58), but with a change in sign to the opposite, can be ignored separately, since the corresponding arrangement of the setting points appears automatically when using symmetry operations, justifying the variability of signs in expressions of the form (57).

The same reasoning is applicable to another initial arrangement of points on a cube face that satisfies condition (58), Fig. 8 (Position 2). One of the possible locations of points defining a 4-plane, which is obtained by attaching one more point to Position 2, is shown in Fig. 9.

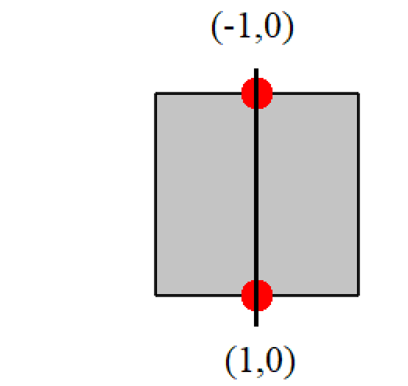
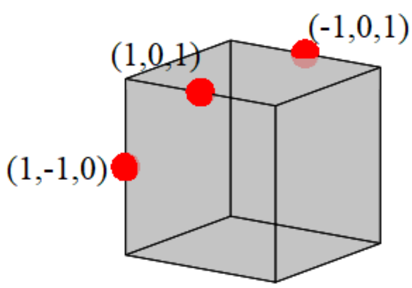
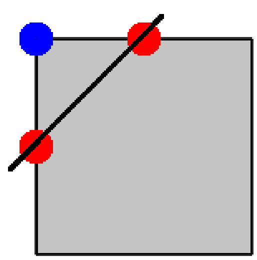



*Figure 8: Additional scheme of the points location, used to construct a geometric classification of neurons with four inputs, on the upper face ("Position 2").*




a)



b)

*Figure 9: The scheme of the points location on the edges of the cube, in which the hyperplane specified by them cuts off 2 or 6 points (at position 2 of the original two points, Fig. 8); three-dimensional construction (a) and construction in projection (b).*

It can be argued in advance that such an arrangement of points will lead to a degenerated neuron. More precisely, any configuration of three reference points, which can be reached, starting from "Position 2", leads to degenerating neurons spread. This follows from the fact that two original points themselves do not lie in parallel to one of the edges.

This can be proved directly, again by composing the equation of the 4-plane through the determinant.

We have:

$$det\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ -1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & -1 & 0 & 1 \end{pmatrix} = 0 \qquad (60)$$

Accordingly, the coordinates of the 4-plane under consideration are given by the expressions

$$w_1 = \det\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ -1 & 0 & 1 \end{pmatrix} = 0 \qquad (61)$$

$$-w_2 = \det\begin{pmatrix} -1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} = -2 \qquad (62)$$

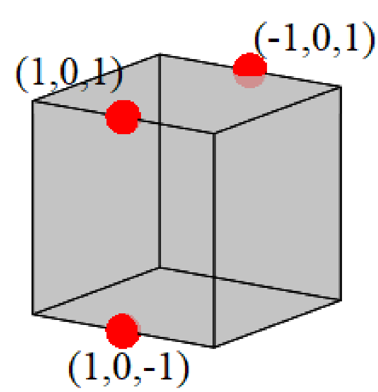
$$w_3 = \det\begin{pmatrix} -1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & -1 & 1 \end{pmatrix} = 0 \qquad (63)$$

$$-w_4 = \det\begin{pmatrix} -1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix} = 0 \qquad (64)$$

It can be seen that, as expected, the normal vector to the given 4-plane corresponds to a degenerated neuron, specifically

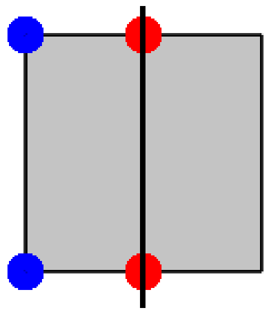
$$\vec{w} = (0, \pm 2, 0, 0) \qquad (65)$$

By direct calculation, it can be shown that the neurons corresponding to the configurations of the control points shown in Fig. 10 and Fig. 11 are also degenerated.



