PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Intellectual system for images restoration

Igor Mardare

Igor Mardare, "Intellectual system for images restoration," Proc. SPIE 5822, Information Technologies 2004, (21 February 2005); doi: 10.1117/12.612043

SPIE.

Event: Information Technologies 2004, 2004, Chisinau, Moldova

Intellectual system for images restoration

Igor Mardare
Department of Design and Manufacturing of Electronic Apparatus
Technical University of Moldova
Stefan Mare Av,168, Chisinau, MD2012, Republic of MOLDOVA
Tel:(3732)237505 Fax:(3732)235236 E-mail: mardarei@mail.md, imardare@block1-gw.utm.md

ABSTRACT

Intelligence systems on basis of artificial neural networks and associative memory allow to solve effectively problems of recognition and restoration of images. However, within analytical technologies there are no dominating approaches of deciding of intellectual problems. Choice of the best technology depends on nature of problem, features of objects, volume of represented information about the object, number of classes of objects, etc. It is required to determine opportunities, preconditions and field of application of neural networks and associative memory for decision of problem of restoration of images and to use their supplementary benefits for further development of intelligence systėms.

Keywords: restoration of images, neural networks

I. FUNCTIONAL ELEMENTS OF IMAGE RESTORATION SYSTEM

Dependent on the available information, skills and knowledge, person uses various resources and technologies: memory, associations, the analogies, ready decisions, both standard specialized and universal algorithms and techniques of decisions, search and generation of new algorithms etc. Similarly, structure of intellectual system of restoration-recognition of images is defined by available methods of analytical technologies, functional devices and algorithmic resources for solving of intellectual problems and features of representation of input information subjected to restoration-recognition. It is also necessary the arbitrator estimating and choosing for concrete condition the most effective method of decision. And at last, it should be trained and should accumulate received knowledge, as it is necessary for any intellectual system.

Required functional devices for decision of problem of restoration-recognition of images are defined from available classes of objects. Presence of defective and true (standard) images assumes application of neural networks trained with the teacher. Defective images, and presence or absence of true images provide application of neural networks trained without the teacher. It is possible the variant when presence of standard image is not an obligatory condition of application of training with the teacher, ignoring the standard image it is possible to apply training without the teacher. Defective images and a set of true images allow using associative memory. The arbitrator on basis of true image defines application of one of kinds of neural networks (figure 1).

Information Technologies 2004, edited by Andrei M. Andriesh, Veacheslav L. Perju,
Proceedings of SPIE Vol. 5822 (SPIE, Bellingham, WA, 2005)
0277-786X/05/$15 · doi: 10.1117/12.612043

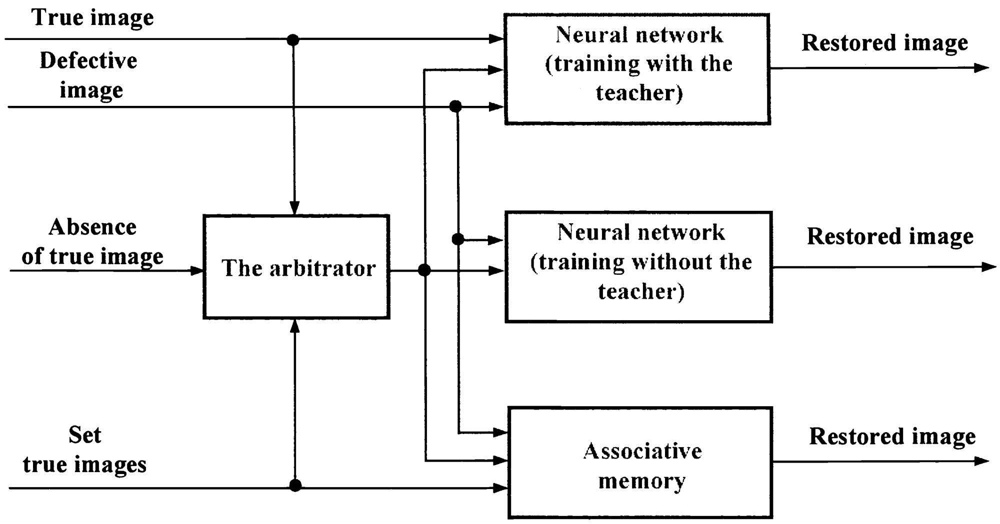


Figure 1: Structure of intelligence system of recognition-restoration of images.

1.1 The neural network trained with the teacher

The neural network trained with the teacher, allows to find unknown functional dependence between defective $x_n=\{x_1, x_2, ..., x_N\}$ and true image of object c_q on basis of set of experimental data such as "input-output": x_1-y_n, x_2-y_n, ..., x_N-y_n, where $y_n=c_q$. Required function $c_q=F(x_n)$ depends on values of sinaptical connections w_{mk} of all neurons of network. Therefore, having set some structure of network, it is necessary to find optimum values of all variable sinaptical factors w_{MK} according to some criterion. Such criterion, in particular, can be criterion of a minimum of average square of mistake on training set Ę. During training the network changes its parameters by repeated updating of sinaptical weights w_{MK}. Weights gradually become such, that each input vector x_N (the defective image) developed a required output vector $y_n=c_q$ (the true image). Result of training is required reflection $x_1 \to c_q$, $x_2 \to c_q$, ..., $x_N \to c_q$. Further, due to generalizing ability of network using smooth continuous activational functions, it can be received correct results for input vector x_{N+1}, not too distinguished from vectors of training set x_n and not met at training. Thus, any defective image x_n, belonging to given class C_Q: $x_n \in C_Q$, can be restored.

