Secondary learning in the assembly neural network

Alexander Goltsev*

Department of Neural Information Processing Systems, V.M. Glushkov Cybernetics Center of Ukrainian Academy of Sciences, Pr. Glushkova 40, Kiev 03680, Ukraine

Received 14 January 2004

Abstract

A neural network with assembly organization is described. The network is artificially partitioned into several sub-networks according to the number of classes that the network has to recognize. The features extracted from input data are encoded into activation of certain patterns of neurons in the sub-networks. During a process of primary learning, Hebb’s neural assemblies are formed in the sub-networks by means of modification of connections’ weights. A procedure of secondary learning, which is named as that of differentiation, is described. The procedure is intended to improve a recognition accuracy of the network by means of additional modification of connections’ weights between the neurons of the same sub-networks. A computer simulation of the network is performed. The differentiation process is studied in a set of experiments on a character recognition task using two types of objects: Ukrainian letters and Arabic numerals of modified US National Institute of Standards and Technology (MNIST) database.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Neurons; Neural assemblies; Learning; Features; Connection weights; Procedure of differentiation; Recognition algorithms

*Tel.: +380-44-497-3775; fax: +380-44-266-1570
E-mail address: agoltsev@adg.kiev.ua (A. Goltsev).
1. Introduction

Numerous neurophysiological studies may be interpreted in favor of Hebb’s hypothesis about structural and functional organization of biological neural networks in the form of neural assemblies [22,33,34]. According to this hypothesis, a neural assembly is a group of nerve cells connected with numerous mutual excitatory high conductive connections. The enhancement of conductivity of the intra-assembly connections is the essence of the learning process, which takes place in the neural network. The aim of the neurons’ binding into a neural assembly is joint, simultaneous, and mutually supporting activation of all neurons of the assembly as a functional unit of the network. An important peculiarity of Hebb’s hypothesis is the assumption that only a small number of neurons constitute a neural assembly in comparison to the total number of neurons in the network.

The present paper falls into the field of artificial neural networks. Although the artificial neural networks are intensively studied during many years, most of them may be classified as different variations of the perceptron (e.g., [41,44,13,31,32,8,28,29,12]). The present paper considers another type of artificial neural networks—assembly neural networks that share some common properties with associative memories (e.g., [47,48,37,25,1,46,45,21]).

Properties of neural networks with various versions of the assembly organization are considered in rather different papers such as [6,7,37,38,5,24,26,23,2,46,35].

The present paper is a continuation of the work described in [15,17–20]. The aim of the paper is to study a special procedure, which is intended to improve a recognition ability of the assembly neural network with analog connections. An early version of this procedure is presented in [17,19] under the name of a differentiation procedure. The procedure is performed on completion of a process of primary learning in the network. That is why this procedure is interpreted in the paper as secondary learning. The process of differentiation is studied in a set of experiments with the computer model of the assembly neural network on a task of recognition of separate handwritten characters of two types: letters of Ukrainian alphabet and Arabic numerals of modified US National Institute of Standards and Technology (MNIST) database [36]. The experiments have shown that the differentiation procedure is an effective tool for enhancement of recognition ability of the assembly neural network, which makes it into rather efficient recognition device. However, it is worth to note that the purpose of the work is not to design a specialized device for recognition of characters; the purpose is an experimental study of the differentiation process itself on the task, which is of some practical use. The differentiation procedure described in the paper could be used in some other recognition devices.

2. A description of the model

The assembly neural network is intended to solve the following task. Let there exist several classes of objects that the network has to recognize. For example, they may be texture classes, character classes, etc. Some numbers of training samples of
each class are given to learn the network. Thus, the learning process taking place in the assembly neural network is of a supervised learning type. In a better case the training samples of each class represent all typical peculiarities of this class. The network task is to classify test samples of the same classes.

In the present paper, all consideration is conducted in terms of features. In many learning machines (such as multilayer perceptrons, for instance) the features arise from the learning process itself and their formation is performed in parallel with the machine learning. Recognition techniques of other type require previously determined algorithms of feature extraction. The assembly neural networks fall into the latter category. Functioning of the assembly neural networks implies that some features are extracted from every input image according to a pre-determined set of algorithms. The given set of algorithms may produce more or less adequate descriptions of the recognized objects. Some discussion on this topic will be continued in Section 5.

In spite of the fact that similar descriptions of learning and recognition algorithms of the assembly neural networks have been presented in the literature (e.g., [18, 20]), it is necessary to give some information about the network’s functioning in the paper in order not to refer a reader to other publications for the details that are required for explanation of the algorithms described below.

