ASTES

Advances in Science, Technology and Engineering Systems Journal Vol. 6, No. 4, 91-99 (2021)

www.astesj.com

Special Issue on Multidisciplinary Sciences and Engineering

ASTESJ

ISSN: 2415-6698

New Neural Networks for the Affinity Functions of Binary Images with Binary and Bipolar Components Determining

Valerii Dmitrienko, Serhii Leonov*, Aleksandr Zakovorotniy

Modeling, System Control and Artificial Intelligence Team, Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, 61002, Ukraine

ARTICLE INFO

Article history:
Received: 01 December, 2020
Accepted: 22 June, 2021
Online: 10 July, 2021

Keywords:
Hamming neural network
Generalized architecture of the Hamming neural network
Maxnet network
Recognition and classification problems
Information technologies
Computer systems

ABSTRACT

The Hamming neural network is an effective tool for solving problems of recognition and classification of objects, the components of which are encoded using a binary bipolar alphabet, and as a measure of the objects' proximity the difference between the number of identical bipolar components which compared include objects and the Hamming distance between them are used. However, the Hamming neural network cannot be used to solve these problems if the input network object (image or vector) is at the same minimum distance from two or more reference objects, which are stored in the weights of the connections of the Hamming network neurons, and if the components of the compared vectors are encoded using a binary alphabet. It also cannot be used to assess the affinity (proximity) binary vectors using the functions of Jaccard, Sokal and Michener, Kulchitsky, etc. These source network Hamming disadvantages are overcome by improving the architecture and its operation algorithms. One of the disadvantages of discrete neural networks is that binary neural networks perceive the income data only when it's coded in binary or bipolar way. Thereby there is a specific apartness between computer systems based on the neural networks with different information coding. Therefore, developed neural network that is equally effective for any function of two kinds of coding information. This allows to eliminate the indicated disadvantage of the Hamming neural network and expand the scope of discrete neural networks application for solving problems of recognition and classification using proximity functions for discrete objects with binary coding of their components.

1. Introduction

Hamming distance R_X between arbitrary binary vectors $J_p = (j_{p1}, j_{p2}, \ldots, j_{pn})$, $J_q = (j_{q1}, j_{q2}, \ldots, j_{qn})$ is determined by the number of binary digits in which the compared vectors do not match [1]. This distance is often used to assess the proximity of discrete objects, which are described using binary alphabets with binary {0, 1} or bipolar {–1, 1} components. On the basis of bipolar neurons, a discrete Hamming neural network has been developed [1], which is successfully used to solve problems of recognition and the proximity measure of binary objects, the components of which are encoded using alphabet elements {–1, 1}, estimation. However, the evaluation of the measure of proximity of the compared objects (binary bipolar vectors) is carried out only by the most noticeable signs - different binary digits. At the same time, more "subtle" features for compared objects are ignored, which are often used to assess the similarity of binary vectors using the affinity (similarity) functions of Russell and Rao, Dice, Kulchitsky, Yule, etc.

In particular, in works [2–5] a brief description of these functions and their application is provided to compare different objects. In studies [2] they are used in the development of artificial immune systems, in works [3] and [4] they are used in the development of recognition systems for various objects. In this case, the characteristics of such objects are encoded by binary features. In work [5] given function is used to classify microorganisms.

In works [2–5], it is described and used not more than 15 commonly used functions. Although their total number is noticeably higher. So, in research in mathematical biology [6], 20 functions are used, and with hierarchical clustering, the set of random binary data (Hierarchical clustering Result of Random Binary Data Set) 76 different functions are indicated [7]. Once

*Corresponding Author: Serhii Leonov, Email: serleomail@gmail.com

again it indicates the relevance of the application of this approach for the classification and assessment of the proximity of various objects, which is described by a plurality of binary features.

2. Problem Statement

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

Analysis of recent publications on the theory and application of neural network Hamming [1 – 10] shows that the Hamming network has distinct disadvantages. In the architecture of the Hemming neural network, highly specialized layers of neurons are used: neuron layer calculating only the dot product of binary bipolar vectors; a neural network that allows you to select only one reference image from the network memory. This image is at the minimum distance from the input image in connection with this network

– It cannot work with the binary input data, when the measure of proximity of objects is estimated using finer characteristics than the Hamming distance. In particular, using the distances [2 – 5, 7];

– It cannot recognize objects, if they are at the same minimum distance from the two or more reference objects [1];

– cannot switch from processing information encoded using a bipolar alphabet to processing information presented using a binary alphabet and vice versa.

All this requires the development of a new Hemming neural network with a more advanced architecture.

When comparing objects with qualitative features encoded using a binary alphabet, for each pair of objects (binary vectors) $J_p = (j_{p1}, j_{p2}, \ldots, j_{pn})$, $J_q = (j_{q1}, j_{q2}, \ldots, j_{qn})$ in works [2 –5] four variable: a, b, f, g are used (table 1).

The variable a is used to count the number of bits with coinciding unit components of the vectors J_p and J_q. The variable b is needed to determine the number of binary digits of vectors J_p and J_q., in which both vectors have zero components, at the same time that means these objects do not have encoded with zeros in the same bits features.

The number of unit components that an object (vector) J_p has, but vector J_q. does not have, is determined using the variable f. The variable g, on the contrary, counts the number of unit components that the vector J_q. has, but are absent in the second vector J_p.

