Neural Networks Art: Solving Problems with Multiple Solutions and New Teaching Algorithm

V. D. Dmitrienko*, A. Yu. Zakovorotnyi, S. Yu. Leonov and I. P. Khavina

National Technical University "Kharkov Polytechnic Institute", Kharkov, Ukraine

Abstract: A new discrete neural networks adaptive resonance theory (ART), which allows solving problems with multiple solutions, is developed. New algorithms neural networks teaching ART to prevent degradation and reproduction classes at training noisy input data is developed. Proposed learning algorithms discrete ART networks, allowing obtaining different classification methods of input.

1. PROBLEM FORMULATION AND ANALYSIS OF THE LITERATURE

The human brain processes information flows continuously from the external environment. However, it can modify and update the stored images, and create new, without destroying what previously memorized. Thus it differs significantly from the majority of neural networks as neural networks (NN), trained by back propagation, genetic algorithms, in bidirectional associative memory, Hopfield networks, etc. very often a new way of learning, situation or association significantly distorts or even destroys the fruits of prior learning, requiring a change in a significant part of weights of connections or complete ret raining of the network [1-4]. Impossibility of using the specified NN solve the problem of stability-plasticity, that is a problem of perception and memorization of new information without loss or distortion of existing, was one of the main reasons for the development of fundamentally new configurations of neural networks. Examples of such networks are neural networks, derived from the adaptive resonance theory (ART), developed by Carpenter and Grossberg [5, 6]. ART neural network classifies the input image to one of the known classes, if it is sufficiently similar to or resonates with the prototype of this class. After you have found a prototype with some accuracy given by the special parameter similarity corresponds to the input image, it is modified to be more like the presented image. When the input image was not like any of the existing prototypes, then it is based on a new class. This is possible thanks to the fact that the network has a significant number of unallocated redundant or elements which are not used as long as it is not necessary (if no unassigned elements, the input image is not network response). Thus, new images can create new classes, but did not distort the stored information.

*Address correspondence to this author at the National Technical University "Kharkov Polytechnic Institute", Kharkov, Ukraine; Tel: +380577076198; E-mail: valdmitrienko@gmail.com

It is developed a number of neural network based on adaptive resonance theory [7−11]. However, these neural networks have significant disadvantages: NN adaptive learning resonance theory (ART) often leads to degradation, and reproduction classes receive a unique solution, even in those cases where there are two or more possible and equivalent solutions, the results depend on the training images in order training sequence, etc.

This requires solving the fundamental problem - the architecture and algorithms to improve the functioning of the ART NN. In particular, it is necessary to develop the theoretical foundations of the ART NN with a new architecture and new learning algorithms.

In this paper we consider the most widely used single-module neural network ART-1, proposed to binary input images or vectors. Basic network architecture shown in Fig. 1 in phantom includes three groups of neurons: field F_1 input processing neurons, consists of two layers of elements, recognition layer Y- neurons and control neurons R, G_1, G_2 (Fig. **1**).