2.1. An outline of the assembly neural network

Every sample of input data presented to the network is processed by means of a pre-determined set of algorithms which result in extraction of some set of features describing the input sample. Each network’s neuron is a representative of a certain feature, which may be extracted from input data. In other words, some neural encoding of an input image is produced through pre-processing.

A neural assembly is a definite group of neurons; all neurons of this group are bound into the assembly by means of mutual excitatory connections that connect all of them between one another. The neural assemblies are formed in a process of primary learning. A neural encoding of the set of features extracted from every training sample is memorized in the form of the corresponding neural assembly. All samples of the available training set are successively presented to the network to be memorized as neural assemblies in it.

The assembly neural network has the following pre-organization: it is artificially divided into several sub-networks. Every sub-network represents one class, which the network has to recognize. All sub-networks consist of the same number of neurons; this number includes all neurons that are necessary for neural encoding of all features extracted from input data during learning. Also, each sub-network includes connections between all its neurons.

Similar pre-organization of neural networks is considered by other authors (e.g., [14, 42, 11]).

During the process of primary learning, each sub-network is trained with samples of the same class so that the neural assemblies representing training samples of this class are formed in the same sub-network. Since the neural assemblies overlap
between one another, they fuse into a bound assembly family in which every neural assembly becomes a part of an integral description of the corresponding class.

The process of secondary learning (differentiation) begins after all neural assemblies become formed in the network. An essence of this process consists in adaptive modification of connection weights within all sub-networks; the aim is to create more adequate descriptions of classes in them.

2.2. Functioning of the assembly neural network

Let there exist \( M \) classes that have to be recognized by the network. Therefore, the assembly neural network has to be partitioned into \( M \) sub-networks. An index \( m \) labels the different sub-networks in the consideration below, \( m = 1, 2, 3, \ldots, M \).

Let every sub-network consist of \( N \) neurons representing all features that may be extracted from input data. Let us enumerate all neurons of each sub-network separately and in the same order so that every neuron of every sub-network, which represents the same feature value, has the same number. An index \( i \) labels the corresponding neurons of different sub-networks in the consideration below, \( i = 1, 2, 3, \ldots, N \).

Some set of features is extracted from each sample of input data presented to the assembly neural network as for learning as for recognition. A set of feature values, which is extracted from an input sample, is transformed into activation of a pattern of initial neural activity in the sub-network under consideration. Let us represent the pattern of initial neural activity of the \( m \)th sub-network, which is generated from the \( x \)th sample, by one-valued components of a binary vector \( G^m(x) \) of \( N \) components. Fig. 6B shows an example of the patterns of initial neural activity used in the experiments described below.

An output characteristic of every network neuron is linear and has no threshold so that any value of an input excitation of a neuron is transmitted to its output without any distortion or delay. A state of each neuron is characterized by its output activity (nonnegative integer). Thus, neurons of the considered neural network are, actually, simple linear summators. An integer-valued vector \( E^m \) of \( N \) components is used to represent activities of all neurons of the \( m \)th sub-network.

Let us introduce a separate integer-valued two-dimensional connection matrix \( W \) of \( N \times N \) components for each sub-network. Initially, all weights of all connections have zero values in the network.

A process of primary learning of the assembly neural network is executed as follows. Let all samples of the available training set of the \( m \)th class be presented to the network in turn for this process. Let us divide the primary learning process into a number of learning steps. The aim is to express explicitly that the entire learning process consists of small elementary portions that are multiply repeated during the process. Each presentation of a new training sample, which results in memorization of the corresponding neural encoding of this sample in the network, is the learning step. Let us denote the \( m \)th connection matrix which is formed after \( k \) previous learning steps by the notation \( W^m(k) \). Let the \( m \),th training sample be presented for the next \( (k + 1) \)th learning step. The pattern of initial neural activity \( G^m(m) \) is set in
the \( m \)th sub-network; this pattern is a neural encoding of the feature set extracted from the \( m \times \)th training sample. Hebb’s learning rule is used. This means that the weights of all connections between all neurons of the activated pattern increase by the value \( \Delta W \). After the \((k + 1)\)th learning step, the connection matrix of the \( m \)th sub-network is calculated by the equation

\[
W_{ij}^m(k + 1) = W_{ij}^m(k) + \Delta W(G_i(mx) \land G_j(mx)),
\]

(1)

where \( \land \) is conjunction, \( i = 1, 2, 3, \ldots, N \), \( j = 1, 2, 3, \ldots, N \).

Let us note that formula (1) describes modification of connections between all activated neurons in the sub-network including those connecting output and input of the same neuron.