From the analysis of the Table. 1 it follows that an increase in the variable a unambiguously indicates an increase in the similarity (affinity) of the compared vectors. There is no such unambiguity with an increase in the variable b, since an increase in this variable can indicate both an increase in the similarity of the compared objects, and their belonging to different classes (if there is no similarity in the variable a). As for the measure of proximity of vectors J_p and J_q with respect to the variables f and g, it is symmetric with respect to these variables. An increase of these variables indicates the differences between the compared vectors increase.

Table 1: Variables for comparing binary vectors with binary components

	J_p	
J_q	1	0
1	$a = \sum_{k=1}^{n} j_{qk} j_{pk}$	$g = \sum_{k=1}^{n} (1 - j_{pk}) j_{qk}$
0	$f = \sum_{k=1}^{n} (1 - j_{qk}) j_{pk}$	$b = \sum_{k=1}^{n} (1 - j_{qk})(1 - j_{pk})$

Variables a, b, f and g from Table 1 are used in a number of well-known binary object similarity functions, in which the presence or absence of features is determined by the components of the binary alphabet. As an example, we can note the affinity (similarity) functions of Sokal and Michener, Jaccard and Needham, and so on. Using the variables f and g, one can easily determine the Hamming distance between binary vectors J_p and J_q

$$R_X(J_q, J_p) = f + g. \quad (1)$$

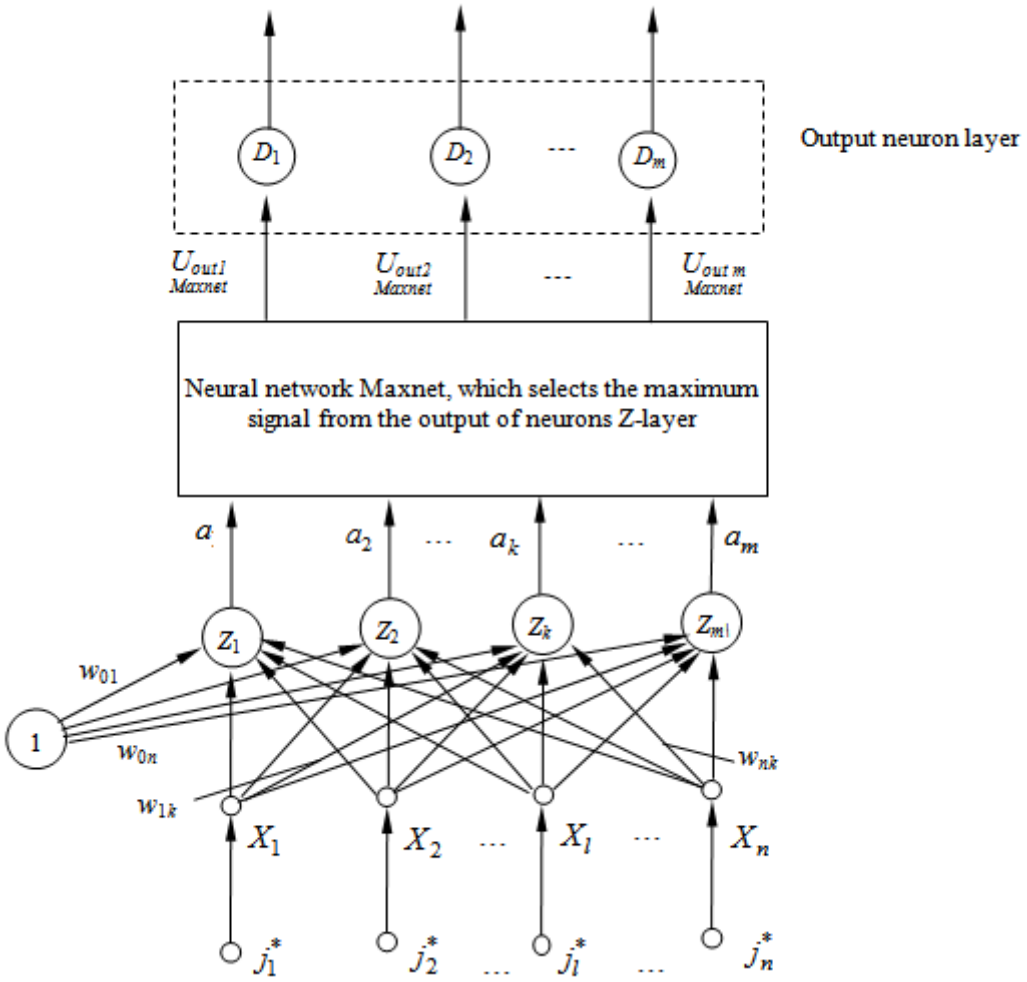
Variables a, b, f and g, as well as the affinity functions of Sokal, Michener, Jaccard, Needman, Hamming distance can be calculated using a new neural network based on binary neurons. The disadvantage of this neural network is the impossibility of processing information presented using the bipolar alphabet, as well as the inability to calculate the Hamming distance using the above affinity functions of Sokal and Michener, Jaccard, etc.

3. Generalized Block Diagram of a Hamming Neural Network

Since the theory, architecture and algorithms for the functioning of the Hamming network are described in detail in modern literature [1, 6 – 10], we will give only a generalized block diagram of the neural network (Figure 1) and a brief description of its functioning.