Field F_1 consists of two layers – input layer S-elements and the interface layer Z-elements. The input layer receives the image requirements and transmit the information received interface layer neurons and manage neurons R, G_1, and G_2. Each element of Z_i ($i = \overline{1, n}$) interface layer is associated with each element Y_j ($j = \overline{1, m}$) discriminating layer Y two kinds of weighted connections. Signals from the interface layer to layer Y transferred to a bottom-up connections with weights W_{ij}^1, and from discriminating layer to the interface – bonds with weights W_{ji}^2 ($j = \overline{1, m}$, $i = \overline{1, n}$). Due to a large number of bonds in Fig. 1 shows only one pair symbol bonds with weights W_{ij}^1, W_{ji}^2 between interface and recognizing layers of elements.
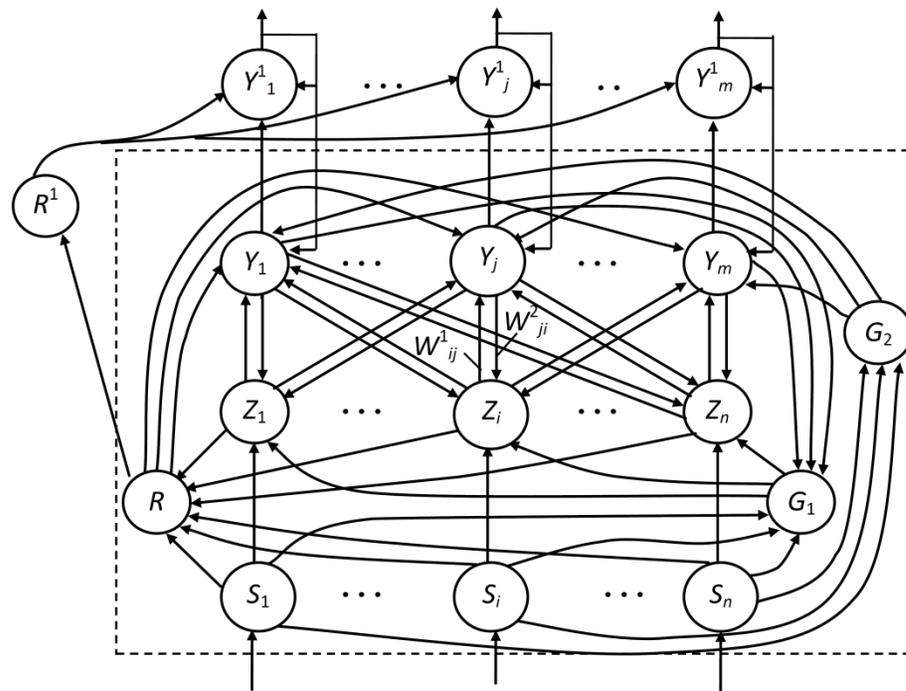
Fig. (1). Architecture Neural Network ART.

Most of the connections shown in Fig. 1 are excitatory: from input layer S- elements to neurons R, G_1, G_2 and Z-layer, from neurons G_1, G_2 respectively to neurons layers Z and Y. Inhibitory signals are transmitted multiple links from the interface elements to R-neuron and from Y-neurons to element G_1. All communications network ART-1 transmit only binary signals 0 or 1.

Each element in the interface or Y-layer neural network ART-1 has three input sources. Arbitrary interface element Z_i ($i = \overline{1, n}$) can receive signals from element S_i input layer, from elements Y-layer and from neuron G_1. Similarly, the element Y_j ($j = \overline{1, m}$) can receive signals from the interface elements, neurons R and G_2. To translate the interface neurons layers or recognizing neurons layers in the active state, you must have two sources of input single excitation signals. Since each of the neurons is considered three possible signal sources, the condition of excitation of these neurons has been called " rules two of three".

Layer Y appears as layer of competing neurons. In steady-state conditions, each element Y_j ($j = \overline{1, m}$) recognizing layer located in one of three states:

− active (output signal $U_{\text{out}Y_j}$ neuron Y_j equal "1");

− inactive ($U_{\text{out}Y_j} = 0$, but neuron may compete);

− braked ($U_{\text{out}Y_j} = -1$ and neuron not allowed to compete)

upon presentation of the current input image).

In the initial state neurons R, G_1, G_2 and the input layer S have zero output signals. at receipt on inputs S-elements binary components the displaying images some of them, received a single input signals, becomes to excited state (U_{out}

= 1). Exciting signals from the outputs of these neurons transform to the state "1" neurons G_1, G_2, as well as supplied to respective inputs of the interface layer neurons. Interface layer neurons, which received the unit signals from neurons of the input layer and element G_1, by the rule two of the three transferred to the active state and send their unit excitatory signals on relations with weights W_{ij}^1 ($i = \overline{1, n}$, $j = \overline{1, m}$) to the inputs of neurons Y_j recognizing layer. Recognizing layer neurons transferred to the active state by rule two of the three, receiving excitatory signals not only from the elements of the interface layer, but from the element G_2. Output signals active Y- neurons determined by the ratio

$$U_{\text{out}Y_j} = \sum_{i=1}^{n} W_{ij}^1 U_{\text{out}Z_i}, \quad j = \overline{1, m},$$

and satisfy condition: $0 < U_{\text{out}Y_j} < 2$.