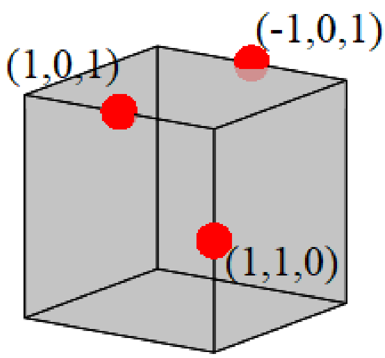

a)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

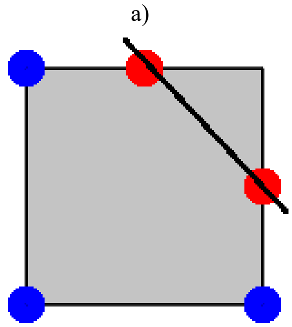


b)

*Figure 10: The scheme of the points location on the edges of the cube, in which the hyperplane specified by them cuts off 4 points (at position 2 of the original two points, Fig. 8); three-dimensional construction (a) and construction in projection (b).*

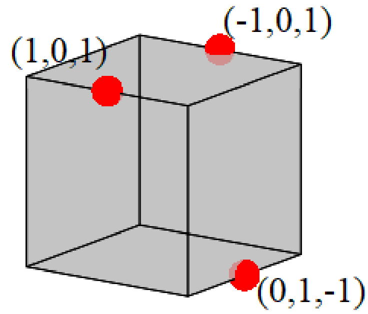



a)

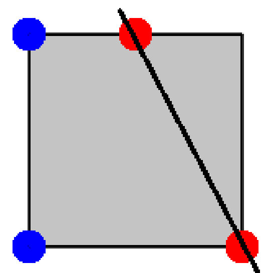


b)

*Figure 11: The scheme of the points location on the edges of the cube, in which the hyperplane specified by them cuts off 6 or 2 points (at position 2 of the original two points, Fig.8); three-dimensional construction (a) and construction in projection (b).*




a)



b)

*Figure 12: An example of a scheme for the points location on the edges of a cube that define a plane of non-general position*

Thus, even more significant restrictions are imposed on the variability of the sequences of weight coefficients corresponding to neurons with four inputs, than it follows from the estimate (22). The choice of possible options is de facto limited only by sets of the form (31) and (57), in which it is allowed to change the places of the values $w_i$ - the coordinates of the 4-plane.

## 4. TO THE METHOD OF CUTTING THE HYPERCUBE FACE IN TRAINING NEURAL NETWORKS: THE POSSIBILITY OF USING TERNARY CODES

In the previous section, in fact, a method of "cutting edges" was proposed, which in the future can be used to train neural networks of at least certain types (for example, those that are designed to solve classification problems associated with the use of neurons with a relatively small number of inputs [2]).

Of course, a separate example of the analysis of a neuron with four inputs, albeit a detailed one, cannot serve as a justification for this method, however, additional considerations can be made that confirm the advisability of its further development.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Let's turn to Fig. 13, which shows a graph corresponding to a four-dimensional cube. The edges that define the cutting of this graph by means of the plane given by equation (23) are marked.

The diagram in Fig. 13 is actually selected one of the four-dimensional hypercube vertices with coordinates (1, -1, 1, -1). This is a vertex which scalar product by the normal vector to the considered plane (28) reaches its maximum value. Since all coordinates of any of the hypercube vertices are equal to $\pm 1$, the choice of the coordinates of the vertex is actually determined only by the alternation of signs in formula (28).

Scheme Fig. 13 emphasizes the following circumstances. A hyperplane corresponding to any combination of weights divides the set of vertices into two subsets, each of which contains the same number of elements. The enumeration of the elements of each these subsets can be given in terms of Hamming coding distances (which once again returns to the question of the relationship between the theory of neural networks and the theory of error-correcting coding [15]).