Following Hebb’s terminology, every learning step is interpreted as binding the neurons of the activated pattern into a neural assembly. At the same time, the neural assembly bound according to formula (1) becomes a constituent of the bound connection structure formed in the \( m \)th sub-network in the learning process.

The same learning procedure is successively repeated for all samples of the \( m \)th training set. This series of learning steps results in the formation of some specific connection structure in the \( m \)th sub-network. Then, analogous series of learning procedures are carried out in all other sub-networks and the primary learning stage is completed.

A recognition process also begins with the same procedure of feature extraction from a test sample; let it be the \( x \)th sample. Unlike the case of the primary learning stage, the pattern of initial neural activity is set in all sub-networks in parallel to create equal opportunities for all sub-networks to win a competition between them. This neural activity spreads (only once) through the formed connection structure within each sub-network and produces some secondary distribution of activities of all network neurons. The vector of secondary activity \( E^m(x) \) for the \( m \)th sub-network is calculated by the equation

\[
E^m_i(x) = G_i(x) \sum_{j=1}^{N} G_j(x)W_{ij}^m.
\]

(2)

As follows from Eq. (2), not all neurons of the network take part in the recognition process. Calculation of the secondary activity is carried out only for those neurons that constitute the pattern of initial neural activity in the network. In order to express formally this peculiarity of the recognition algorithm, the binary factor \( G_i(x) \) is introduced into formula (2); it zeros the outputs of all those network neurons that do not belong to the pattern of initial neural activity.

The network uses a set of \( R \)-neurons for the recognition process; each \( R^m \)-neuron is a representative of the \( m \)th sub-network; let us remind that the index \( m \) labels the different sub-networks, \( m = 1, 2, 3, \ldots, M \). After the initial neural activity spreads through the connection structures within all sub-networks, every \( R^m \)-neuron summarizes secondary activities of all those neurons that constituted the pattern of initial neural activity in the \( m \)th sub-network and represents overall level of activity of these neurons. An integer-valued vector \( R(x) \) of \( M \) components represents
activity of all \( R \)-neurons. The \( m \)th component of this vector is calculated according to the equation

\[
\mathbf{R}^m(x) = \sum_{i=1}^{N} E_i^m(x). \tag{3}
\]

This formula may be rewritten as follows using Eq. (2) and taking into account that the matrix \( \mathbf{W}^m \) is square and symmetrical

\[
\mathbf{R}^m(x) = 2 \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{G}_i(x) \mathbf{G}_j(x) \mathbf{W}_{i,j}^m. \tag{4}
\]

In order to classify the input test sample, the \( R \)-neuron with maximum activity is found. Finding the maximum value \( L \) among \( M \) outputs of \( R \) neurons may be expressed by the formula

\[
L(x) = \max_{m=1}^{M} \mathbf{R}^m(x). \tag{5}
\]

The \( R \)-neuron, which is determined by means of this “winner-take-all” procedure, defines the sub-network winner and, consequently, the class of the test sample presented for recognition.

The considered recognition algorithm can work as in biological neural networks, as in computer models of the assembly neural networks. The following simple version of this recognition algorithm is realized in the computer model of the assembly neural network, which was used in the experiments described below. The weights of all connections between those neurons that constitute the pattern of initial neural activity are simply summarized separately in each sub-network according to Eq. (4). The sub-network with the maximum sum of these weights becomes the winner.

2.3. Generalization of features in the assembly neural networks

An important peculiarity of the presented assembly neural network is that the network is partitioned into several sub-networks each of which represents one recognized class. The multiple duplication of neurons in the network partitioned into several sub-networks may seem wasteful. Therefore, it is necessary to consider some grounds in favor of the partition. The advantages of the assembly neural network partitioned into separate sub-networks are simplicity, accuracy, and clarity of its functioning. Owing to the partition, the network acquires two important properties. The first is an ability of the network to form complicated, non-linearly separable decision regions in the feature space. The other property is the network’s capability to generalize the description of each class separately within the corresponding sub-network.

Let us consider the process of primary learning of the assembly neural network from the point of view of formation of the classes’ descriptions. During the process of binding the neural assemblies in a certain sub-network, the assemblies have
intersections between one another. Some bound neural structure (an assembly family) is generated in the sub-network due to these intersections. Each such structure has a specific pattern of connections; these patterns are rather different in different sub-networks. The family of intersecting neural assemblies becomes a description of the corresponding class in each sub-network. Since the assembly families describing different classes are positioned in different sub-networks, these families have no common neurons at all in spite of their large intersections in feature values.

During saturation of a certain sub-network with a large number of neural assemblies, a degree of adequate description of the corresponding class increases progressively. The more the number of training samples of a certain class used for training, the more the density of connections and the connection weights in definite regions of the sub-network’s connection matrix.