In the Hamming neural network, the input image $J^* = (j_1^*, j_2^*, \ldots, j_n^*)$ is fed to the inputs of X-neurons, which are multipliers of the bipolar image components J^*. Each Z-neuron stores one of the m reference images in the scales of its relations $J_k = (j_{k1}, j_{k2}, \ldots, j_{kn})$, $k = \overline{1, m}$ and compares this image with the input $J^* = (j_1^*, j_2^*, \ldots, j_n^*)$. As a measure of the similarity of the compared images (bipolar vectors), their scalar product is used [1, 8]

$$J_k J^* = \sum_{i=1}^{n} j_{ki} j_i^* = a_k - b_k, \quad (2)$$



where a_k and b_k are, respectively, the number of identical and different components of the compared vectors (images) J_k and J^*.

Since $a_k + b_k = n$, the scalar product (2) can be written in the form

$$J_k J^* = 2a_k - n\,. \tag{3}$$

From relations (2) and (3) it is easy to obtain

$$a_k = \frac{n}{2} + \frac{1}{2} J_k J^* = \frac{n}{2} + \frac{1}{2} \sum_{i=1}^{n} j_{ki} j_i^*\,. \tag{4}$$

Relation (4) can be regarded as a description of the input signal of a neuron Z_k (Figure 1), which has a displacement of $n/2$ and n inputs, that perceive the components of the input image $J^* = (j_1^*, j_2^*, \ldots, j_n^*)$ through neurons X_l ($l = \overline{1, n}$). In this case, the output signals of the X-layer elements repeat their input signals:

$$U_{outXl} = U_{inpXl} = j_l^*,\ l = \overline{1, n}, \tag{5}$$

where

$$U_{outXl} = \begin{cases} +1, & if \quad j_l^* = 1, \\ -1, & if \quad j_l^* = -1. \end{cases} \tag{6}$$

Each element of the X-layer is associated with the input of each neuron of the Z-layer. The scales of these relations contain information about m reference images $J_1, J_2, \ldots, J_m$, stored in the neural network. The scales of relations $w_{1k}, w_{2k}, \ldots, w_{nk}$ contain information about the k-th reference image J_k.

Z-layer elements compute their output signals

$$U_{outZk} = g_Z(U_{inp\ Z_k}) = a_k,\ k = \overline{1, m},$$

where g_Z – function of activation of neurons of the *Z*-layer; a_k – output element's Z_k signal, that arrives at the input of the neuron of the Maxnet network [1, 8]. From the output signals of the *Z*-layer elements, the Maxnet neural network, using an iterative process, extracts a single maximum signal greater than zero. Since the output signals of *D*-layer neurons are calculated by the ratio

$$U_{outDi} = \underset{\text{Maxnet}}{g_z(U_{out\,A_i})} = \begin{cases} 1, if \;\; U_{inpDi} > 0, \\ 0, if \;\; U_{inpDi} \le 0, \end{cases}, \; i = \overline{1, m}, \quad (7)$$

then at the output of the *Z*-layer of *D*-elements there will be only one single signal, which indicates which reference image is closest to the input vector J^*.

Analysis of the architecture of the discrete neural network shown in Figure 1 shows, that the layers of the network elements function relatively independently of each other. In particular, the layer of *Z*-neurons that determine the similarity functions of the input and reference images using the dot product can be replaced by blocks or neurons that calculate the similarity functions using other relations. In particular, the input bipolar signals and neurons of the *X*- and *Z*-layers can be replaced with binary.

The relative independence of the functioning of the main layers of the Hamming neural network from each other makes it possible to propose the use of binary neurons in the first layer of the network to work with binary input images (vectors). And also, to propose more "subtle" comparisons of input and stored in memory images using a variety of similarity (affinity) functions, modified using the Hamming distance (1) R_X.

The purpose of the article is to develop a generalized architecture of the Hemming neural network, which allows you to process input information that can be specified using both bipolar and binary alphabets. Also, based on the new architecture, the development of methods for solving problems of recognition and classification of binary objects is shown. In this case, the Hamming distance is used, as well as the functions of Jaccard, Sokal and Misher, Kulchinsky, etc.

4. Analytical Relationships Connecting Similarity Functions for Binary Objects and Hamming Distance

Since similarity functions for binary vectors (objects) and for objects described using the bipolar alphabet are used to assess the similarity of binary objects, then there should be dependencies connecting the similarity functions for binary objects (vectors) and the Hamming distance. Table 2 shows the aforementioned classical affinity functions (first column of Table 2) and similarity functions, modified using the Hamming distance (second column of Table 2).

