Journal of Theoretical and Applied Information Technology
15th January 2021. Vol.99. No 1

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

ON THE POSSIBILITY OF IMPLEMENTING ARTIFICIAL INTELLIGENCE SYSTEMS BASED ON ERROR-CORRECTING CODE ALGORITHMS

[1]AKHAT S BAKIROV, [2]IBRAGIM E SULEIMENOV

[1]Almaty University of Power Engineering and Telecommunications, Almaty, Republic of Kazakhstan

[2]National Academy of Engineering of the Republic of Kazakhstan, Almaty, Republic of Kazakhstan

E-mail: [1]axatmr@mail.ru, [2]esenych@yandex.ru

ABSTRACT

A new approach to the implementation of artificial intelligence systems is proposed, based on an analogy with the theory of error-correcting coding, as well as on the philosophical interpretation of intelligence as an information processing system that provides, first of all, its compression, for example, by reducing some complex digital image to a set of classification features. The approach is based on the expansion of the binary sequence into a fuzzy Fourier series, implying that the expansion approximates the original function up to a certain number of permissible deviations. This solves a problem similar to that which artificial neural networks solve, leading the recognizable image to the image from the original training set. The analogs of the images that make up the training sample are functions that form the basis for the expansion of the binary sequence into a fuzzy Fourier series and/or their combination.

Keywords: *Artificial Intelligence, Artificial Neural Networks, Error-correcting Codes, Dialectical Positivism*

1. INTRODUCTION

Currently, artificial intelligence (AI) systems are mainly implemented on the basis of artificial neural networks (ANNs) [1-3]. ANNs were originally created as a tool to reveal the mechanisms of brain functioning, and this task still remains more than relevant [4,5]. Moreover, the essence of intelligence as such remains unsolved, which gives rise to numerous discussions about which specific software products can be attributed to AI systems and which not. As noted in [6-8], such a discussion seems pointless, since there is no understanding of what intelligence is as such. We emphasize that all definitions of the concept of "intelligence" that are contained in the current literature (mainly humanitarian [9]) are descriptive in nature, i.e. they de facto represent a list of attributes of intelligence of one particular type - human. Obviously, this significantly narrows the field of research related to AI: intellect, strictly speaking, does not have to be implemented in the form that is inherent in humans. Therefore, attempts to solve the question of the essence of intelligence on the basis of the Turing test and similar approaches [10, 11] also seem methodologically incorrect. Moreover, papers [12] have already appeared in the current literature, which discuss the possibility of the occurrence of spontaneous intelligence in communication networks. We also emphasize that the discussion of the emergence / creation of an intellect other than human can in no way be considered in the plane of “crowding out a person with a machine”: it is necessary to take into account that consciousness and intellect are by no means identical concepts.

Therefore, the urgent question is the creation of an algorithmic basis for AI systems that will be more or less close to human, at least in terms of the nature of information processing and functionality [13,14]. This does not contradict the statement made above about the possibility of the existence of forms of intelligence that is different from human. On the contrary, it is the synthesis of AI built on a different algorithmic basis, including, that serves as the most important tool for understanding the essence of intelligence as such [15].

In this regard, it is appropriate to dwell on a certain methodological contradiction inherent in the ANN of existing varieties. Namely, typical ANNs are built on the basis of the training procedure, which boils down to selecting a set of weighting coefficients that provide a solution to a particular problem [1-3]. Let us conduct a thought experiment: imagine ANN in the form of a physically realizable object. Then the transmission channels of information of fundamentally different types should be brought to this object. One of them serves to adjust the weighting coefficients, and the other serves to supply the information that is processed by the ANN. Obviously, such a separation of the channels of information for the biological prototype is by no means characteristic. Here, the channels for supplying information that is used to train the neural network coincide with those channels through which the information processed by the biological neural network - the brain (vision, hearing, tactile sensations, etc.) is received.

Therefore, next-generation AI systems, which are close in their properties to the biological prototype by the next step, should be at least “methodologically symmetrical”, which clearly correlates with the principle of dialectic symmetry, put forward in [6-8].

In the present work, an attempt is made to overcome the “methodological asymmetry” characteristic of the ANN of existing varieties. The basis for this is the understanding of intelligence as a system of information processing [6-8]. From general methodological positions, the processing of information by the human brain and by artificial systems only remotely approaching it (ANN) is inseparable from any compression procedures (compression of information). So, a person can describe the observed object in a concise (in terms of information) form using the words of a natural language. In the literature [16], this is often interpreted as the allocation of “valuable” or “significant” information, but such a statement of the question does not seem justified, since the idea of the “value” of information requires the formulation of relevant criteria, which, however, was emphasized by the author [16] .

The procedure for pattern recognition by a neural network can also be regarded as a kind of information compression algorithm, since the pattern recognition procedure de facto maps an image specified, for example, in the form of a set of binary variables to a set of the same sequences containing fewer characters (this can be most clearly seen on the example of solving classification problems solved using neural networks [17-19]).

As shown in this paper, new types of AI can be built on the basis of considering intelligence as a system that provides information compression. This is what makes us pay the closest attention to the possibility of using the theory of error-correcting coding [20, 21] for the synthesis of ANN analogues.

Indeed, the procedure for decoding error-correcting codes [20] can also be considered from the point of view of information compression. A certain sequence consisting of a certain number of binary symbols (the code received at the output of the information transmission channel with noise) compares another sequence of binary symbols, but with a shorter length (error corrected code). The noise-resistant code is formed as a sequence of characters containing redundant information, which allows you to correct errors.

This comparison allows a slightly different look at the functioning of the ANN. Indeed, it is generally accepted that a neural network “recognizes an image”. This means that it reduces the observed / obtained image to a certain “standard”, and it is assumed that the “reference” images are set initially; it is on them that the neural network is configured in the learning process.

However, if we talk about solving classification problems, especially in relation to the concept of data mining, then such a statement of the question does not seem entirely justified. Very often it is not known what exactly “reference” images are. On the contrary, among some large number of different images, it is necessary to identify definite classification features and relate to them what is observed in the experiment. Obviously, the selection of the set of some classification features and the correlation of the observed / recorded object with them corresponds to the problem of information compression, which was mentioned above as key for creating AI systems. A huge number of really existing objects of a certain type (for example, horses) correlates the biological prototype of the ANN with the word of the natural language “horse”, which contains a very small amount of information.

We also emphasize that when using typical neural networks it is considered [1-3] that recognized images can deviate from a certain standard by a certain number of errors, that is, by a certain number of binary sequence characters that change the value to the opposite.

It is this, we emphasize once again, that the neural network pattern recognition procedure is related to the error-correcting code decryption procedure. However, if we do not know the "true" code, then the question can be posed differently.

There is a sequence of binary characters. We strive to reflect this information in the form of an approximation to a certain "reference" sequence, but presumably containing some errors. In other words, it is about providing the most "economical" (in terms of the amount of information transmitted) representation of the image in question. This idea can be explained as follows.

Suppose there is some image. We consider it possible to transmit this image with twenty-five percent errors, assuming that the consumer, if "important" information is saved in one sense or another, will recognize the desired image even when 25% of the elements were transferred incorrectly. Actually, the human brain works like this: an image can be recognized as a "horse" even when it is purely sketchy. Psychological experiments are known in which the child accurately recognized the cat in a variety of schematic images.

Therefore, for the implementation of AI of new varieties (for the implementation of ANN analogues that do not require the use of the weighting factor adjustment procedure), the same procedure can be used that is used in noise-resistant coding, but in a slightly different way. Specifically, we are talking about the following. There is a certain code containing N characters, it is asked if we can present this code as a combination of sequences with a significantly smaller number of characters so that the number of errors does not exceed a given value.