The process of formation of the descriptions of classes (assembly families) in all sub-networks is interpreted as phenomenon of generalization of features. First of all, the assembly neural network generalizes those features and their combinations that occur most frequently in the training sets of the corresponding classes.

It is possible to give the following pithy interpretation to the class’s description, which is formed in each sub-network during the primary learning process. Every feature being extracted from an input image is represented with a certain set of neurons in the corresponding sub-network. Let us emphasize that this set of neurons has unchangeable location in the sub-network. Each neural assembly is bound by means of intra-assembly connections that connect all pairs of its neurons. Therefore, every intra-assembly connection represents a definite spatial relation between two features extracted from the training sample. The weight of this connection represents how often this spatial relation occurs in the training set of the corresponding class. Then, the whole bound neural structure formed in each sub-network is an assemblage of pair-wise spatial relations of all features extracted from all samples of the training set of the corresponding class.

Let us consider, concisely, an assembly neural network, which is not partitioned into separate sub-networks; let us name such a network as unpartitioned. The unpartitioned assembly neural network is equivalent in number of neurons to one sub-network of the partitioned one. In the unpartitioned network, the neural assemblies represent training samples of all classes that take part in the process of primary learning. They also have intersections between one another. Due to these intersections, the unpartitioned network comprises such connections that reflect irrelevant relations between those features and their combinations that belong to different classes. In other words, the assembly families of all classes are superimposed on one another fusing into a bound neural structure with almost a homogeneous structure. This means that the network loses a significant part of its recognition capability. Thus, the process of generalization of features, which favors the partitioned network, affects adversely in case of the unpartitioned one.

Experimental study of the generalization phenomenon and comparison between the partitioned and unpartitioned assembly neural networks with binary connections are described in [20].
2.4. A differentiation procedure in the assembly neural network

A process of secondary learning of the assembly neural network is named as that of differentiation. The aim of this process is to increase the recognition capability of the network. The prototype of the present differentiation procedure is described in [17,19].

In the process of differentiation, the assembly neural network model repeatedly considers all available training samples in turn. During each step of the differentiation procedure, the model tries to recognize the current training sample and, then, modifies or does not modify weights of some network’s connections in dependence on the recognition result. A concept of epoch, which is often used in the field of learning algorithms, is useful for further explanation of the differentiation algorithm. The epoch is an entire series of successive considerations of all samples of the training set by the model.

According to the description of the recognition algorithm (see Section 2.2), $R$-neurons serve for representation of all $M$ sub-networks in the competition process between them. The $R$-neuron with maximum activity determines the sub-network winner. Let the $m_i$th training sample of the $m$th class be currently considered by the model. The model tries to recognize it. Let the recognition result be such that the $R^{m_i}$-neuron has the maximum activity among all $R$-neurons. This means that the network recognizes the input training sample rightly by classifying it to the $m$th class. If the activity of the right $R^{m_i}$-neuron exceeds activities of all other $R$-neurons more than by a pre-defined value $\Omega$, then the model proceeds to consider the next training sample. If the activity of the right $R^{m_i}$-neuron exceeds the activity of some wrong $R^v$-neuron less than by the pre-defined value $\Omega$, the following two procedures are fulfilled.

The first procedure is as follows. All weights of connections between the neurons of the neural assembly, which represents the $m_i$th training sample, are increased by the value $\Delta W$ in the right $m$th sub-network, i.e., in the very sub-network in which this neural assembly was initially formed during the primary learning stage. This procedure is described by Eq. (1).

The second procedure consists of the following. All weights of connections between the neurons that represent the pattern of initial neural activity of this training sample in the wrong $v$th sub-network, are decreased by the same value $\Delta W$. This procedure is expressed by the formula

$$W_{ij}(k + 1) = W_{ij}(k) - \Delta W(G_{j}(m_i) \land G_{j}(m_i)),$$

(6)

These two procedures increase the competitiveness of the $m$th right sub-network and decrease the competitiveness of the $v$th wrong sub-network during competition between them in the future processes of recognition of the same and other samples.

Then, the network tries to recognize the same $m_i$th training sample again in order to check whether the done modification of the connection weights is enough for the right recognition of the considered training sample under the above condition. If the condition is not still satisfied, the described two procedures are performed again. Such a modification of the connection weights in the sub-networks repeats until the activity of the right $R^{m_i}$-neuron exceeds activities of all other $R$-neurons by more than
The parameter \( \Omega \) may be defined as some external influence, which counteracts the right classification of the recognized training sample by the network. Evidently, the value of the counteraction \( \Omega \) may vary. In the series of experiments presented below, this value was gradually increased which led to a significant increment of the recognition capability of the assembly neural network after accomplishment of all stages of the differentiation process. In these experiments, the \( \Omega \) value was measured in percentage with respect to the activity of the right \( R \)-neuron corresponding to the class to which the recognized training sample should be classified.