Simple analytical relationships for determining the Hamming distance through the classic similarity functions Russel and Rao,

Table 2: Relationship between Hamming Distance and Similarity Functions for Binary Objects

Classic similarity functions	Similarity functions using Hamming distance	Hamming distance expressed in terms of binary similarity functions
Russell and Rao similarity function $S_1 = a/(a+b+f+g)$	$S_1 = \frac{a}{a+b+R_X}$	$R_X = \frac{a}{S_1} - (a+b)$
Sokal and Michener similarity function $S_2 = \frac{a+b}{a+b+f+g}$	$S_2 = \frac{a+b}{a+b+R_X}$	$R_X = \frac{a+b}{S_2} - (a+b)$
Jaccard and Needham similarity function $S_3 = a/(a+f+g)$	$S_3 = \frac{a}{a+R_X}$	$R_X = \frac{a}{S_3} - a$
Kulchitsky similarity function $S_4 = \frac{a}{f+g}$	$S_4 = \frac{a}{R_X}$	$R_X = \frac{a}{S_4}$
Dice similarity function $S_5 = a/(2a+f+g)$	$S_5 = \frac{a}{2a+R_X}$	$R_X = \frac{a}{S_5} - 2a$
Yule affinity function $S_6 = (ab-fg)/(ab+fg)$	Bulky expression	$R_X = f+g$
Correlation $S_7 = \frac{ab+fg}{[(a+f)(a+g)(b+g)(b+f)]^{1/2}}$	Bulky expression	$R_X = f+g$

Table 3: The Values of the Similarity Functions for the Compared Binary Vectors

	a	b	f	G	S_1	S_2	S_3	S_4	S_5	S_6	S_7	R_X
Values of similarity functions for vectors J_1 and J_k	5	5	2	0	5/12	10/12	5/7	5/2	5/12	1	5/√35	2
Values of similarity functions for equal vectors $J_1 = J_k$	7	5	0	0	7/12	1	1	∞	5/12	1	1	0
Numerical values of similarity functions for vectors J_1 and J_k with opposite components	0	0	7	5	0	0	0	0	0	– 1	1	12

Sokal and Michener, Jaccard and Needham, Kulchitsky, Dice were shown in the third column of the Table 2 for the first time ever. It was not possible to obtain simple analytical dependences connecting the Yule similarity function S_6 and the correlation coefficient S_7. But it is possible to determine the Hamming distance by functions S_6 and S_7 can using the relation (1).

Example. Let's perform the comparison using functions a, b, f and g, $S_1 - S_7$ some vector's pairs:

1) $J_1 = (111001110001)$, $J_k = (101001100001)$;

$J_1 = J_k = (111001110001)$;

$J_1 = (111001110001)$; $J_k = (000110001110)$.

The calculation results are shown in Table 3.

Analysis of the calculation results shown in Table 3 shows that the similarity functions $S_1 - S_5$ depend on the compared vectors. The values of these functions for vectors with opposite components take the minimum values, and for vectors with the same components, on the contrary, the maximum values.

If the compared vectors J_1 and J_k do not all coincide and not all are opposite in their components, then the similarity functions $S_1 - S_5$ take their values between the data of the second and third rows of Table 3, that is, between its maximum and minimum values. The function S6 practically obeys this rule, and the calculated values of which, given in the first and second lines, coincide.

As follows from the data Table 3 and the third column of the Table 2, the Hamming distance in comparison with the similarity functions $S_1 - S_5$ behaves in a certain sense opposite: using the functions $S_1 - S_6$ the similarity of the compared objects (vectors) is calculated, and using the Hamming distance, on the contrary, the difference of objects (vectors), therefore the Hamming distance R_X takes the maximum value for vectors that do not have any of the same components and the minimum distance (equal to zero) for vectors in which all components are the same.

The affinity functions are not limited to the ratios given in Table 2 [2 – 5]. A significant number of these similarity functions indicate the absence of one universal function suitable for comparing any binary objects with binary information encoding. The choice of one similarity function for solving a specific problem is determined, as a rule, by repeated modeling on the initial data in the space of features encoded using a binary alphabet.

5. New Neural Networks that Recognize Binary Images with Binary Components

The generalized block diagram of the Hamming neural network in Figure 1 clearly shows that the three main layers of a neural network can change and function almost independently of each other. In particular, instead of the Maxnet network, another neural network can be used that selects one or several identical maximum signals (if any). This allows you to find reference images that can be at the same minimum distance from the input. It also follows from the network architecture (Figure 1) that the layer calculating the measure of proximity of the input and reference images can calculate various measures of proximity. The relative independence of the main layers of the Hamming neural network makes it possible to propose neural networks for working with binary images (vectors) and using various affinity functions given in Table. 2, between the input and reference binary images. The Hamming distance can also be used (1). The classical architecture of the Hamming neural network [1, 6] for calculating the closeness measure of the input and reference vectors involves the use of the scalar product of two bipolar vectors [1, 8], one of which is a vector of connection scales, and the other is an input vector. In this case, the sum of the products of bipolar components does not contain any zero terms. If the scalar product of two binary vectors J_q and J_p is calculated in the classical Hamming neural network, then it is calculated as follows

$$J_q J_p = \sum_{i=1}^{n} j_{qi} j_{pi} = (a+b) - (f+g),$$

where variables (a, b, f, g) are defined by relations from Table 1.

Thus, if only one neuron is required to calculate the scalar product of two bipolar vectors using a classical neural network, then to calculate the scalar product of two binary vectors, you must first calculate the variables a, b, f and g using the relations given in Table 1.

In Figure 2 a block for calculating the variable b from Table 1 via neuron Z is shown:

$$b = \sum_{k=1}^{n} (1 - j_{qk})(1 - j_{pk}), \quad (8)$$

where $(1-j_{qk})$, $(k=\overline{1,n})$ – scales of neuron relations; $(1-j_{pk})$, $(k=\overline{1,n})$ – neuron Z input vector components; n – number of neuron relations scales Z.