This will solve the problem of information compression, which provides the possibility of synthesizing ANN analogs built on a different algorithmic basis. In this paper, we propose ways to solve this problem, as well as the simplest schemes of ANN analogues that implement the proposed approach.

2. CLASSIFICATION PROBLEMS FROM THE POINT OF VIEW OF THE ERROR-CORRECTING CODING THEORY

Consider a set of N_1 sequences containing N binary characters. Obviously, such a set can obviously be represented as a (0,1) matrix $N_1 \times N$ or as an $N_1 \times N$ diagram (picture) containing dark and bright pixels.

For definiteness, for now, we will consider the problem of the answers of N_1 respondents to questions of a sociological questionnaire or to questions of a psychological test containing N questions and providing only two choices of the answer "Yes" and "No".

The question is whether, by analyzing only this sample, it is possible to isolate classification features that are applicable for analyzing other samples of respondents' answers to the same set of questions. If each of the classification features can also be described in terms of a binary variable, then from a mathematical point of view this question can be reformulated as follows.

There are many sets of Q answers, which obviously contain 2^N elements. It is assumed that there are many classification features Q_0 containing 2^M elements, where M is the number of such features, $M < N$.

It is required to find a mapping of the set Q onto the set Q_0, which can be interpreted as a classification rule that allows establishing a correspondence between a given respondent and a specific set of classification features.

Let us start from the analogy with error-correcting coding [18,19]. Noise-resistant coding assumes that each code sequence A_0 containing M binary symbols (such sequences form the set Q_0) is uniquely associated with a sequence A containing N symbols (such sequences form the set Q): $A = F(A_0)$, and the condition $M < N$.

A code is considered strictly error-tolerant (perfect) if each particular sequence from Q can be associated with a certain sequence from Q_0, so that the following condition is satisfied. If the sequence $B \in Q$ is mapped to the sequence $A_0 \in Q_0$, then $R(A, B) \leq r$, where $R(A, B)$ is the code distance between the sequences A and B, r is an integer (the number of correctable errors) .

Otherwise, errors can be corrected if the set Q is a union of disjoint subsets q_j, each of which is formed by one of the sequences $A = F(A_0)$ and all sequences, the number of "errors" (deviations of binary characters) in which does not exceed r.

Two varieties of strictly noise-tolerant binary codes (extreme or perfect codes) are known - Hamming codes correcting one error, and the Golay code [18], correcting three errors in a sequence containing 23 characters. Moreover, one of the most remarkable results of coding theory is the hypothesis

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

proved in the early 70s of the XX century, according to which there are no other perfect codes correcting t errors for t > 1 [18].

There are also error-correcting codes that allow you to correct a certain number of errors (if their number does not exceed a certain number) and identify the presence of a larger number of errors. An example is the Hamming code (8.4), using sequences of eight binary characters. It allows you to correct an error if it is only one or to identify the presence of two errors in a sequence.

The results of the theory of error-correcting coding can be applied to solve classification problems as follows.

Let us return to the consideration of arbitrary sequences containing N binary characters, which are formed on the basis of various experimental data (for example, during sociological surveys or psychological testing).

It is advisable to consider that such data may also contain certain errors caused by various factors (for example, the respondent's conscious or unconscious desire to distort information, etc.). More generally, the identification of a classification feature (a system of classification features) means that the sequences obtained in the experiments should be divided into some classes, each of which is assigned a sequence containing a smaller number of characters (a sequence of binary variables corresponding to classification features). It is natural to assume that sequences belonging to the same class should be close in terms of the code distance between them, which corresponds to the possibility of compensating for experimental errors of one nature or another.

Therefore, the question of distinguishing a system of classification features in this formulation of the problem reduces to choosing a noise-resistant code that determines the desired mapping of the set of sequences corresponding to different sets of classification features to the set of possible answers.

As a natural criterion for choosing such a mapping, it is permissible to choose the minimum number of errors that occurs when examining the initial set of experimentally obtained sequences (initial sample).

As noted above, this formulation of the problem also meets the tasks that are now being solved with the help of neural networks, since the recognition of a digital image, understood as bringing it to a certain standard, also corresponds to the "correction" of a certain number of errors.

3. THE CONCEPT OF AN ALGORITHM FOR CONSTRUCTING A CLASSIFICATION PROCEDURE FOR EXPERIMENTALLY OBTAINED (0,1) SEQUENCES

For now, for definiteness, we will consider the Hamming code (7.4), writing it as a sequence of characters x_i, $i = 1,2,\dots,7$.

This code can be set, for example, by solving a system of equations

$$\begin{cases} x_1 + x_3 + x_5 + x_7 = 0 \\ x_2 + x_3 + x_6 + x_7 = 0 \\ x_4 + x_5 + x_6 + x_7 = 0 \end{cases} \tag{1}$$

where the addition is carried out modulo 2.

Obviously, the three conditions (1) imposed on vectors from a 7-dimensional space give a set of solutions that is isomorphic to the space of four dimensions. This leads to commonly used formulas expressing the "test" characters of the Hamming code through the "informatic characters"

$$\begin{cases} x_1 = x_4 + x_5 + x_7 \\ x_2 = x_4 + x_6 + x_7 \\ x_3 = x_5 + x_6 + x_7 \end{cases} \tag{2}$$

where information symbols x_4, x_5, x_6, x_7 can be selected arbitrarily.

Record (2) can be reduced to the following form

$$\vec{a} = a_1\vec{e}_1 + a_2\vec{e}_2 + a_3\vec{e}_3 + a_4\vec{e}_4 \tag{3}$$

where the addition of the components of the vectors is carried out modulo 2, and the coefficients a_i also take a value from the Galois field (0,1).

Indeed, it is possible to form four vectors by setting in the sequence (x_4, x_5, x_6, x_7) one of the variables appearing in it equal to one, and the rest to zero.

These vectors have the form

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

$$\begin{cases} \vec{e}_1 = (1,1,0,1,0,0,0) \\ \vec{e}_2 = (1,0,1,0,1,0,0) \\ \vec{e}_3 = (0,1,1,0,0,1,0) \\ \vec{e}_4 = (1,1,1,0,0,0,1) \end{cases} \quad (4)$$

Thus, the error-correcting code can be constructed by multiplying the generating matrix 7×4 by a 4-vector of information symbols [18]:

$$\vec{a} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} \quad (5)$$

Representation (5) is obviously equivalent to representation (3).

Record (3) indicates that, up to one error, any sequence containing 7 binary symbols can be represented through a linear combination of four vectors defined over field (0,1).

The basis $\{\vec{e}_i\}$ that provide "error correction" can be chosen in various ways.

As applied to the classification problem under consideration, this means that, starting from a sample of the initial sequences, one should choose a basis $\{\vec{e}_i\}$ that would minimize the number of errors.

Summarizing, a sum of the form (3) can be interpreted as a fuzzy Fourier series, suggesting that an arbitrary binary sequence (an arbitrary vector $\vec{b}$) can be represented as an expansion in some basic vectors up to a certain number of "errors".

Thus, we can propose the following algorithm for solving the problems of classifying experimentally obtained binary sequences, starting from the theory of noise-tolerant binary codes.