For decision of problem of restoration it is necessary to define architecture of a network. For some types of problems there exist optimum configurations of networks. If the problem cannot be reduced to known types it is necessary to synthesize new network architecture. Now are created networks of various configurations, but only the level-by-level organization of neurons, forming multilayered *networks of direct distribution* in which the output of one layer is input of the subsequent, copies layered structures of certain regions of brain.

As input vector x_n should come to everyone neuron of input layer of network the number of inputs of neurons coincides with number of inputs of network and is defined by dimension of vector of attributes $x_n=(x_1, x_2, ..., x_K)$.
As activation function is used *sigmoid function* (Firm's function), which is rather close to nonlinear transfer characteristic of biological neuron. The main advantage of this function consists in its differentiability on all absciss axis and simple derivative expressed by this function. Fast calculation of derivative accelerates training, and the continuity of derivative allows training of network by gradient methods.
The number of outputs of network is equal to number of inputs according to the requirement of harmony of defective and true images of objects.

Work of multilayered neural network is described by the following system:

$$\begin{cases} x_{m}^{h} = \sum_{k=1}^{K} w_{mk}^{h}\, x_{mk}^{h}\,, \\ y_{m}^{h} = f\,(x_{m}^{h})\,, \\ x_{m}^{h+1} = y_{m}^{h}\;. \end{cases} \tag{1}$$

where k – input number;
m – neuron's number in a layer;
h – layer's number.

Each layer calculates nonlinear transformation from a linear combination of signals of previous layer. In result approximation of required multivariate function $\boldsymbol{y}_n=\boldsymbol{F}(\boldsymbol{x}_n)$ at corresponding parameters of network is achieved.

1.2 The neural network trained without the teacher

The neural network trained without the teacher, does not require a standard vector $\boldsymbol{c}_q$ for output of network and, hence, does not demand comparison with the predetermined answers. The training set will consist only of input vectors $\boldsymbol{x}_n$. Network of Kohonen realizing the given model of training, is a competitive neural network in which neuron compete with each other for the right in the best way to be combined with input vector $\boldsymbol{x}_n$. Wins the neuron, whose vector of weights $\boldsymbol{w}_{mk}$ is the closest to input vector $\boldsymbol{x}_n$. Neuron with the maximal value of output $\boldsymbol{y}_m$ is the winner. Its output becomes equal to "one", when the others neuron outputs are equal to "zero".

The most similar input vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$ make active the same neuron. The training algorithm arranges weights $\boldsymbol{w}_{mk}$ of won m-th neuron and its neighbors so that to correspond to all close input vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$. Hence, to an input vector $\boldsymbol{x}_n$ from given class $\boldsymbol{C}_q$ corresponds the found vector $\boldsymbol{w}_{mk}$, representing the true image $\boldsymbol{c}_q$, which value before training was unknown ($\boldsymbol{w}_{mk}=\boldsymbol{c}_q$). Thus, the result of training depends only on structure of input data and as a result of training is got ability of classification of input vectors in groups of similar vectors.

Training is necessary when true image $\boldsymbol{c}_q$ is absent. When true image is present known value of vector of a nucleus $\boldsymbol{c}_q$ is given to a weight vector of neuron $\boldsymbol{w}_{mk}$. As a result of functioning of network classification of input vectors is made, that means, that restoration is carried out.

1.3 Associative memory

Associative memory allows by a fragment of information $\boldsymbol{x}_n$ (the defective image – a part of true image) to extract from memory all required information c_q (the true image) (figure 2).

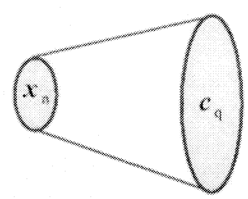


Figure 2: Finding of the standard $\boldsymbol{c}_q$ by a fragment of input information $\boldsymbol{x}_n$.

Let's consider set of standard vectors $\boldsymbol{c}_q=\{c_1, c_2, ..., c_Q\}$, stored in neural network. Let's use as a measure of affinity of two vectors their scalar product: $d(\boldsymbol{x}_n, \boldsymbol{c}_q) = (\boldsymbol{x}_n, \boldsymbol{c}_q)$. The more similar are the vectors, more the measure of affinity is. Then we may expect, that change of vector $\boldsymbol{x}_n$ by time under the law

$$d\boldsymbol{x}_n = \sum_{q=1}^{Q} c_q (c_q, x_n)\, dt\,, \tag{2}$$

will lead to concurrence of vector $\boldsymbol{x}_n$ with the most similar standard $\boldsymbol{c}_q$, i.e. the required association will be found.

Weight factors are determined as follows:

$$w_{ij} = \begin{cases} \sum_{q=1}^{Q} c_{q\,i} c_{q\,j}, & i \neq j \\ 0, & i = j, \end{cases} \tag{3}$$

where i and j – indexes of presinaptical and postsinaptical neurons;
$c_{q\,i}$, $c_{q\,j}$ – i-th and j-th elements of q-th vector-sample.

In associative memory training is reduced to unitary calculation of weight factors of sinaps w_{ij} (3) on basis of information about standards $\boldsymbol{c}_q$. Thus, network simply remembers standards as weight factors before real data receipt on its input.
If weights of network are defin under formula (3), then each standard will represent a local minimum of function of energy, and gradient of which in this point will be equal to zero. However, network not always converges to one of standards. For increase of probability of convergence it is necessary that q vectors c_q, remembered by network, were poorly correlated, i.e. measure of affinity d(c_q, c_{q+1}), ∀q, q+1<Q should be small. Numerical values of maximum permissible affinity of standards strictly are not determined. At infringement of condition of small correlation of standards, the decision, received by network, represents certain average standard, which combine features of many remembered objects.