With the help of neurons ∑1, ∑2, …, ∑n the components $(1-j_{p1}),, (1-j_{p2}),, \ldots, (1-j_{pn})$ of the input vector for neuron Z are computed. These components are accordingly communicated with the scales $(1-j_{q1})$, $(1-j_{q2})$, …, $(1-j_{qn})$, which allows us to calculate the variable b.

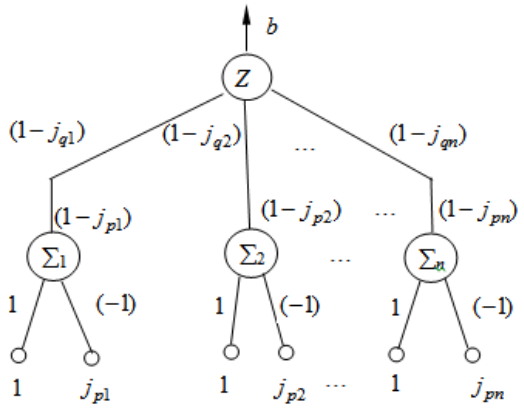


Figure 2: Neuron to calculate a variable b

If in Figure 2 to the inputs of neuron Z to apply input signals j_{p1}, j_{p2}, …, j_{pn} instead of signals calculated by summing elements $\sum_1$, $\sum_2$, …, $\sum_n$ of signals $(1-j_{p1})$, $(1-j_{p2})$, …, $(1-j_{pn})$, then neuron Z will calculate the variable f. To calculate the variable g (see Table 1) in Figure 2 it is necessary to replace the scale coefficients $(1-j_{p1})$, $(1-j_{p2})$, …, $(1-j_{pn})$ with the corresponding scale coefficients j_{p1}, j_{p2}, …, j_{pn}.

To obtain from the neural network shown in Figure 2, the neural network calculating the variable a needs to leave only one neuron Z with relation scales j_{q1}, j_{q2}, …, j_{qn}, to the inputs of which these j_{q1}, j_{q2}, …, j_{qn} signals, respectively have to be sent.

Having neurons or neural components for calculating the variables a, b, f and g, using table. 2 easy to get neural network architectures to compute any affinity function like Dice function S_5 (Figure 3).

The neural network for calculating the affinity function S_5 of the input vector and the reference vector is shown in Figure 3. The network has three layers of neurons. On the first layer of the network, the functions a, f and g are calculated. On the second layer, based on the calculated functions a, f and g, the sum $2a + f + g$ is calculated, which is used by the division unit to calculate the similarity function S_5.

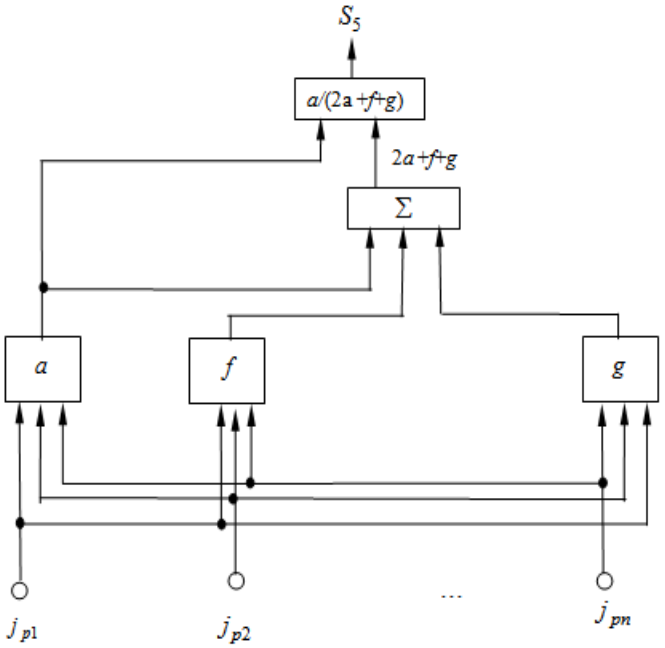


Figure 3: A neural network calculating the Dice function S_5

Similarly, neural network architecture can be obtained for calculating any other affinity functions. Figure 4 shows the architecture of a neural network that calculates the correlation coefficient S_7, the formula for calculating which is given in the very bottom line of the table. 2. With the help of this coefficient the similarity of the input vector and the reference vector, stored in the memory of the neural network, is estimated.

The neural network (Figure 4) for calculating the correlation coefficient has six layers of neurons. On the first layer, using blocks of neurons 1 – 4, functions a, b, f and g are calculated, which on the second layer are used to calculate intermediate variables ab, $(a + f)$ $(a + g)$, $(b + f)$, $(b + g)$, fg using blocks 5 – 10. On the third network layer (fig. 4) for calculating intermediate variables $(a + f)$ $(a + g)$ and $(b + f)$ $(b + g)$ the results of calculations, respectively, of blocks 6, 7, 8 and 9 are used. On the fourth network layer, the intermediate variables of the third network layer are used to calculate the radical expression in the denominator of the correlation S_7 coefficient formula (Table 2). The fifth layer of the neural network is required to determine the numerator and denominator of the correlation coefficient S_7, which is calculated on the last layer of the neural network.

5.1. Binary Neural Networks with the Transformation of Input InformationRrepresented Using the Bipolar Alphabet.

The architecture of neural networks for determining the Dice S_5 function and the correlation coefficient S_7, as well as other similarity functions, given in Table 2, in the general case can be represented in the following form (Figure 5).