There is a basic sample $\{\vec{b}_i\}$, $i = 1, \dots, N_1$; it is required to find a basis $\{\vec{e}_j\}$, $j = 1, \dots, M$, such that when expanding all the sequences (vectors $\vec{b}_i$) from the original sample into a fuzzy Fourier series

$$\vec{b}_i = \sum_{j=1}^{M} a_{ij} \vec{e}_j \quad (6)$$

Hamming total distance

$$S = \sum_{i=1}^{i=N_1} R(\vec{b}_i, \vec{a}) \quad (7)$$

will reach a minimum value.

4. THE RELATIONSHIP OF HAMMING CODES WITH THE WALSH BASIS

From the point of view used, it is important to establish a correspondence between the traditionally used representations for Hamming codes and the basis on which the functions defined over the Galois field (0,1) can be decomposed, i.e. accepting only binary values.

Let us start from the formula (5). It can also be represented as

$$\vec{a} = (a_1 \quad a_2 \quad a_3 \quad a_4) \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad (8)$$

moreover, when it comes to classification problems, the quantities a_i need not be interpreted as information symbols.

This is significant, since in this case it is possible, in particular, to pass from the basis appearing in (3) to any other bases formed by linearly independent linear combinations of the vectors $\vec{e}_j$. Note that the Hamming codes under consideration are linear, i.e. all possible code combinations are obtained by adding a certain number of lines in the matrix (8) modulo 2, which gives 16 combinations that provide, for example, the transmission of a sequence of 4 binary characters; these combinations form a group with respect to the addition operation modulo 2. In addition, the columns of the matrix on the right side of formula (8) can be reordered in any way. In particular, this can be done as follows

$$\vec{a} = (a_1 \quad a_2 \quad a_3 \quad a_4) \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad (9)$$

This form of matrix is often used in error-correcting coding manuals.

Columns can also be arranged as follows

$$\vec{a} = (a_1 \quad a_2 \quad a_3 \quad a_4) \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad (10)$$

From matrix (9), we can go to the matrix

$$\vec{a} = (a_1 \quad a_2 \quad a_3 \quad a_4)\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \quad (11)$$

which corresponds to the Hamming code (8.4).

Matrix (9) is obtained from matrix (11) by deleting the fifth column.

Matrix (11) can also be rearranged by rearranging the columns, bringing it to a form similar to (10).

$$\vec{a} = (a_1 \quad a_2 \quad a_3 \quad a_4)\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \quad (12)$$

Matrices (11) and (13) have the following properties. The arithmetic sum of the elements in each row is exactly 4, and the arithmetic sum of the elements of each column gives an odd number, which implies that

$$\sum_{j=1}^{M} \vec{e}_j = \vec{1} \quad (13)$$

where $\vec{1}$ denotes a vector whose all elements are equal to unity.

Composing linear combinations of vectors, it is obvious from the basis $\{\vec{e}_j\}$ that we can pass to the basis $\{\vec{g}_i\}$ if we use the nondegenerate matrix $\hat{A}$

$$\vec{g}_i = A_{1i}\vec{e}_1 + A_{2i}\vec{e}_2 + A_{3i}\vec{e}_3 + A_{4i}\vec{e}_4 \quad (14)$$

The notation (14) emphasizes that the matrix A˄ has the dimension 4×4. Using (14), in particular, we can pass to the basis to which the matrix (12) corresponds, in which the last row is replaced by the sum of all four rows, that is, by the vector $\vec{1}$.

This transformation corresponds to a non-degenerate matrix, the determinant of which is equal to unity, including when performing modulo 2 calculations.

$$\begin{pmatrix} \vec{g}_1 \\ \vec{g}_2 \\ \vec{g}_3 \\ \vec{g}_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} \vec{g}_1 \\ \vec{g}_2 \\ \vec{g}_3 \\ \vec{g}_4 \end{pmatrix} \quad (15)$$

Consequently, an arbitrary vector from the considered group of addition modulo 2 can be expressed, including, and so

$$\vec{a} = (a_1 \quad a_2 \quad a_3 \quad a_4)\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (16)$$

We use mapping

$$1 \to -1; 0 \to 1 \quad (17)$$

Such a mapping corresponds to the transition from (0,1) codes to (1, -1) codes in which the addition operation modulo 2 is replaced by the operation of direct multiplication of vector elements

$$\vec{c} = \vec{a} \cdot \vec{b} = (a_1b_1, a_2b_2, a_3b_3, \dots, a_nb_n) \quad (18)$$

With this mapping, the matrix on the right-hand side of (16) goes over to

$$\hat{S} = \begin{pmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{pmatrix}, \quad (19)$$

If the values in the rows of this matrix are considered as functions of the column number, then we can see that these functions can be reduced to Rademacher functions.

$$r_n(x) = sgn\left(sin(2^{n+1}\pi x)\right) \quad (20)$$

where sgn(x) – piecewise constant function of a real argument:

$$sgn(x) = \begin{cases} 1, x > 0 \\ 0, x = 0 \\ -1, \ x < 0 \end{cases},$$

$r_n(x)$ – n-th Rademacher function. The domain of definition of Rademacher functions is the range [0; 1]. Each next Rademacher function can be defined as the double compression along the Ox axis of the previous:

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

$$r_n(x) = r_{n-1}(2x)$$

Here are the first five Rademacher functions:

$$r_0(x) = sgn\big(sin(2\pi x)\big),$$
$$r_1(x) = sgn\big(sin(4\pi x)\big),$$
$$r_2(x) = sgn\big(sin(8\pi x)\big),$$
$$r_3(x) = sgn\big(sin(16\pi x)\big),$$
$$r_4(x) = sgn\big(sin(32\pi x)\big).$$

For clarity, the corresponding graphs are shown in Fig. 1. In the (1, -1) representation, modulo 2 addition, by which, in accordance with formula (3), all vectors of the group in question are formed, the code distance between two mismatched elements in which is 4 or 8, is replaced by multiplication.

It is known that the Walsh basis is formed by the multiplication of Rademacher functions, i.e. it can be argued that each of the elements of this group can be associated with one of the functions included in the Walsh basis. We will try to use this analogy to construct the expansion into the fuzzy Fourier series, which was mentioned above.

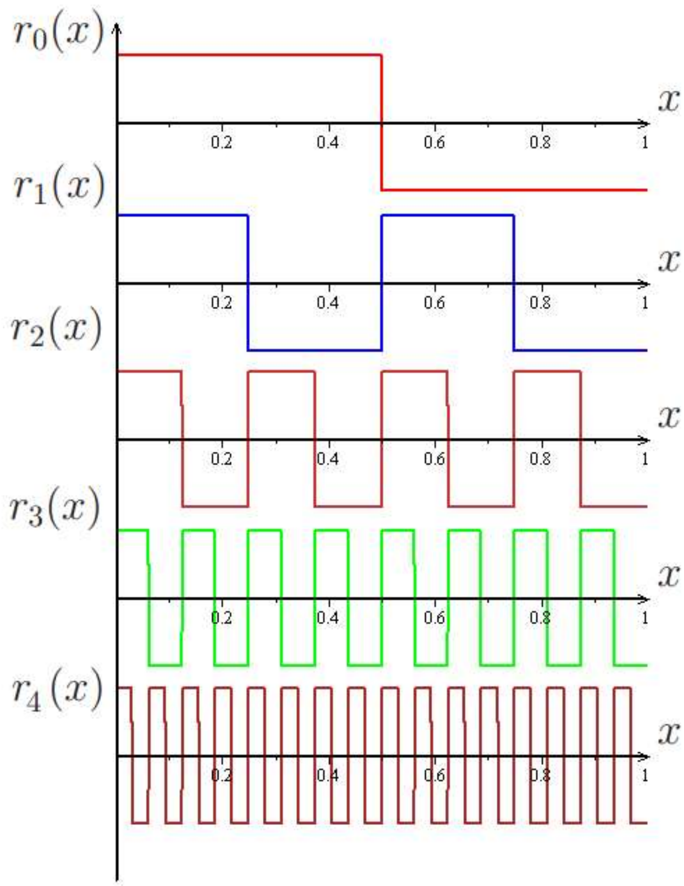

Figure 1: The First Five Rademacher Functions

5. FUZZY DECOMPOSITION OF THE CODE SEQUENCE INTO BASIS FUNCTIONS

A Walsh basis can be constructed using Hadamard matrices generating one of the orderings of this basis [22].