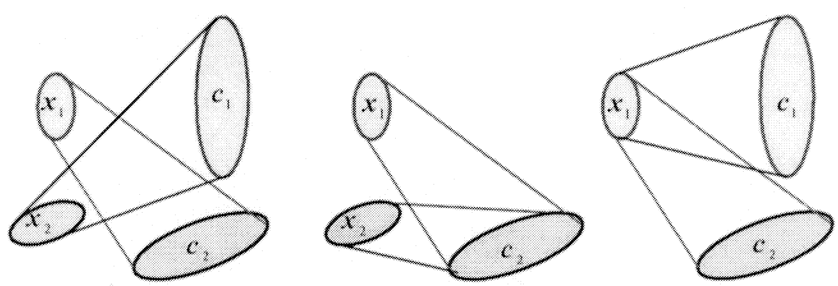
The problem solved by associative memory is formulated as follows. According to input vector, the network will create on output one of remembered before vectors, which is most similar to the given input vector. Or in other words, at a known exemplary set of binary signals (the standard c_q), network should from some no ideal input signal $\boldsymbol{x_n}$ find (recollect) corresponding association $\boldsymbol{A}_{km}$ (if it is present) under the partial information or to give the conclusion about absence of associations for the input data.

Let input object be characterized by a vector of parameters: $\boldsymbol{x}_n=(x_1, x_2, ..., x_K)=(x_i)$, i=1, 2, …, K. Output vector of network $\boldsymbol{y}_p=(y_1, y_2, ..., y_L)=(y_l)$ may have dimension L differed from input vector: $l=1$, 2, …, L. Each element of input x_i and output y_l vectors is equal either +1, or –1.

Let's designate components of n-th input vector $\boldsymbol{x}_n$ – $x_{n\,i}$. Input vectors $\boldsymbol{x}_n$ form set $\boldsymbol{x}_n \in \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$, and output vectors $\boldsymbol{y}_p$ – set $\boldsymbol{y}_p \in \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_P\}$. If network recognizes ("recollects") certain object by vector $\boldsymbol{x}_n$, then at outputs of network appears vector $\boldsymbol{y}_{p_n}$, corresponding to required association $\boldsymbol{A}_{nq}$, i.e. to vector - sample $\boldsymbol{c}_q$. Then, in the assumption, that dimensions of vectors $\boldsymbol{x}_n$ and $\boldsymbol{y}_{p_n}$ coincide, i.e. $\boldsymbol{y}_{p_n} = \boldsymbol{y}_n$, we have:

$$\boldsymbol{y}_n = \boldsymbol{c}_q = \boldsymbol{A}_{nq}(\boldsymbol{x}_n). \tag{4}$$

When image $\boldsymbol{x}_n$, is not recognized, the output vector $\boldsymbol{y}_n$ does not correspond to any of standards c_q, that means absence of association $\boldsymbol{A}_{nq}$ between input image and object stored in associative memory.
If to associative memory are shown some deformed variants of input object $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$, then network itself can create on output the ideal object $\boldsymbol{c}_q$, which it never met.
If two images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are similar enough, they can cause cross associations: – image $\boldsymbol{x}_1$ will cause object $\boldsymbol{c}_2$, and image $\boldsymbol{x}_2$ – object $\boldsymbol{c}_1$ (figure 3.1). Or, various images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ will cause one object $\boldsymbol{c}_2$ (figure 3.2). It is possible also a case when one image $\boldsymbol{x}_1$ will cause various objects $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ (figure 3.3).



1 2 3

Figure 3: Possible variants of cross associations.

Associative memory possesses *ability of generalization*, i.e. will react not only to certain vector, but also to small variations of this vector. For example, if in part deformed vector moves as vector x_n, then network tends to crea vector y_{nr}, which in turn aspires to correct mistakes in x_n (figure 4).

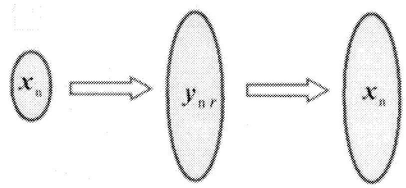


Figure 4: Correction of input vector x_n.

Besides that, vector x_n can be a noised version of standard c_q, but network will be trained to initial vector free from noise. In this case it *takes essence of association*, being trainedby standard c_q, though "saw" only noised approximation x_n (figure 5).

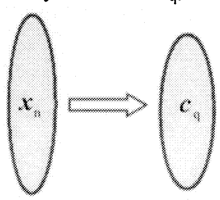


Figure 5: Allocation of standard c_q from noised vector x_n.

The standard object c_q, in its turn, is capable to cause associations of other standard objects c_1, c_2, ..., c_Q. This property is used, for example, at restoration of forgotten objects or search of lost ones. If a person has lost a subject, then he tries to restore the events, connected with loss: to recollect, where did he see this subject last time, in what was dressed, to whom talked, etc. Thus there appears a chain of coherent associations $c_q=A_{nq}(x_n) \rightarrow c_1=A_{q1}(c_q) \rightarrow c_2=A_{12}(c_1)$, determining a consecutive train of thoughts (figure 6), and allowing to find loss.