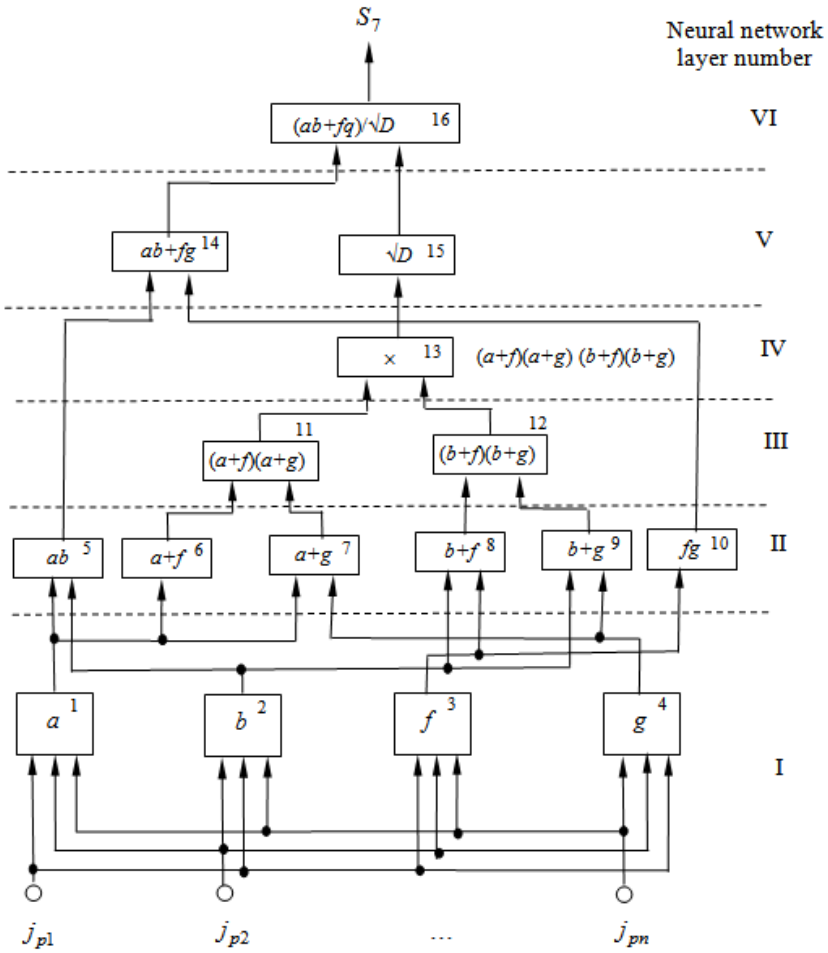


Figure 4: Neural network for calculating the correlation coefficient S_7 (the formula for calculating the coefficient S_7 is given in the bottom line of the Table. 2)

The neural network shown in Figure 5 has four layers of neurons. Both binary and bipolar signal vectors can be applied to the network input (*B*-neuron layer). If binary signals $j_{p1}^{(0,1)}$, $j_{p2}^{(0,1)}$, …, are fed to the input of the neural network, then all bipolar components of the vector ($j_{p1}^{(-1,1)}$, $j_{p2}^{(-1,1)}$, …, $j_{pn}^{(-1,1)}$) are equal to zero. If nonzero bipolar components are fed to the network input, then all components of the input vector ($j_{p1}^{(0,1)}$, $j_{p2}^{(0,1)}$, …, $j_{pn}^{(0,1)}$) are equal to zero. When specifying input information using a binary alphabet, the first three layers of neurons are used in the function. In this case, the input vector of binary signals ($j_{p1}^{(0,1)}$, $j_{p2}^{(0,1)}$, …, $j_{pn}^{(0,1)}$) arrives at the first inputs of neurons B_1, B_2, …, B_n. Second inputs of *B*-neuron layer are connected to the outputs of the neuron layer, which converts the input bipolar signals $j_{pk}^{(-1,1)}$, $k=\overline{1,n}$ to binary $j_{hk}^{(0,1)}$, $k=\overline{1,n}$.

Since the input information to the network defining the similarity function comes in binary form, the input vector ($j_{p1}^{(-1,1)}$, $j_{p2}^{(-1,1)}$, …, $j_{pn}^{(-1,1)}$) has to have all zero components. In this regard, zero signals $j_{h1}^{(0,1)} = j_{h2}^{(0,1)} = \ldots = j_{hn}^{(0,1)} = 0$ also appear at the output of layer 4 of the neural network. Therefore, the second inputs of neurons B_1, B_2, …, B_n will receive zero signals from the outputs of the elements of layer 4. *B*-layer neurons have activation functions of the form

$$U_{outBi} = \varphi(U_{inpBi}) = U_{inpBi} = j_{jhi}^{(0,1)},\ i=\overline{1,n}, \qquad (9)$$

where U_{outBi}, U_{inpBi} – respectively, the output and input signals of the neurons of the *B*-layer. In this regard, the output signals of the neurons of layer *B* repeat the input signals of the neural network $j_{p1}^{(0,1)}$, $j_{p2}^{(0,1)}$, …, $j_{pn}^{(0,1)}$ and are used to calculate the variables *a, b, f* and *g* on the second layer of the neural network, which are necessary to determine a specific similarity function S_d.