Hadamard matrices of length 2^n are constructed by the recursive method as

$$\widehat{H}_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{21}$$

$$\widehat{H}_{2^{m+1}} = \begin{pmatrix} \widehat{H}_{2^m} & \widehat{H}_{2^m} \\ \widehat{H}_{2^m} & -\widehat{H}_{2^m} \end{pmatrix}, \tag{22}$$

Each row in the Hadamard matrix corresponds to one of the functions in the Walsh basis of the corresponding length. The functions included in the Walsh basis are orthogonal.

Let us consider a similar construction for the case when the elements of the matrices take a value in the Galois field (0,1).

Lets determine

$$\hat{H}_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \tag{23}$$

and further

$$\hat{H}_{2^{m+1}} = \begin{pmatrix} \hat{H}_{2^m} & \hat{H}_{2^m} \\ 0 & \hat{H}_{2^m} \end{pmatrix}, \tag{24}$$

In particular, takes place

$$\hat{H}_{2^2} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{25}$$

$$\hat{H}_{2^3} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{26}$$

Using matrices of the form (26), any code combination can be represented through the binary spectrum. For a special case of an 8-character sequence, the corresponding entry is of the form

$$\vec{a} = (w_1 \quad w_2 \quad \dots \quad w_8) \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{27}$$

Record (27) obviously means that an arbitrary code sequence can be represented as a linear combination of row vectors. This representation is unambiguous since the matrices $\hat{H}_{2^m}$ are idempotent

$$\hat{H}_{2^m} \hat{H}_{2^m} = \hat{I}, \tag{28}$$

where $\hat{I}$ – unit matrix.

This, firstly, means that each matrix $\hat{H}_{2^m}$ has an inverse, and secondly, that the set of coefficients $\{w_j\}$ can be obtained by multiplying the row vector corresponding to the original code sequence by the matrix $\hat{H}_{2^m}$ on the right:

$$\vec{w} = \vec{a}\hat{H}_{2^m}. \tag{29}$$

In general, expression (27) is written through the operation of multiplying a row vector by a matrix $\hat{H}_{2^m}$ also on the right.

$$\vec{a} = \vec{w}\hat{H}_{2^m}, \tag{30}$$

Substitution of (30) into (29) gives, by virtue of (28), the identity, which allows us to use expansion (29). We emphasize that in order to expand a certain function in a row into basis functions, the latter do not have to be orthogonal, like the basis on which a vector is expanded in three-dimensional space is not necessarily orthogonal.

Further, in matrix (26) there are 8 rows; their set can be divided into two subsets, which correspond to the following sets of vectors. Choosing the first, second, third and fifth rows, we have

$$\begin{cases} \vec{e}_1 = (1,1,1,1,1,1,1,1) \\ \vec{e}_2 = (0,0,0,0,1,1,1,1) \\ \vec{e}_3 = (0,0,1,1,0,0,1,1) \\ \vec{e}_4 = (0,1,0,1,0,1,0,1) \end{cases} \tag{31}$$

and other lines

$$\begin{cases} \vec{q}_1 = (0,0,0,1,0,0,0,1) \\ \vec{q}_2 = (0,0,0,0,0,1,0,1) \\ \vec{q}_3 = (0,0,0,0,0,0,1,1) \\ \vec{q}_4 = (0,0,0,0,0,0,0,1) \end{cases} \tag{32}$$

Obviously, the set (32) coincides with the vectors that form the previously considered group of 16 elements, the code distances between the mismatched elements of which are 4 or 8.

The problem to be solved, recall, is formulated as follows. There is a code sequence corresponding to the vector $\vec{a}$. It is required to represent it (in the sense of addition modulo 2) through a linear combination of vectors $\vec{e}_i$ so that the deviation of the sum (in the sense of code distance) does not exceed 2. This corresponds to the number of permissible errors also equal to two.

At first glance, such a problem can be solved by simply discarding in row expansion (27) those row vectors that are on the list (32). This, however, does not solve the problem, since, for example, the sum of the vectors from list (32) gives a deviation of three errors:

$$\vec{q}_0 = \sum_{i=1}^{4} \vec{q}_i = (0,0,0,1,0,1,1,0). \quad (33)$$

We establish cases when the "remainder", understood as a linear combination of vectors (32)

$$\vec{q} = \sum_{i=1}^{4} v_i \vec{q}_i, \quad (34)$$

gives a deviation greater than 2.

The coefficients v_i take values (0,1), therefore there are 16 such combinations. If only one of these coefficients differs from zero, then the deviation does not exceed 2, since each of the vectors adds no more than two deviations to the total sum. If only two of the coefficients v_i are different from zero, then the deviation also does not exceed 2 since the sum of the units in the last position gives 0.

There are five possible cases, one of which has already been considered above (33), and the other four correspond to the situation when three of the coefficients v_i are nonzero.

By virtue of the identity that occurs when modulo 2 is added

$$\vec{a} + \vec{a} = 0, \quad (35)$$

for any $\vec{a}$, these four amounts can be represented as

$$\vec{q}_{ei} = \vec{q}_0 + \vec{q}_i \quad (36)$$

Or

$$\begin{cases} \vec{q}_{e1} = (0,0,0,0,0,1,1,1) \\ \vec{q}_{e2} = (0,0,0,1,0,0,1,1) \\ \vec{q}_{e3} = (0,0,0,1,0,1,0,1) \\ \vec{q}_{e4} = (0,0,0,1,0,1,1,1) \end{cases} \quad (37)$$

We will form the amount

$$\begin{cases} \vec{q}_{e1} + \vec{e}_2 = (0,0,0,0,1,0,0,0) \\ \vec{q}_{e2} + \vec{e}_3 = (0,0,1,0,0,0,0,0) \\ \vec{q}_{e3} + \vec{e}_4 = (0,1,0,0,0,0,0,0) \\ \vec{q}_{e4} + \vec{e}_2 = (0,0,0,1,1,0,0,0) \end{cases} \quad (38)$$

It can be seen that the deviation, i.e. the number of units in the right-hand sides of equalities (38) does not exceed 2. This means that when finding the sum

$$\vec{a} = w_1\vec{e}_1 + w_2\vec{e}_2 + w_3\vec{e}_3 + w_4\vec{e}_4 \quad (39)$$

approximately representing the vector $\vec{a}$ through linear combinations of vectors $\vec{e}_i$, we can use the following algorithm.

By the formula (29), the binary spectrum of the sequence is calculated. The components corresponding to the vectors (32) are selected in it. If the number of nonzero components does not exceed 2, then the values w_i are selected as those obtained by directly calculating the spectrum using formula (32). If the number of units is 3, then one of these components changes, more precisely one that minimizes the deviation in accordance with expressions (37). In particular, if the components corresponding to the vectors $\vec{q}_2, \vec{q}_3, \vec{q}_4$ are nonzero, then the factor for the component $\vec{e}_2$ changes to the opposite value.