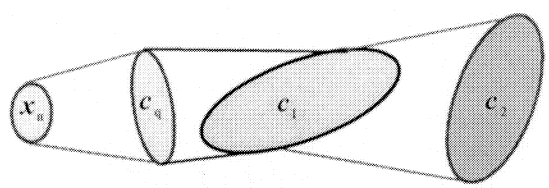


Figure 6: A chain of associations.

Associative memory can be realiz as a neural network, which should restore the defective image, for example, with the imposed noise or containing only important part of initial image. As associative memory can be chosen *bi-directional associative memory*. In comparison with auto associative memory (for example, Hopfield network), bi-directional associative memory enables to find associations between vectors x_n and c_q, which generally have different dimensions.

Let's assume existence of memory inside everyone neuron of 2-nd and 3-rd layers. Also we shall accept, that output signals of neurons change simultaneously with each step of synchronization, remaining constants between these steps. *Long-term memory* is realized in weight files w and w^T. According to (4), each image consists of vector x_n and associated object – vector y_n:

$$y_n = A_{nq}(x_n).$$

Network is capable to remember pairs of associated images x_n and y_n. Associations between vectors are coded in a weight matrix w of third layer. The weight matrix of second layer is equal to transposed matrix of third layer w^T. Preliminary computation of elements of matrix w for j-th neuron is made under the formula:

$$w = \sum_{n=1}^{N} x_n^T y_n . \quad (5)$$

where n – number of pairs of remembered vectors.

2. FUNCTIONING OF SYSTEM OF IMAGE RESTORATION

Functioning of system of image restoration consists of two stages: training and direct restoration.

2.1 Functioning of multilayered network of direct spreading

Training of multilayered network of direct spreading is made with the teacher. Let it is required to train a network to restore defective image $\boldsymbol{x}_1$ (input image). Training with the teacher provides presence of true image $\boldsymbol{c}_1$ (the standard image) concerning which the network will be trained. Input vector $\boldsymbol{x}_n=(x_1, x_2, ..., x_K)$ and standard vector $\boldsymbol{c}_n=(c_1, c_2, ..., c_K)$, form training pair $\boldsymbol{x}_n$-$\boldsymbol{c}_n$. The space of objects $\boldsymbol{X}$ is formed by set of input vectors $\boldsymbol{x}_n=\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ for each class of objects $\{\boldsymbol{C}_1, \boldsymbol{C}_2, ..., \boldsymbol{C}_Q\}\subset\boldsymbol{C}_q$ and standard vector $\boldsymbol{c}_n$ corresponding to each class $\boldsymbol{C}_q$ and $\boldsymbol{X}=\boldsymbol{C}_q$. On figure 7 the training pair of vectors $\boldsymbol{x}_1$-$\boldsymbol{c}_1$, belonging to set $\boldsymbol{C}_1$. is represented.

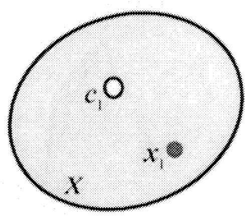


Figure 7: Training pair: intput $\boldsymbol{x}_1$ and standard $\boldsymbol{c}_1$ of object images.

When defective image of object $\boldsymbol{x}_1$ come on input of network it is realized consecutive correction of weights of network $\boldsymbol{w}$ along vector of mistake $\boldsymbol{\xi}(\boldsymbol{w})$, and on output of a network appears a number of images $y_1, y_2, \dots, y_E$ gradually coming nearer to standard image $\boldsymbol{c}_1$ in process of reduction of function of mistakes (figure 8).

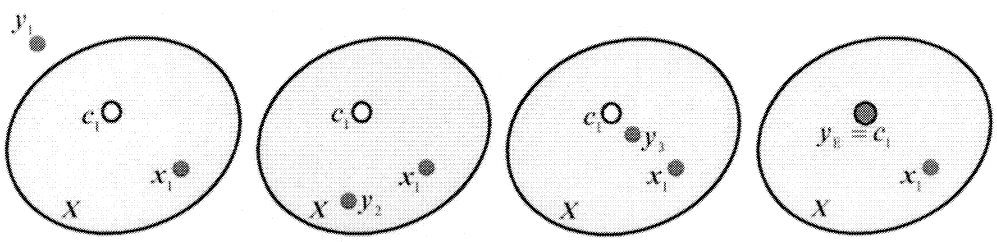


Figure 8: Input vector $\boldsymbol{x}_1$ and sequence of output vectors $y_1, y_2, \dots, y_E$, received as a result of correction of weights of a network.

The mistake $\boldsymbol{\xi}$ in the beginning of training is very great because of random initialization of weights. Therefore network is absolutely inadequate to solving problem and generates on output only vector of noise y_1. Closer by the end of training the mistake $\boldsymbol{\xi}$ gradually decreases, reaching zero or comprehensibly low level. In this case $y_E = \boldsymbol{c}_1$.

For image with other defects $\boldsymbol{x}_2 \subset \boldsymbol{C}_1$, process of approximation to standard vector $\boldsymbol{c}_1$ will take place for smaller number of intermediate images by preliminary adjusted weights of network $\boldsymbol{w}$ and by partial training of networks (figure 9).

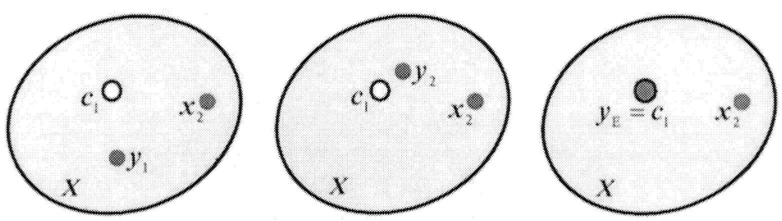


Figure 9: Input image x_2 and sequence of images $y_1, y_2, \dots, y_E$, received as a result of correction of weights of network.