If the input information at the inputs of the neural network shown in Fig. 5, comes in the form of a bipolar vector $j_{p1}^{(-1,1)}$, $j_{p2}^{(-1,1)}$, …, $j_{pn}^{(-1,1)}$, then the neurons of layer 4 convert bipolar signals into binary according to the ratio

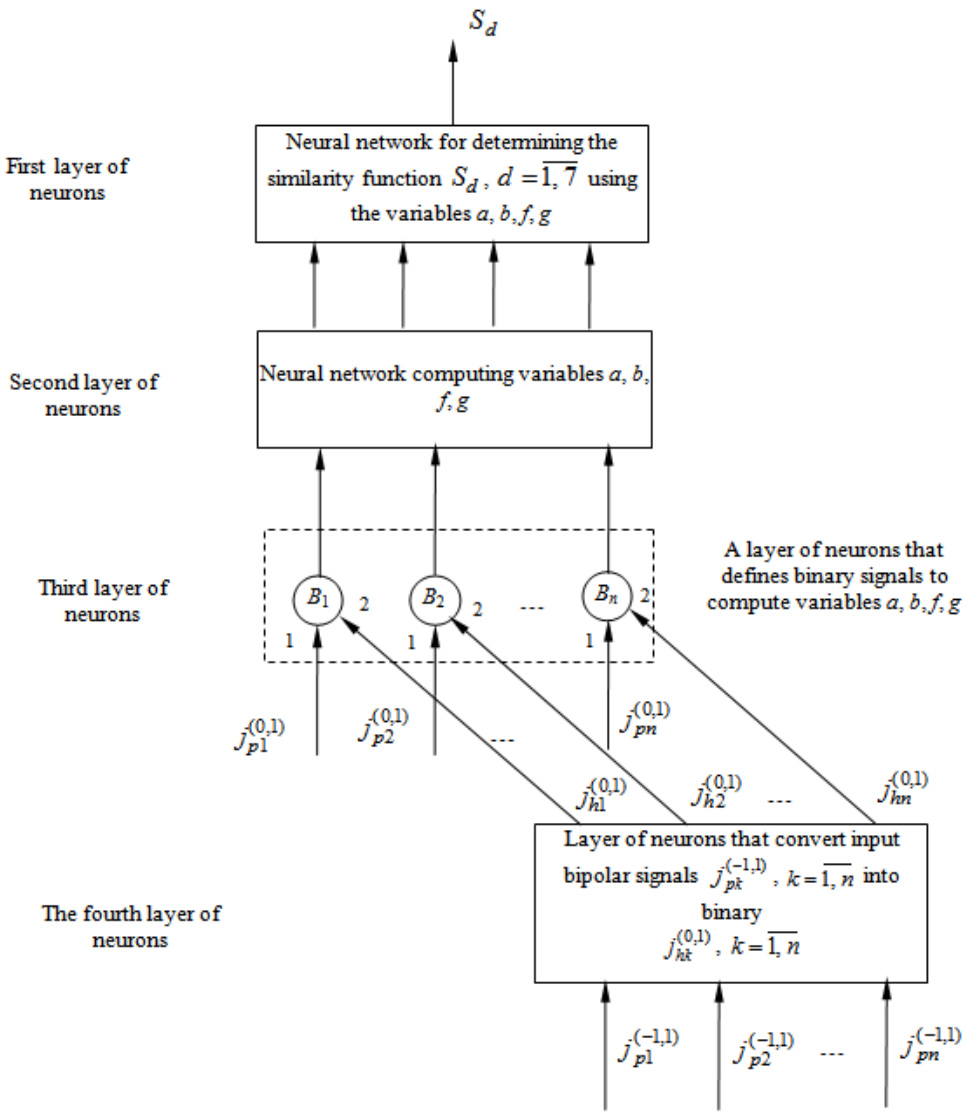


Figure 5: Architecture of neural networks for determining similarity functions S_1 – S_7 when the input information using binary or bipolar alphabets is specified

$$j_{hi}^{(0,1)} = \begin{cases} 1, \ if \ \ j_{pi}^{(-1,1)} = 1, \\ 0, \ if \ \ j_{pi}^{(-1,1)} = -1, \end{cases} \quad i = \overline{1,n}. \qquad (10)$$

The signals j_{h1}, j_{h2}, …, j_{hn} from the outputs of the neurons of layer 4 arrive at the second inputs of neurons $B_1, B_2, \ldots, B_n$, the first inputs of which receive zero signals $j_{p1}^{(0,1)}$, $j_{p2}^{(0,1)}$, …, $j_{pn}^{(0,1)}$, since the information necessary to determine the similarity function, in this case, is set in bipolar vector ($j_{p1}^{(-1,1)}$, $j_{p2}^{(-1,1)}$, …, $j_{pn}^{(-1,1)}$).

Thus, the architecture of the neural network shown in Figure 5, provides the computation of similarity functions for both binary and bipolar input signals. Obviously, a block of neurons can be synthesized in a similar way, which transforms input information from a binary alphabet into a bipolar one.

Similar blocks for transforming input information from one alphabet to another can be used at the inputs of any neural networks.