It follows from the above description and Eq. (6) that the differentiation procedure can generate connections with negative weights between neurons of the same sub-networks. The connections with negative weights are usually named as inhibitory. Formation of a large number of inhibitory connections was recorded in the experiments described below.

3. Experiments with separate handwritten letters

The differentiation process in the assembly neural network is experimentally studied on a task of recognition of separate handwritten letters of Ukrainian alphabet. To create a database for experiments with the letters, different persons wrote separate letters of Ukrainian alphabet on a paper using usual ball pens. These drawings were scanned and transformed into binary images of 64 × 64 pixels. Examples of such images are shown in Fig. 1.

A simple set of features is used for description of the letter shapes in the experiments. This set is designed mainly for research purposes and hardly could be used for practical applications because of its low effectiveness. This is one cause why this feature set is not considered in the present paper. The other cause is that the feature set is presented in [18].

In experiments with the letters, the features extracted from the letters are represented in the assembly neural network according to the method of “float” coding. This method of coding is sufficiently described in the literature (e.g., [39,27,17,18,20,10]). That is why there is no consideration of this method in the paper. It is, only, worth to make the following note concerning the “float” coding method. In the experiments with the letters, every sub-network consists of 32 neural columns of 32 neurons each. The feature set extracted from every input letter is encoded into activation of one neural float of five neurons each in each of 32 neural columns. Owing to this, all neural assemblies formed in the network during the primary learning process consist of the same number of neurons.

The experiments consisted of the following. The assembly neural network was divided into 32 sub-networks according to the number of letters of Ukrainian alphabet. First of all, the process of primary learning was carried out on all available
training samples of Ukrainian letters, which resulted in formation of the assembly family structures in all 32 sub-networks. In all stages of the experiments, the same training and test sets were used. The training set consisted of 80 samples of every letter of Ukrainian alphabet (2560 samples, a total). The network performance was measured using the test set which consisted of 20 samples of each letter (640 samples, a total). The samples of the training set as well as those of the test set were chosen randomly from the available database without any overlap between them.

After the primary learning stage was completed, the differentiation procedure began according to the description of Section 2.4. The first step of the differentiation procedure was executed for $\Omega = 0$. After this procedure was finished, the network acquired an ability to recognize all samples of the training set with no errors. The network performance was tested on the test set. The second step of the differentiation process was done with some increment of the parameter $\Omega$. New measurement of the recognition accuracy was taken using the same test set. The subsequent steps were carried out analogously. Let us remind that after each step of the differentiation procedure, the network became able to recognize all samples of the training set with no errors under condition that every recognition process was counteracted with the current parameter $\Omega$.

The numbers of epochs, i.e., the numbers of the full cycles of consideration of all samples of the training set, that were required for successful completion of the current step of the differentiation process were recorded in the experiments. The experimental results are expressed in graphical form in Figs. 2 and 3.

Fig. 2 shows a curve of error percentage committed by the assembly neural network during recognition of the test set of handwritten letters of Ukrainian
alphabet depending on the parameter $\Omega$. In other words, the curve in Fig. 2 depicts a dependence of the network’s recognition ability upon a value of counteraction, which the network has overcome by the corresponding stage of the differentiation process. As it is seen in Fig. 2, the differentiation process is rather an effective procedure for increment of the recognition ability of the assembly neural network with analog connections.

A curve in Fig. 3 shows how many times the assembly neural network has to consider all samples of the training set in order to complete successfully the current step of the differentiation process. Fig. 3 demonstrates that the differentiation process is a rather computationally expansive procedure, which takes a great number of epochs. It is seen in Fig. 3 that steepness of the curve rises together with increment of the parameter $\Omega$. Thus, the differentiation process is naturally limited with a steep increase of the computational cost of its each subsequent step. Let us note that in contrast to long differentiation process the subsequent recognition process is very quick.

It follows from the curve of Fig. 2 that the differentiation process increases the recognition ability of the network from 13.4% of errors to 8.4%. Let us take as a benchmark for the recognition ability such a percentage that is achieved by the network after accomplishment of the differentiation procedure for $\Omega = 0$. Then in
the experiments presented in Fig. 2, the differentiation procedure decreases the network’s percentage of errors by 37%.

The curve of Fig. 2 may be transformed into another form as it is depicted in Fig. 4. The curve in Fig. 4 shows the dynamics of change of the network’s recognition ability in relative units in dependence on a progress of the differentiation process.