Then in Y-layer recognizing neurons occurs lateral process of selection single neuron Y_J with the highest output signal. All neurons Y-layer, besides winning Y_J, transferred to inactive status "0" ($U_{\text{out}Y_j} = 0$), and the winning neuron − to a state with a single output. Single signal winning Y- neuron retards managing neuron G_1, and also comes with on relations with weights W_{Ji}^2 to the inputs of the elements of the interface layer. Since the elements of the interface layer conform to the rule two of the three, then in the absence of the excitation signal from the neuron G_1 in an active state remain only those interface elements, which receiving unit signals from the element of the input layer,

and from winning Y_J-neuron recognizing layer. Braking signals from active elements of the interface layer is applied to the inputs *R*-element, which receives also excitatory signals from the input layer neurons. With the help of *R*-element calculated value of parameter similarity *p* between the input image and image, stored in the weights of winning neuron connections:

$$p = \|U_{\text{outZ}}\| / \|U_{\text{outS}}\| = (\sum_{i=1}^{n} U_{\text{outZ}_i}) / (\sum_{i=1}^{n} U_{\text{outS}_i}),$$

where $\|U_{\text{out}}\|$ – norm of the vector U_{out}; $U_{\text{outZ}} = (U_{\text{outZ}_1}, ..., U_{\text{outZ}_n})$, $U_{\text{outS}} = (U_{\text{outS}_1}, ..., U_{\text{outS}_n})$ – vector output signals, respectively *Z*- and *S*-layer elements.

If the calculated similarity value indicates coincidence with a given accuracy input and stored in the memory image, then the network resonance occurs. In this case on the output *R*-element will be zero output signal, and will be training weights winning neuron connections. When parameter value is less than the similarities preassigned on the output control *R*-element single signal appears, which victory *Y*-neuron braked ($U_{\text{outY}_J} = -1$) and actually deprived of the opportunity participate in competitions upon presentation of the current image. Then in *Y*-layer selects the new winning neuron. If the input image was not like none of the stored, then all distributed *Y*-neurons, eventually, are inhibited, and the winner becomes undistributed *Y*-element, who remembers in his weights new image.

Neural network ART-1 training is usually done using a fast learning algorithm [9, 12].

Fast Algorithm for training the neural network ART-1

The algorithm adopted such designations: S^k – *n*-measured input vector, $k = 1, ..., q$;

q – number of input vectors;

$U_{\text{outZ}} = (U_{\text{outZ}_1}, ..., U_{\text{outZ}_n})$ – *n*-measured vector of output signals of interface elements layer;

$\|X\|$ – norm of vector *X*;

p – parameter of likeness between an entrance vector and vector, kept in the scales of connections of winning neuron *Y*-layer; range of legitimate values of parameter: $0 < p < 1$;

W_{ij}^1 – weight of connection from an element Z_i ($i = 1, ..., n$) to the element Y_j ($j = 1, ..., m$); range of legitimate initial values $0 < W_{ij}^1 \leq 1/(L - 1 + n)$; recommended initial value: $W_{ij}^1 = 1/(1 + n)$; all calculations are executed taking into account the use *K*-value signals;

L – constant, excelling unit; recommended value: $L = 2$;

W_{ji}^2 – weight of connection from an element Y_j to the element Z_i; recommended initial value.

Step 1. The first step is initializing parameters *L*, *p* and weights W_{ij}^1 of connections from an interface layer to a

weekend, and connections W_{ji}^2 ($i = 1, ..., n$; $j = 1, ..., m$) from a weekend layer to the interface.

Step 2. The terms of stop are analyzed, and while they are not executed, steps 3 – 14 of algorithm will be realized.

Step 3. For every teaching entrance vector S^k ($k = 1, ..., q$) are executed steps 4 – 13.