Next image of object with other defects $x_N \subset C_1$ will come nearer to standard image c_1 even faster (figure 10).

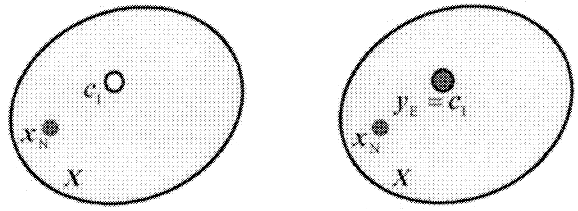


Figure 10: Input image x_N and directly received image y_E, corresponding to standard c_1.

Thus, quality of training of network raises with increase of number of training examples. Too small number of examples can cause "over training" of network when it functions well on examples of training sample, but is bad – on test examples subordinated to the same statistical distribution. Complex objects also demand increase of number of training examples, what is necessary for achievement of generalizing ability of network.

After use of each subsequent training pair x_2-c_1 it is necessary to return to previous training pair x_1-c_1 for correction of weights w as adjustment of weights for x_2-c_1 can partly break adjustment of weights for x_1-c_1. Then it is required to check up again, and if necessary to change adjustments for training pair x_2-c_1. Such process named *training of network*, leads to final specification of weights of network. The less weights vary at transition from x_1-c_1 to x_2-c_1 and on contrary from x_2-c_1 to x_1-c_1, the less probability of distortion already stored objects is. Well teached and trained network will directly restore images with any defects without intermediate images, that is to give out on any input vector $x_1, x_2, ..., x_N$ output vector $y_E = c_1$.

Then training of network on others input $x_1, x_2, ..., x_N$ and output vectors $c_2, c_3, ..., c_Q$, belonging to the following classes $C_2, C_3, ..., C_Q$ is carried out (figure 11).

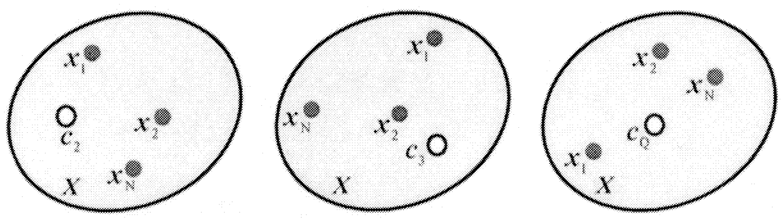


Figure 11: Input $x_1, x_2, ..., x_N$ and standard $c_2, c_3, ..., c_Q$ vector, used for training of neural network.

After use of all training pairs, with each subsequent standard vector c_{n+1} it is necessary to return to training pairs of previous standard vector c_n for correction of weights w as adjustment of weights for vector c_{n+1} can partly break adjustment of weights for vector c_n. It means that direct restoration of previous image is partly broken because of adjustment of weights for subsequent images. Then it is required to check up again, and if necessary to change adjustment for vector c_{n+1}. And, the less weights vary, the less probability of distortion of already stored images is. The increase of dispersion of parameter of network w, increases number of images, been stored in it. The subsequent trainings of network allow with sufficient accuracy to distinguish the big variety of objects with various defects and noises $x_1, x_2, ..., x_N$, belonging to set of classes $C_q=\{C_1, C_2, ..., C_Q\}$,and not used at training of network.

More powerful variant of training with the teacher assumes, that it is known only critical estimation of correctness of output of neural network, but not standard vector c_n.

Essential lack of training with the teacher consists in no interpretness of results. The answer of network is a complex nonlinear function of input values $y_n=F(x_n)$, which don't allow to understand, what features of input example x_n cause reaction of network y_n. However in problems of restoration interpretation is not so important, as result by itself.

2.2 Functioning of Kohonen network

The problem of training of neural network trained without the teacher is to learn network to activate the same neuron for similar input vectors. If number of input vectors is equal to number of nucleus (neuron) and each class is represented by single vector then training is not required. It is enough to give to nucleus value of input vectors, and each vector will activate its Kohonen neuron. If number of classes is less than number of input vectors training will consist in consecutive correction синаптических neuron weights. On each step of training one of vectors is randomly gotten out of initial data set, and then is searched vector of neuron factors most similar to it. The most similar vector of factors defines neuron-winner, which have the maximal output value. Similarity is understood as distance between the vectors, calculated in Evklid space. For i-th neuron-winner we have:

$$|\boldsymbol{x}_n - \boldsymbol{w}_i| = \min_j\{|\boldsymbol{x}_n - \boldsymbol{w}_j|\}. \tag{6}$$

Updating of weight factors is made according to expression:

$$\boldsymbol{w}_i^{t+1} = \boldsymbol{w}_i^t + h_i^t(\boldsymbol{x}_n^t - \boldsymbol{w}_i^t), \tag{7}$$

where $\boldsymbol{w}_i^{t+1}$ – new value of weight which have gained i-th neuron;

$\boldsymbol{w}_i^t$ – previous value of this weight;

h_i^t – function of neighborhood of neuron;

$\boldsymbol{x}_n^t$ – random chosen input vector on t-th iteration.