As it is mentioned above, Eq. (6) allows formation of inhibitory connections between neurons of the same sub-networks during the differentiation procedure. In the experiments described above, recording of the inhibitory connections was performed along with measurement of their weights. The experiments showed that a large amount of inhibitory connections were created in the network and it turned out that the maximal inhibitory weight was comparable with the maximal excitatory weight in the network.

Every inhibitory connection, which has grown between two neurons of the same sub-network, is expression of a negative correlation between the corresponding two features extracted from the training samples of a certain class. This may be interpreted so that the differentiation procedure makes the pairs of neurons connected with inhibitory connections some new features indicating non-belonging of these pair-wise feature combinations to the class.

The experimental results presented above certainly demonstrate that the differentiation procedure is an effective tool for increment of the recognition ability of the assembly neural network with analog connections.
4. Experiments with separate handwritten numerals

The conclusion made above about effectiveness of the differentiation procedure can be considered only as preliminary in particular because the database used in the experiments with the letters is rather small. This conclusion should be confirmed in more experiments with using other recognition objects with larger database and other feature extraction algorithms. Also, it would be worth to compare the recognition rate of the assembly neural network with the best recognition results. There is a database, which is widely used for testing of different recognition techniques. It is MNIST database [36]. This database is very large, it contains 60 000 handwritten Arabic numerals in its training set and 10 000 ones in the test set. Fig. 5 shows the examples 3001–3100 of the MNIST database test set. MNIST database was used in the experiments that were conducted in the same way as the experiments with the letters. Other feature extraction algorithm was applied in these experiments.

4.1. Algorithm of extraction of features for description of character shapes

Each numeral in MNIST database is an analog, black and white image of 28 × 28 pixels. To be used in the experiments described below, every image of such a type is transformed, first of all, into a binary one by means of a simple operation of thresholding. The obtained binary image is put in a binary screen of 32 × 32 pixels in
which the feature extraction procedure is executed. In this screen the numeral is drawn by one-valued pixels. At first, a center of gravity of the numeral is found and the numeral is shifted so that its center of gravity coincides with the screen center.

The pixels of the original numeral are transformed by a series of operations described below into a set of features. The extracted set of features is represented by activation of some pattern of neurons in the sub-networks. Every sub-network consists of 32 × 32 neurons arranged in a two-dimensional neural matrix which, by chance, has the same size as the binary input screen. The neurons of the neural matrix of every sub-network are divided into 32 neural columns of 32 neurons each.

The feature set which is extracted from an input numeral consists of two parts. The first half of the extracted feature set is represented in the upper 16 neurons of all neural columns of the neural matrix. The remaining lower 16 neurons of each neural column serve for representation of the second half of the extracted feature set.

The first half of the feature set is extracted by means of the following sequence of operations. The first operation is some expansion of the character within the binary screen. A prototype of this operation is presented in [16]. The operation is as follows. Every pixel of a binary two-dimensional matrix (excluding its boundary pixels) has eight contiguous pixels. The procedure of expansion is performed step by step. In each step, all eight pixels that are contiguous with every one-valued pixel of the matrix are set one-valued independently on their values at the previous step. Several steps of such an expansion transform the numeral into a swollen continuous patch, which only vaguely resembles the original numeral. And all empty inner regions of the numeral (if they are present) become filled up with one-valued pixels in the patch.

The second operation of the sequence is subtraction of the original character from the swollen patch. After accomplishment of this operation, an imprint of the original character drawn with zero-valued pixels arises within the one-valued patch.
A set of 32 beams of different orientations is drawn in the binary screen. The beams are directed from a screen boundary to the screen center. Angles between the neighboring beams are equal. The first one-valued pixel is found during going along one beam from outer boundary of the screen to the screen center (which coincides with the character’s center of gravity). A distance between the found first one-valued pixel and the center is measured. Thirty-two such distances are measured in the image in the same way. A maximum distance is found among these 32 distances. All distances considered below in the explanation of the feature extraction procedure are normalized to this maximum distance.

The described first part of the feature set takes into account only those one-valued pixels of the transformed image that are crossed with the beams. Fig. 6A illustrates the above description by example of the numeral “4”. The figure shows all one-valued pixels of the swollen patch (the former numeral “4”) that are crossed with 32 beams.

The beams are considered in turn. A distance between every one-valued pixel (crossed with a beam) and the screen center is measured. This distance is normalized to the maximum distance of the character, which has been found earlier. The normalized distance between every one-valued pixel crossed with the mth beam and the screen center is associated with one definite neuron of the mth neural column of the neural matrix. This means that this neuron gets activated in case of detection of such one-valued pixel in the image presented for the feature extraction procedure.