Step 4. Set zero signals of all recognizing elements of weekend *Y*-layer:

$$U_{\text{outY}_j} = 0, \quad j = 1, ..., m.$$

By an entrance vector S^k activating *S*-elements of input layer:

$$U_{\text{outS}_i} = S_i^k, \quad i = 1, ..., n.$$

Step 5. The norm of output signals vector neurons of entrance layer is calculated:

$$\|U_{\text{outS}}\| = \|S^k\| = \sum_{i=1}^{n} S_i^k.$$

Step 6. For the elements of interface layer entrance and output signals are formed:

$$U_{\text{inpZ}_i} = U_{\text{outS}_i}, \quad i = 1, ..., n,$$

$$U_{\text{outZ}_i} = U_{\text{inpZ}_i}, \quad i = 1, ..., n.$$

Step 7. Settles accounts for output signal of every not brakes *Y*-neuron:

if $U_{\text{outY}j} \neq -1$, then $U_{\text{outY}_j} = \sum_{i=1}^{n} W_{ij}^1 U_{\text{outZ}_i}$, $j = 1, ..., m.$

Step 8. While not found *Y*-neuron of output layer, weight vector of which in accordance with the specified value of likeness parameter of *p* corresponds an entrance vector S^k, are executed steps 9 – 12.

Step 9. In *Y*-layer ascertain a neuron Y_J, meeting a condition

$$U_{\text{outY}_J} \geq U_{\text{outY}_j}, \quad j = 1, ..., m.$$

If such elements few, gets out an element with the smaller index. If $U_{\text{outY}_J} = -1$, then all elements are put on the brakes and entrance image can not be classified or memorized.

Step 10. Calculating output signals *Z*-elements of interface layer with the use *K*-value boolean operations:

$$U_{\text{outZ}_i} = U_{\text{outS}_i} W_{Ji}^2, \quad i = 1, ..., n.$$

Step 11. The norm of output signals vector of interface layer is calculated:

$$\|U_{\text{outZ}}\| = \sum_{i=1}^{n} U_{\text{outZ}_i}.$$

Step 12. Checked up the condition of teaching possibility of the selected neuron Y_J.

If $\left\|U_{\text{ouℤ}}\right\|/\left\|S^k\right\| < p$, that condition is not executed end element Y_J is braked, i.e. $U_{\text{out}Y_J} = -1$. After it passing carried out to the step 8 of algorithm.

If $\left\|U_{\text{ouℤ}}\right\|/\left\|S^k\right\| \geq p$, condition of teaching neuron Y_J possibility is executed and progressing to the next step of algorithm.

Step 13. Adapt oneself weight of connections of element Y_J taking into account the use K-value signals:

$$W_{iJ}^1 = \frac{L U_{\text{ouℤ}_i}}{(L - 1 + \left\|U_{\text{ouℤ}}\right\|)}, \, i = 1, \ldots, n,$$

$$W_{Ji}^2 = U_{\text{ouℤ}i}, \, i = 1, \ldots, n.$$

Step 14. The conditions of stop are checked up.

The conditions of stop can be: absence changes of scales net W_{ij}^1, W_{ji}^2 during an epoch, achievement of the specified number of epochs and so on.

Step 15. Stop.

II. NEW ALGORITHMS FOR TRAINING NEURAL NETWORKS ART-1 FOR NOISY INPUT

During practical calculations revealed deficiencies fast learning algorithm NN ART-1, do not allow to use them effectively in real systems management and recognition, where the original data are noisy. In noisy input image fast training of the neural network algorithm can lead to degradation and reproduction classes, that is, the learning algorithm becomes inoperative. Demonstrate this lack of a specific example of a learning neural network ART-1.

Example 1. Suppose you want by using a neural network divided into two classes set of vectors:

$M = \{ S^1 = (1,1,1,1,0,0,0,0),$ $S^2 = (0,1,1,1,0,0,0,0),$

$S^3 = (1,0,1,1,\,0,0,0,0),$ $S^4 = (1,1,0,1,0,0,0,0),$

$S^5 = (1,1,1,0,0,0,0,0),$ $S^6 = (0,0,0,0,1,1,1,1),$

$S^7 = (0,0,0,0,0,1,1,1),$ $S^8 = (0,0,0,0,1,0,1,1),$

$S^9 = (0,0,0,0,1,1,0,1),$ $S^{10} = (0,0,0,0,1,1,1,0)\}$.