Training consists of two phases: at initial stage is chosen great enough value of training speed and radius that allows to arrange neuron vectors according to distribution of examples in sample, and then, at small speeds of training exact fine tuning of weights is made. Before the beginning of training it is required to initialize weight factors of neuron $\boldsymbol{w}_{mk}$. Usually to синаптическим neuron weights initially are given the normalized uniformly distributed small random numbers. Training of Kohonen network with uniformly distributed random vectors of weights (nucleus of classes) is graphically presented on figure 12.

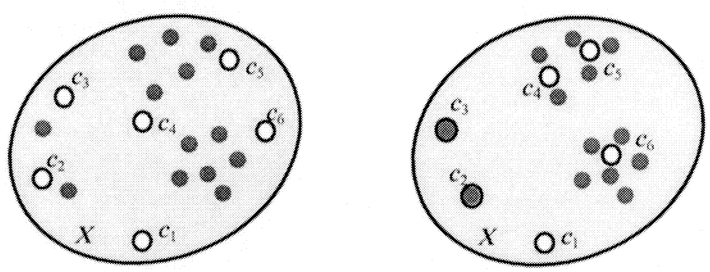


Figure 12: Training of Kohonen network: left - untrained network; right - trained network.

In the field of space $\boldsymbol{X}$ in which nucleus are far from all training vectors, neuron c_1 will never win, and its weights will not be corrected at training. In fields with great number of input vectors, density of nucleus is small, and unlike objects will activate the same neuron c_6. These lacks are connected with initial assignment to neuron weights uniformly distributed random numbers. The problem is solved by allocation of nucleus according to density of input vectors. But distribution of input vectors often is unknown. In this case is used *method of convex combination*, which allows to distribute nucleus of classes (vectors of weights) according to density of input vectors in space $\boldsymbol{X}$. This method is realized as follows:

- assignment of identical initial value to all weights:

$$\boldsymbol{w}_{ij} = \frac{1}{\sqrt{\dim \boldsymbol{X}}}, \tag{8}$$

where dim $\boldsymbol{X}$ – diameter of a class.

In result, according to the requirement нормировки, vectors of weights receive the length equal to unity.

- creation of training set $\{\boldsymbol{x}_n\}$ and carrying out of training with vectors

$$\beta(t)\boldsymbol{x}_n + \frac{1-\beta(t)}{\sqrt{\dim \boldsymbol{X}}}, \tag{9}$$

where t – time of training;
β (t) – monotonously growing function in an interval [0,1.]

In the beginning of training β(t)=0 and all vectors of weights and training set have the same value (figure 13.1). In process of training β (t) grows, training vectors diverge from a point with coordinates $1/\sqrt{\dim \boldsymbol{X}}$ and come nearer to its true values $\boldsymbol{x}_n$ (figure 13.2) which are reached at β(t)=1. Each vector of weights grasps group or one training vector and traces it in process of growth of β. In result remains no untrained neuron in network, and density of vectors of weights corresponds to density of vectors of training sets (figure 13.3). Process of increase β demands many iterations that leads to increase of training time.

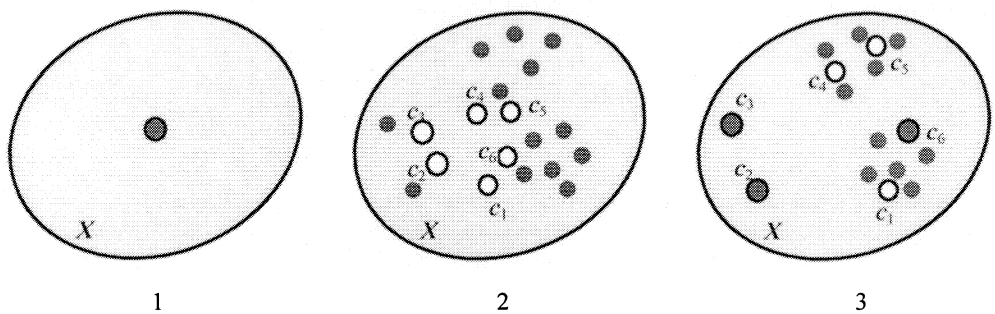


Figure 13: Training by a method of convex combination.

From the point of view of restoration of defective images, application of method of convex combinations:
- does not need obligatory presence of nucleus of a class (the true image). The nucleus of a class is formed during training of network on basis of available defective images of object, and can be specified when appear new defective images;
- at presence of only defective images of object it is possible with high degree of accuracy to specify a location of nucleus of class – the true image of object, by calculation of center of gravity of a class as, when Evklid measures of affinity is chosen, nucleus of class, minimizing sum of measures of affinity, coincides with the center of gravity of objects:

$$\boldsymbol{c}_q = \frac{1}{N_q} \sum_{n:q(n)=q} \boldsymbol{x}_n, \qquad (10)$$

where N_q – number of objects $\boldsymbol{x}_n$ in a class q.
The structure of clasters can be shown by visualization of distances between vectors of neuron weights. Distances between such vector and it's nearest neighbours in a grid are the elements of a matrix of distances (U-Matrix). Values of distances define colors by which the unit will be painted. In gradation of grey colour, the more is the distance, the darker painted unit is. For a colour palette the distance is defined according to a colour scale (figure 14). On presented Kohonen map two classes of objects are determined. By black points are marked vectors of defective images used at training. Empty cells designate vectors of all possible defective images belonging to given classes of objects. Thus:
- the analysis of Kohonen map allows to specify priori what defects of image can be restored.