If the number of units is 4, then all the quantities w_i should be replaced by the opposite. This follows from the fact that the vector

$$\vec{e}_0 = \sum_{i=1}^{4} \vec{e}_i = (1,0,0,1,0,1,1,0) \quad (40)$$

only in one position differs from vector (33). Really,

$$\vec{q}_0 + \vec{e}_0 = (1,0,0,0,0,0,0,0) \quad (41)$$

Thus, starting from an analogy with the methodology for constructing a Walsh basis, one can indeed indicate an explicit algorithm for finding the coefficients of the expansion of sequences of 8 characters into a fuzzy Fourier series.

This algorithm can be generalized, which is discussed below.

6. CLASSIFICATION ALGORITHM (0,1) OF SEQUENCES: REDUCTION TO PERIODICITY

Using isomorphism (17), the expansion of an arbitrary sequence containing 8 binary symbols into a fuzzy Fourier series can be represented as a direct product of vectors

$$\vec{a} = \vec{e}_1^{\,p_1} \vec{e}_2^{\,p_2} \vec{e}_3^{\,p_3} \vec{e}_4^{\,p_4} \quad (42)$$

where p_i - binary characters, and vectors $\vec{e}_i$ correspond to the first four Rademacher functions (Fig. 1). Looking ahead a bit, we note that the fifth Rademacher function shown in this figure will be

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

needed when considering a code containing 16 binary characters.

$$\begin{cases} \vec{e}_1 = (-1,-1,-1,-1,-1,-1,-1,-1) \\ \vec{e}_2 = (+1,+1,+1,+1,-1,-1,-1,-1) \\ \vec{e}_3 = (+1,+1,-1,-1,+1,+1,-1,-1) \\ \vec{e}_4 = (+1,-1,+1,-1,+1,-1,+1,-1) \end{cases} \quad (43)$$

It follows from relations (42) and (43) that each vector from the considered group can be represented in terms of a pair of vectors, each of which contains half as many symbols

$$\vec{a} = (\vec{a}^0, \vec{a}^1), \quad (44)$$

and besides

$$\vec{a}^1 = \pm\vec{a}^0 \quad (45)$$

The sign in formula (45) can be considered by introducing the variable s_1, which also corresponds to isomorphism (17).

$$\vec{a}^1 = s_1\vec{a}^0 \quad (46)$$

Further, this procedure, based on the fact that the form of vectors (43) corresponds to the Rademacher functions, can be continued by writing

$$\vec{a}^0 = (\vec{a}^{00}, \vec{a}^{01}) \quad (47)$$

moreover, an expression similar to (45) holds

$$\vec{a}^{01} = \pm\vec{a}^{00}, \quad (48)$$

which allows you to enter another binary character s_2, characterizing the expansion used

$$\vec{a}^{01} = s_2\vec{a}^{00} \quad (49)$$

At the final step, we can write

$$\vec{a}^{00} = (\vec{a}^{000}, \vec{a}^{001}) \quad (50)$$

and besides

$$\vec{a}^{001} = s_3\vec{a}^{000}, \quad (51)$$

but the vector $\vec{a}^{000}$ degenerates into a single binary symbol

$$\vec{a}^{000} = (a_1); a_1 = s_4 \quad (52)$$

Thus, an arbitrary vector from the group used to decompose an arbitrary binary sequence containing 8 characters into a fuzzy Fourier series can be represented as

$$\vec{a} = (a_1, s_3a_1, s_2a_1, s_2s_3a_1, s_1a_1, s_1s_3a_1, s_1s_2a_1, s_1s_2s_3a_1) \quad (53)$$

Obviously, the four characters a_1, s_1, s_2, s_3 are also acceptable like informational.

The representation corresponding to isomorphism (17) creates certain conveniences. Namely, for the case when the sequence in question is included in the group defined by expression (42), the quantities s_1, s_2, s_3 can be expressed through scalar products of vectors directly obtained from the vector $\vec{a}$. Indeed, by virtue of structure (53),

$$\vec{a}^0 \cdot \vec{a}^1 = a_1a_4 + a_2a_5 + a_3a_6 + a_4a_8 = 4s_1 \quad (54)$$

$$a_1a_3 + a_2a_4 = a_5a_7 + a_6a_8 = 2s_2 \quad (55)$$

$$a_1a_2 + a_5a_6 = 2s_3 \quad (56)$$

The calculation of such combinations is completely correlated with the calculation of the binary spectrum using matrix (26); the difference is that they allow us to identify the number of errors.

This is most clearly seen when calculating a combination of the form

$$W = a_1a_8 + a_2a_7 + a_3a_6 + a_5a_4 = 4s_1s_2s_3 \quad (57)$$

Its advantage over the above is as follows. If four characters a_1, s_1, s_2, s_3 are considered as informational, then they can be set arbitrarily. It follows from relations (43) that in this case a sequence containing 8 characters can be obtained by periodicity or by antiperiodicity, i.e.,

$$(a_8, a_7, a_6, a_4) = a_1a_2a_3a_5(a_1, a_2, a_3, a_5) \quad (58)$$

where it is taken into account that the product $s_1s_2s_3$ de facto determines the parity of the sequence, which is considered as the source (a sequence containing information symbols)

$$s_1s_2s_3 = a_1a_2a_3a_5 \quad (59)$$

Thus, we have proved that when constructing the error-correcting Hamming code (8.4), we can use the following procedure.

The initial symbols are located at positions 1,2,3,5, and the values of the remaining symbols are determined in accordance with (58), i.e. either periodically or aperiodically.

Accordingly, we can propose the following algorithm for correcting one error and identifying two errors in the Hamming code (8.4).

The scalar product (57) is calculated. The following options are possible: $W = \pm 4, \pm 2, 0$. If the case $W = \pm 4$ is realized, then the sequence does not contain errors, i.e. it belongs to a group whose elements are described by formula (42). If $W = \pm 2$, then the sequence contains one error, if $W = 0$ – two. That is, the calculation of the scalar product, in contrast to the standard procedure for using the Hamming code, allows you to immediately determine the number of errors, as well as the parity of the sequence in the case when the error is correctable. Indeed, if the value of W is nonzero, then the sign of this quantity uniquely determines the value of the product (59), and therefore this quantity $s_1 s_2 s_3$ itself, since all the factors take the values +1 or - 1.

Further, if there is only one error in the sequence (53), then one of the sequences (a_8, a_7, a_6, a_4) and (a_1, a_2, a_3, a_5) must be true. More precisely, the sequence whose parity corresponds to the parity of the scalar product (57) is true. Thus, to restore the original sequence, it is enough to perform only two operations: calculate the scalar product (57) and establish the parity of the sequence (a_1, a_2, a_3, a_5). Looking ahead a bit, we note that the result allows us to propose a circuit of an analog of a neural network that corrects one error in the Hamming code.

We now consider a generalization of the proposed approach to the case of sequences containing a larger number of characters.

It is convenient to start from the correspondence established above between the Hamming codes and the Rademacher functions and the Walsh basis. The next step in constructing matrices generating a Walsh basis for binary sequences, i.e. matrices of the form (26), leads to the matrix $\hat{H}_{2^4}$ containing 16 rows.