In particular, Fig. 13 emphasizes that the same set to which the vertex with coordinates (1, -1, 1, -1) belongs, and the vertices which coordinates differ from the vertex coordinates (1, -1, 1, -1) by the Hamming distance, equal to 1.

The peculiarity of a neuron with four inputs is that this set must also include three vertices that are spaced from the vertex (1, -1, 1, -1). Three other vertices, also spaced from the indicated one by Hamming distance 2, will enter another set due to obvious symmetry, as Fig. 13. Any other neurons with an even number of inputs will have the same feature. This follows from the fact that for an even number of neurons, the maximum Hamming distance on the corresponding hypercube is also even, therefore, when constructing a diagram similar to Fig. 13, vertices will appear on it, spaced from both the selected one and from it symmetric at the same Hamming distance.

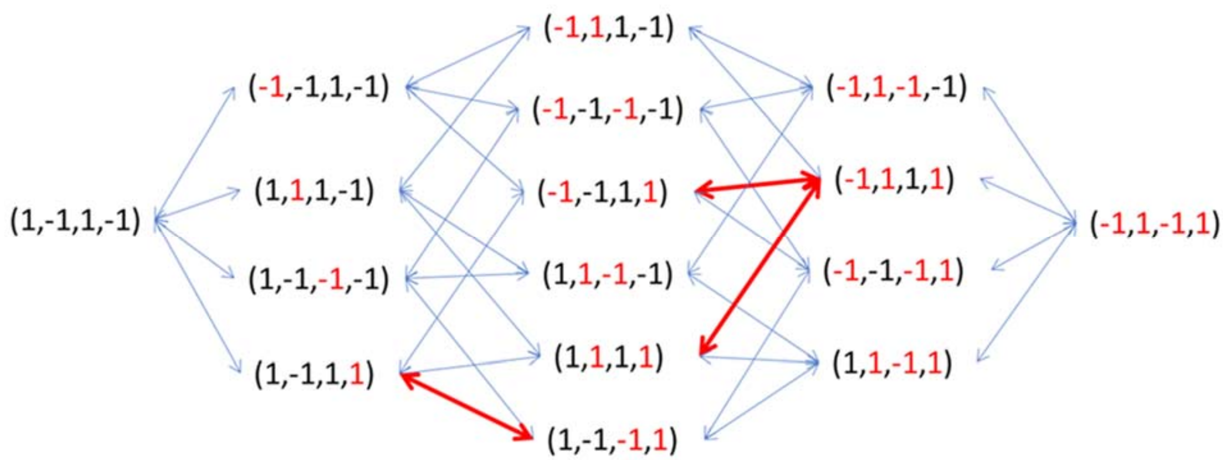



*Figure 13: The graph corresponding to the vertices of the four-dimensional cube, the edges, cutting along which defines the plane given by equation (23), are marked.*

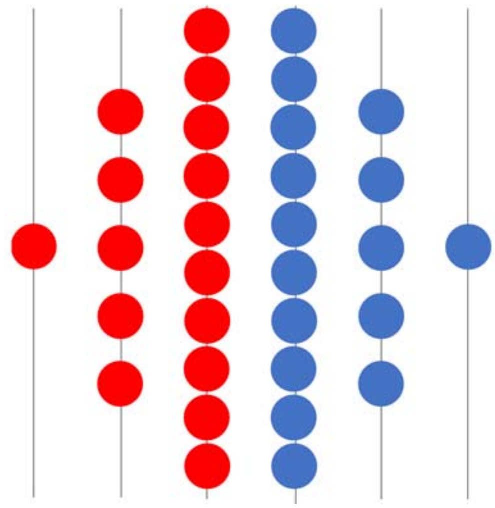


*Figure 14: Scheme corresponding to cutting the graph of a 5-dimensional cube, corresponding to the selection one of the vertices.*

On the contrary, if the number of inputs of a neuron is even, then the selection of one – arbitrary – vertex generates a well-defined division of vertices into two subsets according to the Hamming distance (Fig. 14).