Classification mentioned vectors can be performed in many ways. One of the most suggests itself − assigned to the first class of vectors S^1, S^2, ..., S^5, to the second class − vectors S^6, S^7, ..., S^{10}, i.e. to build classes of vectors, differing from each other by not more than two components. These classes of vectors can be viewed as classes, which obtained by the impact of noise of vectors S^1, S^6, resulting which in the vectors S^2, S^3, S^4, S^5, S^7, S^8, S^9, S^{10} lost by one unit component. Let's try to get this classification using neural network ART-1. Let us take a network with parameters: $m = 8$ − number Y-neurons in discriminating network layer; $n = 8$ − number of neurons in the input layer of the network; $p = 0,75$ − similarity parameter; selected

such numeric values based on the fact, that the vectors S^1, S^2, ..., S^5; S^6, S^7, ..., S^{10}, in each class differ from each other by not more than two components, and the parameter of similarity is calculated using only single component vectors; $W_{ij}^1 = 1/(1 + n) = 0,111$ − initial weights values of connections W_{ij}^1 ($i = \overline{1,8}; \, j = \overline{1,8}$); $W_{ji}^2 = 1$ − initial weights values of connections W_{ji}^2 ($j = \overline{1,8}; \, i = \overline{1,8}$).

During training the neural network obtained the following results.

Presentation the neural network of the first vector S^1 and calculation the equilibrium weights of connections by the formulas (1) leads to the following weight matrix:

$$W^1 = \begin{vmatrix} 0,400 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0,400 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0,400 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0,400 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 & 0,111 \end{vmatrix},$$

$$W^2 = \begin{vmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{vmatrix}.$$

Analyzing example shows, that to determine the number of selected classes of input vectors rather one of the matrices W^1, W^2 weights of connections, so in the future will only use a matrix of weights connections W^1 as a more informative.

Consistent presentation of the neural network of the following four vectors S^2, S^3, S^4, S^5 and calculation of the equilibrium weights of connections will result to following results. Presentation of the neural network training vector S^2 leads to, that in the first column of the matrix W^1 and the first row of the matrix W^2 there will be only three non-zero coefficient. Therefore, when the similarity parameter equal 0,75, subsequent training the neural network by vector S^3 cause the distribution one more recognizing Y-neuron. Further training neural network by vectors S^4, S^5 require the allocation of two more neurons from

discriminating layer neural network. As a result, the matrix W^1 weighting coefficients connections between layers interface and recognizing neurons will have the form:

$$W^1 = \begin{vmatrix} 0 & 0,500 & 0,500 & 0,500 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0,500 & 0 & 0,500 & 0,500 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0,500 & 0,500 & 0 & 0,500 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0,500 & 0,500 & 0,500 & 0 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0 & 0 & 0 & 0 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0 & 0 & 0 & 0 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0 & 0 & 0 & 0 & 0,111 & 0,111 & 0,111 & 0,111 \\ 0 & 0 & 0 & 0 & 0,111 & 0,111 & 0,111 & 0,111 \end{vmatrix}.$$

It is easy to verify that subsequent submission to the input of the neural network as a vector S^6, S^7, ..., S^{10}, as well as vectors S^1, S^2, ..., S^5 does not lead to a change in the first four columns of the matrix W^1 and the first four rows of the matrix W^2. Consistent presentation of the neural network of the following four vectors S^6, S^7, S^8, S^9, S^{10} and calculation of the equilibrium weights of connections results in the release of four other neurons of the neural network from recognizes layer. As a result, the matrix W^1 weighting coefficients connections between layers of interface and recognition will be the following:

$$W^1 = \begin{vmatrix} 0 & 0,500 & 0,500 & 0,500 & 0 & 0 & 0 & 0 \\ 0,500 & 0 & 0,500 & 0,500 & 0 & 0 & 0 & 0 \\ 0,500 & 0,500 & 0 & 0,500 & 0 & 0 & 0 & 0 \\ 0,500 & 0,500 & 0,500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,500 & 0,500 & 0,500 \\ 0 & 0 & 0 & 0 & 0,500 & 0 & 0,500 & 0,500 \\ 0 & 0 & 0 & 0 & 0,500 & 0,500 & 0 & 0,500 \\ 0 & 0 & 0 & 0 & 0,500 & 0,500 & 0,500 & 0 \end{vmatrix}.$$

Thus, the analysis of the solution shows an example that the network ART-1 during training can not share a set of vectors S^1, S^2, ..., S^{10} into two classes, because it is reproduction classes. Vectors S^1, S^6, which could become the reference vectors of the two classes degraded in the process of presenting the following vectors, which leads to duplication of classes.