Fig. 6. Extraction of the feature set for description of characters’ shapes. (A) An intermediate result of transformation of the handwritten numeral “4” into the first half of the feature set. (B) Representation of all extracted features in a sub-network, which is divided into 32 neural columns of 32 neurons each (activated neurons are depicted with black color).
Location of the activated neuron within the neural column is determined with the normalized distance between the pixel and the screen center. The longer the distance, the higher the activated neuron rises in the \( m \)th neural column. However, it is worth to remind that in every sub-network only 16 upper neurons of each neural column are used for representation of the described first part of the feature set extracted according to this algorithm.

The second half of the feature set is somewhat simpler than the previous one. To perform the extraction procedure, the original binary image of a numeral is put again to the same input screen of 32 \( \times \) 32 pixels and centered in it as well. The same set of 32 beams of different orientations is drawn in the binary screen. The described second part of the feature set also takes into account only those one-valued pixels of the image that are crossed with the beams. The distance between every one-valued pixel (crossed with a beam) and the screen center is measured. This distance is normalized to the maximum distance of the numeral, which has been found earlier. The normalized distance is the feature value, which is associated with one certain neuron of the lower half of 16 neurons of the neural column, which serves for representation of all one-valued pixels crossed with the beam under consideration.

Thus, all feature values of both halves of the feature set that are extracted with the aid of the same beam are represented in the same neural column of 32 neurons. Fig. 6B illustrates this description depicting a whole neural encoding of the numeral “4”.

The pattern of activated neurons, which represents all features extracted from the input character, is some integral description of the character. This description is invariant to translation and scale changes of a character. It is robust with respect to some deformation of a character and takes into account both outer and inner contours of a character. However, the description is not invariant to rotation of a character.

4.2. Differentiation procedure in the assembly neural network

The assembly neural network was divided into 10 sub-networks according to the number of the recognized numeral classes. The experiments were performed in complete correspondence with the description of Section 3 and consisted of the following stages. First of all, the procedure of the primary network learning (see Section 2.2) was carried out in which 60 000 neural assemblies were formed in 10 sub-networks. Then, the process of differentiation began which concerned all 60 000 training samples of MNIST database. After accomplishment of every successive step of the differentiation procedure (with growth of the parameter \( \Omega \)), the network’s performance was measured using the whole test set of MNIST database.

The experimental results are expressed in graphical form in Fig. 7 in the same way as in Fig. 2.

The experiments have shown that after several steps of the differentiation procedure, the number of errors committed by the network significantly decreased up to the minimum of 207 errors (among 10 000 sample of the test set). Thus, the model’s recognition rate has achieved 97.93%. The curve of Fig. 7 shows that the recognition ability of the assembly neural network has been increased from 3.26% of
errors to 2.07%. The same result may be expressed in the relative units of percentage introduced above in Section 3 so that the differentiation procedure decreases the network’s percentage of errors by the same 37% as in the experiments with the letters.

Although the differentiation procedure in the assembly neural network takes a rather long time, the subsequent recognition process is very quick. The process of recognition of all 10000 test samples of MNIST database takes 5 min and 25 s on PC of 2000 MHz.

It is worth to mention that shapes of the curves in Figs. 2–4 and 7 slightly vary in case of variation of steps of growth of the parameter $\Omega$ during the differentiation process.

Let us note that both curves in Figs. 2 and 7 somewhat increase after passing their minimums. It is a consequence of a well-known phenomenon of overfitting (e.g., [43,30,40,9]), which takes place in any learning machine.

Shapes of the curves of Figs. 2 and 7 are similar. This fact demonstrates that the dynamics of the differentiation procedure do not alter noticeably in case of using other feature sets and/or other recognized objects. It follows from both series of experiments described in the paper that the accomplishment of the differentiation procedure in the assembly neural network leads to a significant growth of absolute and relative recognition capability of the network. And the size of growth of the network’s relative recognition capability does not change appreciably in case of using different feature sets and/or recognized objects.
5. Discussion and conclusions

The assembly neural network model considered in this paper has much in common with the models described in [17,19]. The referred models used a dynamical recognition algorithm. To distinguish both recognition algorithms, let us name the one considered in Section 2.2 as statical. The present assembly neural network has the following advantages over the networks with the dynamical recognition algorithm.

1. In the assembly neural network with the statical recognition algorithm, the recognition process is performed by means of only one spreading the initial neural activity through the network connection structure. The referred models applied a recurrent and dynamical process for the network convergence, which consisted of a series of iterations. In each iteration, some pattern of neural activity spread through the network connection structure. The simplification of the recognition algorithm makes the present assembly network much faster.