Moreover, there is a close and quite obvious connection between the edges that the hyperplane cuts and the coordinates of such a selected vertex, and this connection takes place for neurons with the number of inputs of any parity. This connection is most conveniently reflected in terms of ternary sequences and, consequently, ternary logic. Indeed, the method used above actually assumes a well-defined identification of the edges of the hypercube – through the coordinates of their center points. In other words, sequences in which only coordinate values equal to $\pm 1$ appear to mark the vertices of the hypercube. Similar sequences, in which all elements except one are equal to $\pm 1$, and this one element is equal to 0 – the edges of the hypercube, etc.

More precisely, the number of zeros in such a sequence corresponds to a certain geometric element of the hypercube.

In the most illustrative three-dimensional case, there is no sequence of the form $(a_1, a_2, a_3)$,, where all $a_i = \pm 1$ mark the vertex. If one of the elements $a_i$ vanishes, it marks an edge, if two – a face. A sequence in which all elements are equal to zero corresponds to the coordinates of the center of the cube and can be considered to mark the cube as a whole.

Therefore, using ternary sequences, you can determine an analogue of the Hamming distance to determine the distance between any geometric elements of the hypercube.

It is easy to show that the points marking the edges, through which the plane cutting the hypercube into two parts passes, will lag behind the selected (in the sense of Fig. 13) vertex by the maximum distance.

In particular, if you first specify only one edge, through which the hyperplane will pass, cutting the hypercube into two parts, then a certain set of vertices will already be selected, which are located at the maximum distance from its center.

Selecting the next edge narrows this set, etc., thus, it becomes possible to implement a learning algorithm for neural networks with threshold activation functions, starting from the analog of the Hamming distance for ternary sequences. Consequently, there is a definite scope for the development of learning algorithms for neural networks, which consider the real degree of variability of the weight coefficients.

The convenience of the method for analyzing the functioning of neurons by cutting the edges of a hypercube is also that it allows one to carry out a sequential classification of neurons with different sets of weight coefficients starting from diagrams similar to Fig. 13.

Journal of Theoretical and Applied Information Technology
15th June 2021. Vol.99. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In particular, this method can be used to show that any non-degenerated neuron with an odd number of inputs can be immediately assigned to a sequence of weight coefficients of the form

$$\vec{w} = (a_1, a_2, \dots, a_{2m+1}); a_j = \pm 1 \quad (66)$$

For the proof, it is sufficient to refer to the diagram in Fig. 4: if there is a hypercube vertex such that the product of its radius vector by the normal to the cutting hyperplane reaches its maximum value, then this hyperplane can be replaced by a hyperplane defined by a vector of the form (66) without changing the sets of vertices into which the hypercube is cut.

This conclusion shows that there is a fairly extensive class of neural networks built on non-degenerated neurons with an odd number of inputs, the neurons in which de facto perform the operation of the scalar product of one binary sequence of the form (66) to another similar sequence. One of these sequences corresponds to the hypercube vertex (i.e., describes a set of logical variables entering the input of the neuron), and the other corresponds to the hyperplane cutting the hypercube (i.e., a set of weight coefficients). The symmetry between these sequences makes it much easier to train this type of neural network.

Let us now discuss the question of the adequacy of using the threshold function of neuron activation. We emphasize that in a significant part of applications (especially those related to classification problems), continuous activation functions are used only because they provide the possibility of implementing those learning algorithms for neural networks for which the differentiability of the considered function is critical.

All of these algorithms are ultimately based on the fact that they set certain rules for changing the vectors s $\vec{w}$ in the learning process. The proposed method for cutting edges allows you to implement a whole set of algorithms in which the change in these vectors in the learning process is displayed through the discrete movement of the image points in the ternary code space.

The correspondence between the functioning of neural networks built on elements with threshold and continuous activation functions can also be established in terms of ternary logic.

Let us turn to Fig. 15. It shows a "ternary" activation function, which changes abruptly at two points, as well as a continuous activation function, the length of the transition region of which corresponds to the length of the region of zero values for the "ternary" activation function.