Failure to solving the problem of classification of the set of vectors associated with features of the architecture and neural network learning algorithm ART-1. First, the network connections in the weights of a neuron remembers recognizing the intersection of binary input vectors. Second, the proximity of binary vectors is defined by using parameter similarity, taking into account only single elements compared vectors. In the analyzed example intersection of set M_1 binary vectors S^1, S^2, ..., S^5 and set M_2 of

vectors S^6, S^7, ..., S^{10} give the vector with all zero components:

$$S^0 = S^1 \cap S^2 \cap S^3 \cap S^4 \cap S^5 = S^6 \cap S^7 \cap S^8 \cap S^9 \cap S^{10} = (0, 0, 0, 0, 0, 0, 0, 0).$$

Thus, at the crossing of considered sets of vectors completely lost the information about the individual components of the binary vectors. Storing information in the form of crossing of the input binary vectors or an image in a neural network also leads to loss of information. For small values of similarity parameter may be lost most the information about classified or recognizable images. In this regard, there is an idea about memorizing information as a union, and not crossing binary images or vectors. In this case, in this example we have:

$$S^1 \cup S^2 \cup S^3 \cup S^4 \cup S^5 = S^1;$$
$$S^6 \cup S^7 \cup S^8 \cup S^9 \cup S^{10} = S^6.$$

Remembering two vectors S^1, S^6 results in weight matrices connections W^1, W^2, with which could correctly classified all of the input image, if after the presentation of input vectors S^1, S^6 there was no adaptation of neural weights, change it memory.

However, remembering association binary vectors without a teacher also has its drawbacks, since can easily appear matrix of weights connections, having in their respective columns (matrix W^1) or rows (matrix W^2) for distributed *Y*-neurons minimum number of zero elements or even not having them at all. Therefore, when memorizing the combining binary vectors needed correction the work of algorithm or participating teacher.

Examples of successful use at training a neural network crossing or union of the input images indicate to the fact, that, apparently, may be prepared learning images also with using a combination of these operations, as well as using other operations on binary sets.

Thus, we can conclude, that adapting the weights of network connections in the form, wherein it is used in a neural network ART-1, is the lack of the network. The disadvantage, as shown by the above example, also may be the lack of teacher. Similar problems also arise in training the continuous NN ART.

Thus, the adaptation of weights of distributed recognizing communications *Y*-neurons net in the form, wherein it is used in the network ART-1, often fails to solve even trivial tasks of classification and training, then there is a significant lack of the network.

Moreover:

– final result of learning discrete network ART may depend on presentation of a sequence images;

– no training mode networks with teacher, that restricts the class solved by neural network problems classification and recognition.

To prevent degradation and reproduction classes, that is, for the correct and stable operation of the modules on based of neural network ART-1, proposed ban adaptation weights of connections of distributed recognition neurons. For this, at step 11 learning algorithm adaptation of connections weights of recognizing elements produced only, if these neurons were distributed earlier. Otherwise − weight relationships recognition elements not adapted. With this algorithm, the set of training M vectors, if they are served in the order, wherein they recorded in the set, is divided into two classes − to the first class relate vectors S^1, S^2, ..., S^5, to the second class − vectors S^6, S^7, ..., S^{10}, that is, it turns out the most natural classification vectors. However, if the order of training vectors changes, for example, vectors S^1, S^6 in their subsets not subject to the net first, then changed also classification of the vectors.