6. Conclusions

Binary and bipolar alphabets are used in various information technologies and methods for solving problems of recognition, classification, and estimation of the proximity of various binary objects. However, there is certain isolation in the application of methods based on different alphabets. The estimation of the measure of proximity of the compared bipolar vectors (objects, images) is mainly carried out using the dot product of vectors or Hamming distance. On their basis, the Hamming neural network has been developed and widely used. However, the Hamming neural network cannot be used when the recognition problem can have several solutions, to solve problems in cases where the components of binary vectors or images are encoded using a binary

alphabet. It cannot be used to estimate the affinity of binary vectors using the functions of Jaccard, Sokal and Michener, Kulchitsky, etc. In this regard, based on the generalization of the architecture of the Hamming network, new neural networks have been developed using binary coding of input information and the above similarity functions for binary objects. This expands the area of application of neural networks for solving problems of recognition and classification of input information using proximity functions using more "subtle" signs of proximity of discrete objects with binary coding. At the same time, when describing the general architecture of the generalized Hemming neural network, it is noted that the Maxnet network can be replaced by a neural network that allows you to define several solutions (if they exist). This allows, if necessary, to synthesize neural networks that allow obtaining a predetermined number of solutions in advance.

The idea of developing blocks for converting input information from one binary alphabet to another made it possible to propose not only a generalized Hamming network working with input information encoded using both binary and bipolar alphabets, but also modifications of any other binary neural networks that have worked so far with input information encoded with just one binary alphabet.

Thus, the scientific novelty of the results obtained is as follows.

For the first time, on the basis of the Hamming neural network, is proposed the concept of constructing new neural networks for solving problems of classification and recognition of binary objects using different distances and proximity functions. In doing so, the following principles are used:

1) adaptation of the first layer of neurons that determine the similarity functions of the input and reference images depending on the applied distances. This has allowed to develop a unified approach to the synthesis of neural networks that use different similarity function;

2) replacement Maxnet neural network in a neural network that can allocate one or more identical maximum signal. This allows you to select one or more reference images that are at the same minimum distance from the input image.

Also for the first time obtained the architecture and algorithms of functioning the neural network that can process the input information provided by both the bipolar and in binary form.

In real conditions some uncertainties while specific features comparison can be arisen with the binary objects' juxtaposition. For example, while the fruits or plants juxtaposition the color can be changed in different time intervals. In this case there could be uncertainties, that will be described with the third truth state.

Wherein, the third truth value could be interpretated in different ways [11 – 14]:

– like the intermediate value between truth and false;

– like the lack of information;

– like a specific paradoxical or even meaningless value, that gives such a meaningless value too;

– like values of three-valued paraconsistent logics, in which the third truth value can be interpreted in some statements as both true and false at the same time;

– like uncertainty of the three-valued logics, that describes quantum mechanics interpretations, etc.

The theory of making solutions' development in conditions of binary objects is a promising direction in our opinion.

Conflict of Interest

There is no conflict of interest reported between the authors.

References

[1] V.D. Dmitrienko et al., Neural networks: architecture, algorithms and usage. Tutorial, Kharkiv, NTU "Khpi", 222, 2020.

[2] S.A. Babichev, Theoretical and practical principles of information technology for processing gene expression profiles for gene network reconstruction. Dissertation for the degree of Doctor of Technical Sciences in specialty 05.13.06 – Information Technology, Kherson, Kherson National Technical University, 382, 2018.

[3] V.D. Dmitrienko et al., Methods and algorithms of artificial intelligence systems, Kiev, Kafedra, 282, 2014.

[4] A. For, Perception and pattern recognition, Moscow, Engineering, 272, 1989.

[5] R.S. Michalski A recent advance in data analysis: clustering jbjects into classes characterized by conjunctive concepts, Invited chapter in the book Progress in Pattern Recognition. **1**, North-Holland Publishing Company, Amsterdam-NewYork-Oxford, 33-49, 1981.

[6] Z. Chay et al., "Russel and Rao Coefficient is a Suitable Substitute for Dice Coefficient in Studying Restriction Mapped Genetic Distances of Escherichia colis". – iConcept Pocket Journal Series: Computational and Mathematical Biology, January 2010.

[7] S. Choi et al., "A Survey of Binary Similarity and Distance Measures", Systemics, Cybernetics and Informatics, **8**(1), 43-48, 2010.

[8] V.D. Dmitrienko et al., "Neural networks for determining affinity functions", 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 1-5, 2020, doi: 10.1109/HORA49412.2020.9152830.

[9] L.S. Yampolsky, Neurotechnology and neurosystems, Kiev, Monograph, "Dorado-Printing", 508, 2015.

[10] L. Fausett, Fundamentals of Neural Networks: Architectures, Algorithms and Applications, New Jersey: Prientice Hall. Int., 483, 2006.

[11] N. Tomova, "The emergence of three-valued logics: logical and philosophical analysis", Bulletin of Moscow University. Series 7. Philosophy, Moskow, MSU, 68-74, 2009.

[12] A.S. Karpenko, Development of multi-valued logic, Ed. 3-ht, Moskow, Publishing house LCI, 448, 2010.

[13] A.S. Karpenko, "Bochvar's three-valued logic and literal paralogics", RAS, Institute of Philosophy, Moskow, IF RAS, 110, 2016.

[14] N.E. Tomova, "Natural three-valued logocs and classical logic", Logical Investigations, **19**, 344-352, 2013.