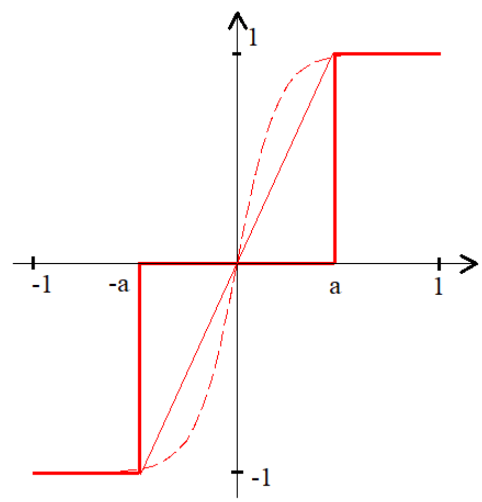



*Figure 15: Comparison of the neuron activation function corresponding to ternary logic, with a piecewise linear activation function*

Obviously, the result obtained using these two functions will be different. However, the “ternary” activation function allows using simple methods to establish those ranges of values of input variables at which the value of the output variable will differ from $\pm 1$ . For many applications of neural networks, for example, solving classification problems, such values correspond to an incompletely trained neural network: any classification problem can be described in terms of logical functions.

Consequently, the most significant conclusion that can be drawn from the comparison illustrated in Fig. 15 is that the use of continuous activation functions with a transition region imposes de facto additional restrictions on the variability of the sequences of weighting coefficients, at least for neural networks that solve certain types of problems.

Indeed, from the admissible sequences of the weight coefficients of each individual neuron, all those sequences should be excluded that do not provide an exact division of the set of all vertices of the hypercube into two subsets. Simplifying, in this case, the separation is carried out not by an infinitely thin plane, but by a region enclosed between two parallel hyperplanes.

## 5. CONCLUSION

Thus, a consistent geometric interpretation of the functioning of neurons included in an artificial neural network, which provides a mapping of a set of logical variables ("input image") to another set of logical variables, leads to the conclusion that the variability of the sequences of weight coefficients that define the functioning of an individual neuron, impose very serious restrictions. In particular, for neurons with a relatively small number of inputs, such sequences can be specified explicitly, which is done in this paper.

This conclusion is a significant step forward in terms of overcoming the thesis about the logical opacity of neural networks, which de facto has become widespread only due to the fact that the training of neural networks used in practice is the result of a computer experiment. Indeed, the reduction of operations performed by a neural network to logical ones, de facto, makes it possible to reveal a specific algorithm according to which a specific neural network functions. This conclusion seems to be important, first of all from the point of view of methodological problems of artificial intelligence - until the thesis about the logical opacity of neural networks is finally overcome, there will be more than serious difficulties in order to reveal the essence of intelligence at a level sufficient for operationalization. of this concept.

This conclusion in the long term creates the prerequisites for the development of learning algorithms for neural networks that provide a significant reduction in the number of computations, as well as the disclosure of the mechanism according to which the trained neural network makes a decision. Indeed, if there is a trained neural network, then from the sets of weight coefficients obtained empirically, using the proposed technique, one can go to the sets of weight coefficients corresponding to cutting certain edges of the hypercube, i.e. to well-defined logical operations that each of the neurons performs.

The most obvious practical application of the provided conclusions is associated with the replication of systems built on the use of trained neural networks. In this case, the analysis of sets of weight coefficients, will reduce the functioning of neurons to certain logical operations that, makes it possible to go over to explicitly prescribed algorithms built on the basis of logical operations, and here quite certain prospects for the use of ternary logic open up.

**REFRENCES:**

[1] Manish Mishra, Monika Srivastava, "A view of Artificial Neural Network", 2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), p. 1-3.

[2] David J. Livingstone, "Artificial Neural Networks", Humana Press 2009

[3] Jürgen Schmidhuber, "Deep learning in neural networks: An overview", Neural Networks, Volume 61, January 2015, Pages 85-117

[4] Ajay Shrestha, Ausif Mahmood, "Review of Deep Learning Algorithms and Architectures", IEEE Access (Volume: 7), p. 53040 – 53065, 22 April 2019.