Applying the procedure proposed above to it, it is necessary to distinguish lines that differ from the line in which there are only units by a code distance of 8 and 16. These lines also correspond to the Rademacher functions (Fig. 1) and correspond to the following vectors written in the form corresponding to isomorphism (17).

$$\begin{cases} \vec{e}_1 = (-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1) \\ \vec{e}_2 = (+1,+1,+1,+1,+1,+1,+1,+1,-1,-1,-1,-1-1,-1,-1,-1) \\ \vec{e}_3 = (+1,+1,+1,+1,-1,-1,-1,-1,+1,+1,+1,+1-1,-1,-1,-1) \\ \vec{e}_4 = (+1,+1,-1,-1,+1,+1,-1,-1,+1,+1,-1,-1,+1,+1,-1,-1) \\ \vec{e}_5 = (+1,-1,+1,-1,+1,-1,+1,-1,+1,-1,+1,-1,+1,-1,+1,-1) \end{cases} \quad (60)$$

Obviously, the set of vectors (60) generates a group, each vector of which can be represented as

$$\vec{a} = \vec{e}_1^{\,p_1} \vec{e}_2^{\,p_2} \vec{e}_3^{\,p_3} \vec{e}_4^{\,p_4} \vec{e}_5^{\,p_5} \quad (61)$$

Here, as in the similar formula (42), a direct product of vectors is used, each element of which can take the value +1 or - 1.

We emphasize that the set of vectors from the group generated by vectors (60) covers the set of sequences containing 16 binary symbols in the following sense.

The total number of such sequences is $2^{16} = 65536$; the number of code combinations that differ from the specified code distance 3 is

$$1 + 16 + \tfrac{1}{2} 16 \cdot 15 + \tfrac{1}{6} 16 \cdot 15 \cdot 14 = 697 \quad (62)$$

The total number of code combinations that differ from the specified code distance 4 is

$$1 + 16 + \tfrac{1}{2} 16 \cdot 15 + \tfrac{1}{6} 16 \cdot 15 \cdot 14 + \tfrac{1}{6} 16 \cdot 15 \cdot 14 \cdot 13 = 2517 \quad (63)$$

The number of code combinations in the group generated by vectors (60) is $2^5 = 32$. Multiplying this number by the result (62), we get 22304, which is about 34% of the number of possible code combinations. Multiplying 32 by the result (63), we get 80544, which exceeds the number of possible code combinations containing 16 characters.

In other words, a sequence of binary characters can be expanded into a fuzzy Fourier series with a

Journal of Theoretical and Applied Information Technology
15th January 2021. Vol.99. No 1

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

tolerance of 4 errors. We emphasize that in this case the ratio of the number of permissible errors to the total number of characters in the code sequence remains the same as for the case of the 8-digit sequence considered above.

Bringing an arbitrary sequence to one of the elements of the group can be carried out in the same way as was used above, i.e. using the matrix $\widehat{H}_{2^4}$ containing 16 rows.

However, if a certain sequence $\vec{a}$ contains 4 errors, then such a recovery will be ambiguous. It becomes unambiguous when the sequence $\vec{a}$ contains 3 errors. In this case, you can use the same procedure that was applied above to the Hamming code (8.4), i.e. in fact, symmetry considerations related to the periodicity of all sequences that appear in (61).

Namely, any sequence $\vec{a}$, due to the periodicity of the functions corresponding to the sequences in (61), can be represented as

$$\vec{a} = (\vec{a}^0, \vec{a}^1), \tag{64}$$

where

$$\vec{a}^1 = s_1 \vec{a}^0 \tag{65}$$

Repeating the same arguments that led to formula (53), we find that the vectors (64) are representable in the form

$$\vec{a}^0 = \begin{pmatrix} a_1, s_4 a_1, s_3 a_1, s_3 s_4 a_1, s_2 a_1, s_2 s_4 a_1, \\ s_2 s_3 a_1, s_2 s_3 s_4 a_1 \end{pmatrix} \tag{66}$$

$$\vec{a}^1 = \begin{pmatrix} s_1 a_1, s_1 s_4 a_1, s_1 s_3 a_1, s_1 s_3 s_4 a_1, s_1 s_2 a_1, \\ s_1 s_2 s_4 a_1, s_1 s_2 s_3 a_1, s_1 s_2 s_3 s_4 a_1 \end{pmatrix} \tag{67}$$

Provided that the sequence in question contains 3 errors, they can be corrected in the same way as described above by making the following verification scalar products.

$$W = a_1 a_{16} + a_2 a_{15} + a_3 a_{14} + a_4 a_{13} + a_5 a_{12} + a_6 a_{11} + a_7 a_{10} + a_8 a_9 = 8 s_1 s_2 s_3 s_4 \tag{68}$$

$$W_1 = a_1 a_8 + a_2 a_7 + a_3 a_6 + a_5 a_4 = 4 s_2 s_3 s_4 \tag{69}$$

$$W_2 = a_1 a_8 + a_2 a_7 + a_3 a_6 + a_5 a_4 = 4 s_2 s_3 s_4 \tag{70}$$

This error correction algorithm, as can be shown, can be generalized to the case of sequences containing 2^{2m} characters with values $m > 2$.

Thus, instead of the traditionally used procedure for correcting errors in binary sequences based on the consideration of polynomials in binary variables, we can propose a procedure based on analogies with Fourier series decompositions. In this case, fuzzy Fourier series naturally arise, the main feature of which is the possibility of fuzzy expansion of the considered function in a series in an incomplete system of basis functions. For sequences containing 2^m characters, these functions are Rademacher functions that take logical values.

7. CONSTRUCTION OF ANALOGUES OF NEURAL NETWORKS WITH CLEARLY DEFINED FUNCTIONING ALGORITHMS

The possibility of “error correction” using the periodization algorithm allows us to propose the following scheme, which provides, for example, the solution of classification problems (Fig. 2). This example refers to 8-digit sequences, but as follows from the above, it can be generalized to the case of any sequences containing 2^{2m} binary characters.

The basis of this scheme are analogues of formal neurons (Ni), which have three logical inputs, and perform the following logical operation

$$\begin{cases} F(x,y,z) = x, z = 0 \\ F(x,y,z) = y, z = 1 \end{cases} \tag{71}$$

In this circuit, the input z is the control. If it is equal to zero, then a value is formed at the output that is equal to that which is implemented at input x, and if it is unity, then at input y. Note that such a scheme can well be implemented using typical electronic components, as well as molecular informatics methods.

The circuit of Fig. 2 includes four such elements, the inputs of which are supplied in pairs with variables corresponding to the sequences (a_1, a_2, a_3, a_5) and (a_8, a_7, a_6, a_4). As shown above, if the analyzed sequence contains only one error, then one of these sequences is true. Therefore, the classification problem (the allocation of four classification features) in this scheme is reduced to determining the value of the control variable z.

In accordance with the above, the sign of the quantity W, formula (57) is determined by the parity of the sequence (a_1, a_2, a_3, a_5), which is considered as the initial one. Accordingly, in the diagram of Fig. 2, the element Q1 is included, which calculates the scalar product (57) and determines its sign.

This element has two outputs, on one of which a symbol is formed that corresponds to the parity of the scalar product, and on the second - a symbol that corresponds to the ability to correct the error. This symbol takes on a zero value when the error cannot be fixed.

The diagram of Fig. 2 also includes the element Q2, which computes the product $a_1 a_2 a_3 a_5$. The signal from the output of the parity of the element Q1 and the signal from the output of the element Q2 is supplied to the element Q3, which performs the exclusive-OR operation. Thus, if the parities of the scalar product W and the product coincide, then the logical unit is formed at the output of the element Q3, and the elements Ni form the symbols corresponding to the sequence (a_1, a_2, a_3, a_5) at the inputs. If the above parities do not coincide, then a logical zero is formed at the output of Q3, and symbols corresponding to the sequence (a_8, a_7, a_6, a_4) are formed at the outputs of the analog of the neural network. If the scalar product takes the value +2 or –2, a logical unit is also formed at the output of the element Q3, i.e. the true sequence is the initial sequence (a_1, a_2, a_3, a_5).