III. TRAINING ALGORITHMS OF NEURAL NETWORK ART-1 WITHOUT ADAPTATION WEIGHTS OF CONNECTIONS DISTRIBUTED RECOGNITION NEURONS

General lack of different systems classification using a variety of neural networks or other approaches or methods consists in, that it turns out only one classification, due to the method of teaching or raised by teachers. At the same time, when there N various objects, it is possible to obtain 2^N. Polar cases of these classifications: one class contains all the objects; each of the N classes contains one object. At $N = 10$ number of possible ways of classifying exceeds 1000. Therefore require different approaches to select the best classification. One of these approaches may be specifying the number of classes or minimization, or maximizing the number of classes under certain restrictions. Introduction such restrictions necessary, because otherwise while minimizing the number of classes will be obtained a single class, and while maximizing the number of classes is obtained N classes. One such limitation can act parameter of similarity images. When parameter similarity, equal to unity, it is obtained N classes, when parameter similarity, equal or near to zero − one class. As a second constraint can act the presence or absence of ability to adapt weights of connections distributed recognition neurons.

In this regard, the dependence of the final result of learning a discrete network ART from sequence of presentation training images may be considered not as a drawback NN, but as its dignity, can create different classification at purposeful change input training sequence. Therefore, for discrete neural network ART-1 learning offers many new learning algorithms:

1. Training algorithms, in which the input image can not adapt weights of connections distributed recognition neurons after the resonance (used in noisy data):

1.1. The learning algorithm of neural networks ART-1 the sequence of images, ordered by increasing values of their norm, wherein, the first to network are presented images with minimal norm. Applies in cases, when it is desired to obtain an increased number of classes of images.

1.2. The learning algorithm of neural networks ART-1 the sequence of images, ranked by descending value of their norms, wherein, the first to network are presented images with the maximum norm. Applies in cases, when it is desired to obtain minimum number of image classes.

1.3. The learning algorithm NN ART-1 the sequence of images, ranked by teacher, wherein in the training sequence can be used subsequence as with increasing, as with decreasing norm of the input images.

1.4. The learning algorithm discrete neural network ART, in which the initial values of weights connections are given by teacher, using complex input images, which are derived from initial using Boolean operations "AND", "OR", "NOT", other mathematical operations, heuristics or combinations thereof forming methods of training images.

2. The learning algorithm, using a joint application of the algorithm rapid learning NN and learning algorithms without adaptation of network weights upon the occurrence of resonance.

An example of such algorithm can serve an algorithm, wherein in the first stage (first epoch) used the fast algorithm of neural network ART-1 learning, and at later epochs used learning algorithms without adaptation weights of connections recognizing distributed neurons of network upon the occurrence of resonance.

IV. ART NEURAL NETWORK FOR SOLVING PROBLEMS WITH MULTIPLE SOLUTIONS

A general lack systems of recognition and classification based on neural networks, including networks ART − obtain a unique solution, even in cases, when there are two or more possible and equivalent solutions. In recognition of such images recognition result may depend on the order of filing in the training mode reference pictures, given by teacher. Show it on the example.

Example 2. Recognition image, located on the borders of several classes.

Take ART-1 network with parameters: $m = 5$ − number Y-neurons in discriminating network layer; $n = 12$ − number of neurons in the input layer of the network; $p = 0,5$ − similarity parameter images.

Let the teacher used as reference the following three vectors: $S^1 = (1,1,1,1,0,0,0,0,0,0,0,0)$,

$S^2 = (0,0,0,0,0,0,0,0,1,1,1,1)$,

$S^3 = (0,0,0,0,1,1,1,1,0,0,0,0)$, who served in the training mode in the following order: S^1, S^2, S^3.

Submission a neural network these vectors and calculation the equilibrium weights of connections from the formulas (1) leads to the following matrix W^2 weighting coefficients:

$$W^2 = \begin{vmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{vmatrix}.$$

When the input of the neural network considered recognition mode vector $S^* = (0,0,1,1,1,1,0,0,0,0,0,0)$ then it will be assigned to the first class, that is, a single output signal will be on the output of recognizing neuron Y_1, because this neuron has a minimum index. If chosen neuron with maximal index, then the winner neuron became neuron Y_2. A similar situation arises when applied to the input neural network vector $S^{**} = (0,0,0,0,0,0,1,1,1,1,0,0)$, which will be assigned to the second class because training image S^2 was filed before the image S^3. To exit from this situation necessary to use a neural network ART, which ability to define multiple solutions (Fig. **1**).