[5] Sun-Chong Wang, "Artificial Neural Network. In: Interdisciplinary Computing in Java Programming", The Springer International Series in Engineering and Computer Science, vol 743. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-0377-4_5

[6] Bawa V.S., Kumar V., "Linearized sigmoidal activation: A novel activation function with tractable non-linear characteristics to boost representation capability", Expert systems with applications – 2019. – T. 120. – C. 346-356.

[7] Graupe D., "Principles of artificial neural networks", World Scientific, 2013. – T. 7.

[8] Jan Chorowski, Jacek M. Zurada, "Learning Understandable Neural Networks with Nonnegative Weight Constraints", IEEE Transactions on Neural Networks and Learning Systems (Volume: 26, Issue: 1, Jan. 2015)

[9] Johannet A., Vayssade B., Bertin D., "Neural networks: from black box towards transparent box. Application to evapotranspiration modeling", International Journal of Computational Intelligence. – 2008. – T. 4. – №. 3. – C. 163-170.

[10] Jackson P. C., "Introduction to artificial intelligence", Courier Dover Publications, 2019.

[11] Lasseri R. D. et al., "Taming nuclear complexity with a committee of multilayer neural networks", Physical review letters. – 2020. – T. 124. – №. 16. – C. 162502.

[12] Ibragim E. Suleimenov, Yelizaveta S. Vitulyova, Akhat S. Bakirov, and Oleg A. Gabrielyan, Artificial Intelligence: what is it? //

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Proceedings of the 2020 6th International Conference on Computer and Technology Applications (ICCTA '20). Association for Computing Machinery, New York, NY, USA, 22–25. https://doi.org/10.1145/3397125.3397141

[13] Rahwan, I., Simari, G. R., "Argumentation in artificial intelligence", Heidelberg Springer, 47, 2009.

[14] Ibragim Suleimenov, Aliya Massalimova, Akhat Bakirov, Oleg Gabrielyan. Neural Networks and the Philosophy of Dialectical Positivism. MATEC Web Conf. 214 02002 (2018) DOI: 10.1051/matecconf/201821402002

[15] Suleimenov E Ibragim, Gabrielyan A Oleg, Bakirov S Akhat and Yelizaveta S Vitulyova. Dialectical Understanding of Information in the Context of the Artificial Intelligence Problems // IOP Conference Series: Materials Science and Engineering. - №630, 2019 DOI: 10.1088/1757-899X/630/1/012007

[16] Suleimenov I. E., Sapanova E. S., Moldakhan I. Quantum nature of weighting factors of neural networks // bulletin of almaty university of power engineering and telecommunications. – 4(3) (43), 2018

[17] Yelizaveta S Vitulyova, Bakirov S Akhat, Dina B Shaltykova, and Suleimenov E Ibragim. Prerequisites for the analysis of the neural networks functioning in terms of projective geometry // IOP Conference Series: Materials Science and Engineering. - №946, 2020 DOI: 10.1088/1757-899X/946/1/012001

[18] Suleimenov I., Bakirov A., Moldakhan I. Formalization of Ternary Logic for Application to Digital Signal Processing //Energy Management of Municipal Transportation Facilities and Transport. – Springer, Cham, 2019. – C. 26-35. https://doi.org/10.1007/978-3-030-57453-6_3

[19] I Moldakhan, DB Shaltikova, ZM Egemberdyeva, IE Suleimenov. Application of ternary logic for digital signal processing // IOP Conference Series: Materials Science and Engineering. - №946, 2020 DOI: 10.1088/1757-899X/946/1/012002

[20] Akhat S Bakirov, Ibragim E Suleimenov. On the possibility of implementing artificial intelligence systems based on error-correcting code algorithms // Journal of Theoretical and Applied Information Technology. – 99(1), 2021 DOI: http://www.jatit.org/volumes/Vol99No1/8Vol99No1.