2. The present assembly neural network does not need any regulator of neural activity such as Milner’s structure of internal inhibitory connections [33,34], Braitenberg’s central extra-network regulator [6], and the system of excitation-inhibition [3,4,26,17,19]. This circumstance is very important. Owing to it, the present assembly neural network has got rid of the main complexity of the previous models and has become a very simple device.

3. The present differentiation algorithm generates a large number of inhibitory connections between neurons of the same sub-networks. Formation of such connections is considered as an important cause of the algorithm’s effectiveness. In the previous version of the differentiation procedure [17,19], it was not implied to form inhibitory connections between neurons of the same sub-networks. It was so because in the previous models any gradual influences between the network’s neurons were realized in statistically appreciable fascicles of binary connections.

Also in the assembly neural network with dynamical recognition algorithm, the differentiation process used to correct only the first step of the network’s convergence process. The subsequent steps were not touched upon the procedure of differentiation. Since in the assembly neural network with statical recognition algorithm the neural activity spreads only once through the connection structure during the process of recognition, the present differentiation algorithm corrects the whole recognition process.

Let us also consider drawbacks of the present assembly neural network model and their causes.

1. The evident disadvantage of every neural network partitioned into sub-networks is the necessity of proportionally increasing the number of neurons, when increasing the number of classes that the network has to recognize. However, due to such partitioning the network acquires the following properties. The network forms complicated, non-linearly separable decision regions in the
feature space. The network creates independent descriptions of all recognized classes in its sub-networks. The network has very simple algorithms of functioning.

2. The assembly neural network does not form a feature set during the process of learning but requires to be provided with some feature set, which should describe the recognized objects. Of course, during primary and secondary learning the network creates its own secondary features in the sub-networks, but the fact that the network cannot work without initial feature set is its important disadvantage. The recognition rate of the assembly neural network depends first of all on the given initial feature set whose quality may vary considerably. The better the feature set, the higher the network's performance. Therefore, the recognition rate, which is demonstrated by the model in the above experiments, characterizes mostly quality of the feature set presented in Section 4.1, rather than the very recognition ability of the assembly neural network itself. Thus, designing of another feature set, which should describe the character shapes more adequately, is a topical task for a future work.

3. The most visible drawback of the present model is its recognition rate of 97.93% achieved in the above experiments on MNIST database which is not too high. Indeed, this rate is still low in comparison, e.g., with the rate of 99.46% reported by Kussul and Baidyk in [29]. However, it is worth to note that the top recognition percentages of different devices that are shown on MNIST database are achieved with the aid of multiple expansion of the MNIST training set by means of multiplex artificial distortions of its numerals. Using this method considerably increases the training set size and, therefore, the recognition rate. The result of 97.93% of the present model has been achieved without any artificial expansion of the training set size; this fact should be taken into account for correct appraising of this result.

A process of generalization of features takes place separately in each sub-network of the assembly neural network. The generalization phenomenon is a consequence of intersections between the neural assemblies. The process of generalization results in creation of primary descriptions of classes in all sub-networks. The differentiation procedure, which is performed afterwards, corrects these descriptions of classes by means of tuning the connection weights within the sub-networks. The differentiation process makes the connection structure of every sub-network more specific, irregular, and containing many inhibitory connections. The differentiation procedure has to be performed in any assembly neural network with analog connections for its normal functioning. The experimental results of the paper demonstrate that the differentiation procedure is rather effective tool for increment of the recognition ability of the assembly neural network in case of any feature sets and recognized objects.

An important advantage of the assembly neural network is that it does not use any sophisticated operations, mechanisms, methods or techniques. All procedures applied in the network are very simple and do not require any complex calculations.
References

Alexander Goltsev received the M.Sc. degree (with honors) in Physics and Electronics (Bionics) from the Dnepropetrovsk State University (Dnepropetrovsk, Ukraine) in 1971. He obtained Ph.D. degree in Engineering from V.M. Glushkov Institute of Cybernetics of Ukrainian Academy of Sciences in 1976 (Kiev, Ukraine). From 1971 he works at the same Institute of Cybernetics, which is V.M. Glushkov Cybernetics Center now. Goltsev holds the position of senior researcher at the Department of Neural Information Processing Systems of International Research and Training Center of Information Technologies, which is a part of V.M. Glushkov Cybernetics Center. He temporarily worked at Linz University (Linz, Austria, 1993–1994) and at German National Research Center for Information Technology (Sankt Augustin, Germany, 2001). His research interests are neural networks, image processing, computer vision, and robotics. The whole list of publications consists of 64 items.