To extend the capabilities of discrete neural network ART-1 and get all the possible solutions of the problem of recognition to the basic architecture of ART-1 added another control neuron R^1, the inverting outputs of the neuron R, and a recording neurons layer Y_j^1 ($j = \overline{1,m}$). Each neuron Y_j^1 associated with binary unidirectional communication with the corresponding neuron Y_j.

Before recognition mode neurons Y_j^1 ($j = \overline{1,m}$) translated into a passive state ties on the circuits, not shown in Fig. (**1**). Neurons Y_j^1 ($j = \overline{1,m}$), go to the active state by the rule "two out of three" − in the presence of single neurons of the output signals Y_j and control neuron R^1. The unit output signal of the neuron Y_j^1 through a feedback circuit fixes its unit output and inhibits neuron-winner Y_j.

After that, the network begins searching for a new neuron-winner. The search process continues until, until all recognizing distributed neurons not will inhibited. In this case on the outputs of neurons Y^1 may not be no unit signal, since the input image not similar to any of the images, stored in the weights of network connections, or on outputs of Y^1-neurons may be one or more unit signal, respectively indicating belonging the input image to one or several classes of images.

CONCLUSION

Thus, it is developed a new architecture of discrete neural network adaptive resonance theory, which allows to specify one, two and an increasing number of decisions (if they exist) in pattern recognition and classification of black-and-white images.

Proposed new learning algorithms of discrete neural networks ART, prevent degradation and reproduction classes at training networks with noisy input data.

Overcome one of the drawbacks common classification systems based on neural networks, consists in the fact that it turns out only one classification, due to the method of teaching or raised by teacher. Proposed learning algorithms of discrete neural networks ART, allowing to obtain a variety of methods classifying input data.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

Decleared none.

REFERENCES

[1]   Suzuki K. Artificial neural networks: Architectures and applications. Publisher: In Tech 2013; p. 256.
[2]   Bianchini M. Handbook on neural information processing (Intelligent Systems Reference Library). Springer 2013; p. 499.
[3]   Huajin T, Kay CT, Zhang Y. Neural Networks: Computational models and applications. Publisher: Springer, 2007; p.322.
[4]   Zadeh L. Neural networks theory. Publisher: Springer, 2007; p. 421.
[5]   Grossberg SA. Competitive learning: from interactive activation to adaptive resonance. Cognitive Sci 1987; 11: 23-63.
[6]   Carperter GA, Grossberg SA. Massively parallel architecture for self-organizing neural pattern recognition machine. Comput Vis, Graph Image Process 1987; 37: 54 -115.
[7]   Leonov S Yu, Dmitrienko VD, Gladkikh TV. K-value adaptive resonance theory of the neural network for analyzing the operability of computing devices. World App Sci J 2014; 30 (12): 1932-8.
[8]   Bodjanskij EV, Rudenko OG. Iskusstvennye nejronnye seti: arhitektury, obuchenie, primenenija. -Har'kov. TELETEH 2004; p. 372.
[9]   Dmitrienko VD, Havina IP, Havin VL, Verezeb NV. Modelirovanie tehnologicheskih processov mehanoobrabotki metodami iskusstvennogo intellekta. Har'kov NTMT 2009; 224 s.
[10]   Quek C, Tung WL. A novel approach to the derivation of fuzzy membership functions using the FalconMART architecture. Pattern Recognit Lett 2001; (22): 941 -58.
[11]   Galushkin AI. Nejrokomp'jutery i ih primenenie na rubezhe tysjacheletij v Kitae. V 2-h tomah. Tom M Gorjachaja linija-Telekom 2004; 367 s.
[12]   Fausett L. Fundamentals of Neural Networks: Architectures, Algorithms and Applications. -New Jersey: Prentice Hall. Int., Inc., 1994; p. 461.