The element Q0 in this circuit corresponds to the commonly used first layer of the neural network; it reflects the way in which the input signals are routed to the elements of Ni.

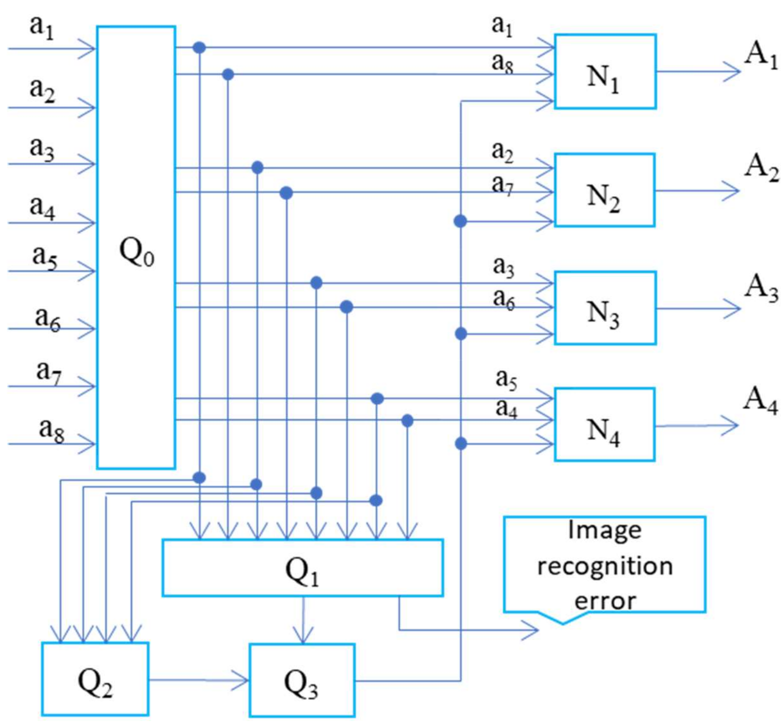

Figure 2: A functional Diagram Of An Analogue Of a Neural Network That Provides a Digital Image To a Basis Image Is The Correction Of One Error In The Hamming Code

Consider an illustrative example of solving the classification problem by the proposed method.

Table 1 presents the questions of the questionnaire distributed among undergraduates of the Almaty University of Power Engineering and Telecommunications of the first year of study. Respondents were asked to answer each question only “Yes” or “No”. A total of 28 people took part in the survey. The nature of the distribution of responses is illustrated in Table 2.

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

Table 1. Questionnaire Questions Used For Illustrative Example

Question Number	Question
1	Do you like beer?
2	Does a graduate student have the right to publish an article criticizing the pseudo-scientific work of teachers (if any)?
3	The Kazakh media are actively discussing the issue of legalization of tokal (second wife). Do you think polygamy really needs to be legalized?
4	Have you had to cheat on exams?
5	Do you perform religious ceremonies?
6	Are there any technical disciplines (subjects) that are part of the master's program at your university that are obviously useless (from your point of view)?
7	During your graduate studies, did you use the material support of your parents?
8	Do you think the results of your master's thesis can benefit someone?

Table 2. Distribution Of Answers Among Respondents And The Results Of Applying The Error Correction Procedure

Respon-dent number	Source sequence								*W*	Num-ber of errors	Corrected / Correct Sequence			
	a1	a2	a3	a4	a5	a6	a7	a8			A1	A2	A3	A4
1	1	-1	1	-1	1	-1	1	1	2	1	1	-1	1	-1
2	1	-1	1	-1	1	1	1	-1	2	1	1	-1	1	-1
3	1	-1	1	-1	1	-1	1	-1	4	0	1	-1	1	-1
4	1	-1	1	-1	1	-1	1	1	2	1	1	-1	1	-1
5	1	1	1	1	1	-1	1	1	2	1	1	1	1	1
6	1	-1	1	-1	1	1	1	-1	2	1	1	-1	1	-1
7	1	1	1	-1	1	-1	-1	1	-2	1	1	1	1	-1
8	1	-1	1	-1	1	-1	-1	-1	2	1	1	-1	1	-1
9	1	-1	1	-1	1	-1	1	1	2	1	1	-1	1	-1
10	1	-1	-1	1	-1	-1	-1	1	2	1	1	-1	-1	1
11	1	-1	1	-1	1	-1	1	-1	4	0	1	-1	1	-1
12	1	-1	1	-1	1	-1	1	-1	4	0	1	-1	1	-1
13	1	-1	1	-1	1	-1	1	-1	4	0	1	-1	1	-1
14	-1	1	1	1	1	-1	1	-1	-2	1	-1	1	1	1
15	-1	-1	1	-1	-1	-1	1	-1	4	0	-1	-1	1	-1
16	-1	-1	1	-1	1	-1	1	-1	2	1	1	-1	1	-1
…		…												
27	1	-1	1	-1	1	-1	-1	-1	2	1	1	-1	1	-1
28	1	-1	1	-1	1	-1	1	-1	4	0	1	-1	1	-1

Each set of answers (answers of one respondent) can be considered as (8.4) Hamming code and, accordingly, apply the error correction procedure to them. The corresponding results are also presented in Table 2. In this case, the presence of one error in the sequence makes it possible to classify this survey participant as one of the classification groups, and the presence of two errors is interpreted as the inability to classify according to this set of features.

In the language of the theory of neural networks, this can be formulated as follows. There are respondents whose "digital images" can be assigned to a certain class (one error or lack of error); There are also respondents who do not fit into this classification (the image is not recognized).

However, upon receipt of the result shown in Table 2, the classification criteria were de facto randomly selected. Indeed, the total number of errors substantially depends on the choice of the sequence of answers. So, for the sequence corresponding to Table 2, the total number of errors is 6, but if you use other sequences (Table 3), then the total number of errors will change, as the last column of this table shows.

Table 3. Dependence Of The Total Number Of Errors On Permutations In a Sequence Of Binary Symbols

Sequence								The total number of "errors" (among them double, uncorrectable)
a1	a2	a3	a4	a5	a6	a7	a8	26 (6)
a3	a4	a1	a7	a3	a2	a6	a8	36 (10)
a1	a2	a3	a5	a8	a7	a6	a4	24 (5)
a8	a1	a5	a3	a4	a6	a2	a7	36 (11)
a1	a2	a3	a5	a8	a7	a4	a6	22(4)
a7	a2	a8	a1	a6	a4	a5	a3	36 (11)

We emphasize that changing the order of answers in the considered sequence de facto means that instead of basis (42), another basis is used that has similar properties.

In other words, it is possible to minimize the number of errors by choosing a suitable basis. This procedure is analogous to the neural network training procedure, which is emphasized in Fig. 3, which shows a circuit similar to fig. 2, but showing that outputs from neurons of the first layer can be fed to inputs of Ni elements in other combinations. When constructing the circuit of Fig. 3, a specific sequence is used that corresponds to the minimum number of errors (this sequence is highlighted in Table 3 in color).