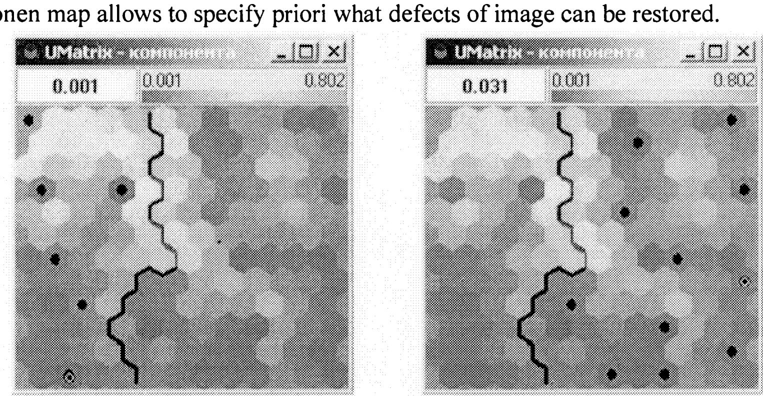


Figure 14: Coloration of Kohonen map and display of vectors of two clasters.

2.3 Functioning of bidirectional associative memory

Training of network is made using training set consisting of pairs of vectors $\boldsymbol{x}_{\mathrm{n}}$ and y_{n}, and is realized by calculation of weight matrix as sums of products of all vector pairs of training set according to (5):
For restoration of associated object $\boldsymbol{x}_{\mathrm{n}}$, this object is set momentary on output of 2-nd layer, processed by a matrix of weights $\boldsymbol{w}$, then is deleted and network is transformed in other state, elaborating vector $y_{\mathrm{n}r}$ on output of 3-rd layer:

$$y_{\mathrm{n}r} = f(\boldsymbol{x}_{\mathrm{n}}\boldsymbol{w}), \quad r = 1, 2, ..., \mathrm{R}, \tag{11}$$

where r – number of a relaxation of network;
R – number of relaxations up to achievement of global minimum of function of energy of network.

The vector $y_{\mathrm{n}r}$, reacting through the transposed matrix $\boldsymbol{w}^{\mathrm{T}}$, creates a new vector $\boldsymbol{x}_{\mathrm{n}r}$ on output of 2-nd layer:

$$\boldsymbol{x}_{\mathrm{n}r} = f(y_{\mathrm{n}r}\boldsymbol{w}^{\mathrm{T}}), \tag{12}$$

Then this process is repeated for new vectors $\boldsymbol{x}_{\mathrm{n}r}$ and $y_{\mathrm{n}r}$:

$$\begin{cases} y_{\mathrm{n}r} = f(\boldsymbol{x}_{\mathrm{n}r}\boldsymbol{w}), \\ \boldsymbol{x}_{\mathrm{n}r} = f(y_{\mathrm{n}r}\boldsymbol{w}^{\mathrm{T}}). \end{cases}$$

Each similar cycle (relaxation) specifies values of output vectors of 2-nd and 3-rd layers and approaches them to pair of associated vectors $\boldsymbol{x}_{\mathrm{n}}$ and y_{n}. Process proceeds up to achievement of point of stability of network at which processing of current signals of network does not cause their change. This condition of network $y_{\mathrm{nR}}=y_{\mathrm{n}}$ will corresponds to determined association

$$y_{\mathrm{n}} = \boldsymbol{A}_{\mathrm{nq}}(\boldsymbol{x}_{\mathrm{n}}).$$

Bidirectional associative memory has restrictions on a maximum quantity of associations, which it is capable to reproduce precisely. At excess of a limit, network may develop incorrect output signal, reproducing unforeseen associations. If as activational function is used function in which threshold value $\boldsymbol{T}$ gets out for everyone neuron, then such associative memory is *not homogeneous*. In this case bidirectional associative memory can have up to 2^{M} stable states.

If Q vectors - samples are chosen as random vectors, and

$$\mathrm{Q} < 0{,}68\frac{\mathrm{M}^2}{(log_2\mathrm{M}+4)^2}, \tag{13}$$

and if each vector has $log_2\mathrm{M} + 4$ component, equal "+1", and the rest, equal "-1" then it is possible to receive not homogeneous network having 98 % of vectors - samples as stable states. For example, if M=1024, then Q <3637, that is substantial improvement in comparison with homogeneous networks, but these are much less than 2^{1024} possible states.

Despite of limited capacity of bidirectional associative memory, its false answers and some unpredictability of behaviour, it has many advantages: fast convergence of processes of training and restoration of information.

2.4 Functioning of arbitrator

For rational application of neural networks trained with the teacher and without the teacher, as well as associative memory for restoration of images, unification of them is possible, on basis of rules of theory of collective decisions. Let in some situation $\boldsymbol{X}$, by means of algorithm $\boldsymbol{A}$, decision $\boldsymbol{S}$ is made:

$$\boldsymbol{S} = \boldsymbol{A}(\boldsymbol{X}), \tag{14}$$

Let's assume that exists L various algorithms of solving a problem, i.e.

$$S_l = A_l(\boldsymbol{X}), \quad l=1, 2, ..., L,$$

where S_l – decision received by algorithm A_l.