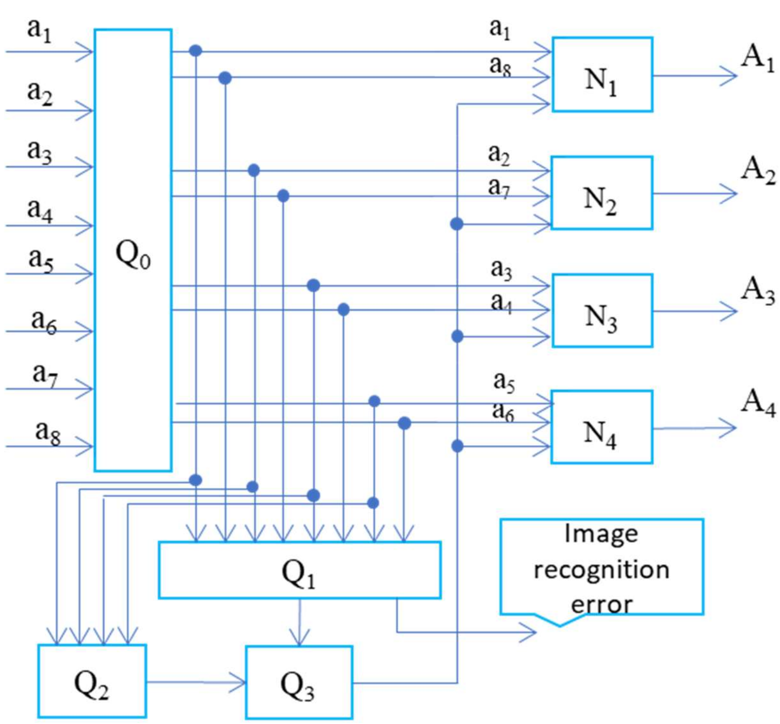

Figure 3: An Illustration To The Selection Of Basic Functions By Which Expansion Is Carried Out Is An Analogue Of The Neural Network Training Procedure

Thus, one can propose an analog of a neural network, the functioning of which is based on an explicitly prescribed algorithm, and an analog of the training procedure here is expressed in a change in the nature of the connections between analogs of neurons.

8. CONCLUSION

Thus, the interpretation of intelligence as a system that solves the classification problem and provides information compression allows us to propose a new approach to the implementation of AI, based on the use of analogies with the theory of error-correcting coding.

Such AI systems are based on the expansion of a binary sequence corresponding to a digital image in a fuzzy Fourier series, which approximates the original sequence up to a certain number of errors (deviations). As an example of a fuzzy Fourier series, we can consider the expansion of sequences of binary symbols in a series in Rademacher functions; this basis is obviously not complete, but it allows us to provide a representation of an arbitrary binary sequence with an accuracy of 25% of errors.

Such a representation de facto corresponds to the same problem that neural networks solve, leading the image arriving at the inputs of the ANN to a certain image from among those on the basis of which the ANN was trained. The images from the training set within the framework of the proposed approach correspond to the components of the fuzzy Fourier series or their combinations. The advantage of the proposed approach is the ability to explicitly specify the algorithm in accordance with which the AI system operates.

In addition, with this approach, the methodological contradiction characteristic of typical ANNs is removed due to the fact that their training procedure is a procedure for setting the weighting coefficients of connections between neurons, which is implemented independently. The proposed approach implies that the concept of weighting coefficients is not used at all; instead, an analogue of the training procedure is implemented through the restructuring of connections within the ANN analogue.

REFERENCES:

[1] Van Gerven M., Bohte S., Artificial neural networks as models of neural information processing, Frontiers in Computational Neuroscience, 11, 2017, p. 114. https://doi.org/10.3389/fncom.2017.00114

[2] Walczak S., Artificial neural networks, Advanced Methodologies and Technologies in Artificial Intelligence, Computer Simulation, and Human-Computer Interaction. – IGI Global, 2019, pp. 40-53.

[3] Kim D. H., MacKinnon T., Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks, Clinical radiology, 73(5), 2018, pp. 439-445. https://doi.org/10.1016/j.crad.2017.11.015

[4] Prieto A. et al., Neural networks: An overview of early research, current frameworks and new challenges, Neurocomputing, 214, 2016, pp. 242-268. https://doi.org/10.1016/j.neucom.2016.06.014

[5] Priddy K. L., Keller P. E., Artificial neural networks: an introduction, SPIE press, 68, 2005.

[6] Ibragim E. Suleimenov, Yelizaveta S. Vitulyova, Akhat S. Bakirov, and Oleg A. Gabrielyan., Artificial Intelligence: what is it? Proceedings of the 2020 6th International Conference on Computer and Technology Applications (ICCTA '20). Association for Computing Machinery, New York, NY, USA, 22–25, 2020 DOI: https://doi.org/10.1145/3397125.3397141

[7] 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

[8] Vitulyova Y. S., Bakirov A. S., Baipakbayeva S. T. and Suleimenov I. E., Interpretation of the category of "complex" in terms of dialectical positivism, IOP Conference Series: Materials Science and Engineering, 946, 2020 DOI: 10.1088/1757-899X/946/1/012004

[9] Pfeifer R., Scheier C., Understanding intelligence, MIT press, 2001.

[10] Marcus G., Rossi F., Veloso M., Beyond the turing test,Ai Magazine, 37(1), 2016, pp. 3-4. https://doi.org/10.1609/aimag.v37i1.2650

[11] Khan M., Shah T., Batool S. I., A new implementation of chaotic S-boxes in CAPTCHA, Signal, Image and Video Processing, 10(2), 2016, pp. 293-300. https://doi.org/10.1007/s11760-014-0741-5

[12] Chen, J., & Burgess, P. (2018). The boundaries of legal personhood: how spontaneous intelligence can problematise differences between humans, artificial intelligence, companies and animals, Artificial Intelligence and Law, 27(1), 2019, pp. 73-92.

[13] 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

[14] Vitulyova Y. S., Bakirov A. S., Shaltykova D. B. and Suleimenov I. E., 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

[15] Suleimenov I.E., Pak I.T., Bakirov A.S., Prolegomens to the fourth law of dialectics, Modern problems of informatics and computational technologies: Mat. scientific conf. (July 2 - 5, 2018), Almaty: IIVT MON RK, 2018, pp.266-276

[16] Chernavskii D S, The origin of life and thinking from the viewpoint of modern physics, Phys. Usp. 43, 2000, pp. 151–176 DOI: 10.1070/PU2000v043n02ABEH000609

[17] Gilda S., Source code classification using Neural Networks, 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), IEEE, 2017, pp. 1-6. DOI: 10.1109/JCSSE.2017.8025917

[18] Yu S., Jia S., Xu C., Convolutional neural networks for hyperspectral image classification, Neurocomputing, 219, 2017, pp. 88-98. https://doi.org/10.1016/j.neucom.2016.09.010

[19] Ibragim S. et al., Example of the Use of Artificial Neural Network in the Educational Process, Future of Information and Communication Conference, Springer, Cham, 2020, pp. 420-430. DOI: https://doi.org/10.1007/978-3-030-39445-5_31

[20] MacWilliams F. J., Sloane N. J. A., The theory of error correcting codes., Elsevier, 16, 1977.

[21] Tomlinson M. et al., Error-Correction Coding and Decoding: Bounds, Codes, Decoders, Analysis and Applications, Springer Nature, 2017.

[22] Hazra L., Mukherjee P., Walsh Functions, Self-similarity in Walsh Functions and in the Farfield Diffraction Patterns of Radial Walsh Filters, Springer, Singapore, 2018, pp. 1-15. https://doi.org/10.1007/978-981-10-2809-0_1