Set of algorithms $\{\boldsymbol{A}\}=\{A_1, A_2, ..., A_L\}$ forms collective of algorithms of decision of problem (*collective of deciding rules*) if for set of decisions S_l in any situation $\boldsymbol{X}$ deciding rule $\boldsymbol{F}$ is determined:

$$\boldsymbol{S} = \boldsymbol{F}(S_1, S_2, ..., S_{\mathrm{L}}, \boldsymbol{X}). \tag{15}$$

Algorithms A_l are members of collective: A_1 – algorithm of restoration of image by multilayered network of direct distribution; A_2 – algorithm of restoration of image by Kohonen network; A_3 – algorithm of restoration of image by associative memory.
In a problem of restoration of image, $\boldsymbol{X}=\boldsymbol{x}_{\mathrm{n}}$ – the description of image of object,

S_l – the decision of *l-th* member of collective, and $\boldsymbol{S}$ – the collective decision. Individual and collective decisions S_l and $\boldsymbol{S}$ form class to which the restored image belongs.

Function $\boldsymbol{F}$ defines a way of generalization of individual decisions in the decision of collective $\boldsymbol{S}$.
The most interesting are the collectives of algorithms in which there is a dependence of weight of each deciding rule A_l on presence/absence of true image and/or features of restored image $\boldsymbol{x}_{\mathrm{n}}$. For example, in the first case it is necessary to choose between neural network trained without the teacher and associative memory, which do not require standard images.
The weight of deciding rule A_l can be defined as:

$$\chi_{A_l}(\boldsymbol{x}_{\mathrm{n}}) = \begin{cases} 1, \text{ если } \boldsymbol{x}_{\mathrm{n}} \in B_l \\ 0, \text{ если } \boldsymbol{x}_{\mathrm{n}} \notin B_l \end{cases}, \tag{16}$$

where B_l – field of competence of deciding rule A_l.

Weights of deciding rules are chosen so, that

$$\sum_{l=1}^{L} \chi_{A_l}(\boldsymbol{x}_{\mathrm{n}}) = 1 \tag{17}$$

for all possible values $\boldsymbol{x}_{\mathrm{n}}$.

Ratio (16) means, that the decision of collective is defined by that rule A_l, to which competence belongs the image $\boldsymbol{x}_{\mathrm{n}}$ and presence/absence of true image. Such approach represents two-level procedure. At the first level is defined the belonging of image $\boldsymbol{x}_{\mathrm{n}}$ and presence/absence of true image to competence of one of rules A_1, A_2, ..., A_L, and on the second – is realized deciding rule which competence is maximal. The decision of this rule is identified with the decision of all collective.

The basic stage in such organization of collective decision is definition of areas of competence. They can be searched using probabilistic properties of rules of collective, or it is possible to apply a hypothesis of compactness and to consider, that to close rules there should correspond compact areas. During training first are allocated compact sets and corresponding them areas, and then in each of these areas the deciding rule gets out. The decision of such rule, which is true in certain area, is identified with the decision of all collective.

Deciding the majority of problems person mainly uses memory, from which takes the decision. Information gets In memory during long-term training. In result many problems are stored in head of a person and each problem has one or several decisions. If for a new problem in memory of a person is no required decision then the person will use the intelligence for development of new algorithm. In intellectual system of restoration of image, having information about object, search should be conducted in associative memory. At absence of information (for the first time meeting unfamiliar object), neural networks with training without the teacher are applied. Thus, the level of competence of associative memory is higher, than that of neural network.

The intellectual system should independently distribute intellectual resources for decision of problems of restoration of images.

3. CONCLUSIONS

Despite of comprehensible results in decision of a big class of problems the submitted intellectual system of restoration of images realizes *the mechanistic* approach, which don't take into account sense, semantics, character, intonations, a context of restored image. Each individual distinguishes close, clear and accessible to him nuances in the image. For example, observing image of a picture, various people define differently for themselves the contents, events, stages, development of a plot, transfer of mood, color scale, structural line, rhythmic, proportions, prospect, etc. The knowledge of historical, cultural, geographical, household, technical, biological and other aspects, allows to comprehend the defective image of object better.

Naturally, at intelligent perception of defective image, it is easier to find its missing elements, and to restore the image. Process of restoration of defective images by systems of artificial intelligence is closer to functioning of an alive brain and is more effective at attempt of using listed above factors. The system of artificial intelligence should "understand and analyze the image, allow it to give not only informational estimations, but also semantic ones, and use them for adjustment of parameters of neural network. Thus it is necessary to solve not only a problem of restoration of image of object but also its semantic recognition.

REFERENCES

1. A. Zmitrovici. Intellectual information systems / HTOOO “Teтra Sistems”, Minsk, 1997.

2. R. Kallan. The basic concepts of neural networks. The publishing house "Williams", 2001.

3. I. Cornea, I. Mardare. Bidirectional associative memory in problems of restoration of images. 3st International Conference "Informational technologies – 2003 ", Apr. 7-11, 2003, Chişinău, (p.233).

4. I. Mardare. Restoration of the image with use of a neural network. Acta Academica 2002, pp.246-255.

5. F. Uosserman. Neurokomputing technics: the theory and practice. – M.: World, 1992.

6. I. Cornea, I. Mardare. Restoration of Images with Application of Neural Networks. Proceedings of 2002 IEEE-TTTC. International Conference on Automation, Quality and Testing, Robotics. May 23-25, 2002, Cluj-Napoca, Romania, pp.95-98.

7. I. Cornea, I. Mardare. Methods of Artificial Intellect in Images Restoration. CSCS-14. 14th International Conference on Control Systems and Computer Science. 2-5 July, 2003, Bucharest, Romania, pp.141-143.

8. I. Mardare. Images Restoration by means of Associative Memory. CSCS-14. 14th International Conference on Control Systems and Computer Science. 2-5 July, 2003, Bucharest, Romania, pp.144-148.