American Journal of Intelligent Systems 2015, 5(1): 34-41
DOI: 10.5923/j.ajis.20150501.04

Development of the Learning Process for the Neuron and Neural Network for Pattern Recognition

Ramaz Khurodze

Department of Artificial Intelligence, Georgian Technical University, Tbilisi, Georgia

Abstract The goal of our work is development of a neuron and a neural network which recognize any object without mistakes. It worked out by learning method, with totality of simple (single-layer) neurons. The work deals with the formal neuron and neural network learning process through the realizations of a set of learning sets. At the same time, the features and the feature space used for the recognition process are evaluated. The feature space for all the types and realizations is of the same dimension and binary. The learning process is carried out by means of recognition procedures; this, in case of incorrect recognition, as much as possible, draws together the changes in neuron weighting coefficients as well as the threshold (structuring etalon descriptions) with the neural process of recognition. A set of realizations for the learning clusters of each pattern is used for the learning process. The learning algorithm comprises two stages. The first stage represents structuring etalon description of its own, the second – the correction of the received description in relation to other patterns of descriptions by using the same patterns of the learning set's realizations. The correction of the results received in the recognition process is carried out by means of changing the weighting coefficients through using the award algorithm (procedure). In case of the incorrect recognition of some realization, it is presented to the neuron until we get the correct recognition through coefficients changing (error correction of mistakes) which may require neuron threshold changing.

Keywords Formal neuron, Learning process, Recognition

1. Introduction

Elaborating Neural Network has a long history of development. McCulloch and Pitts (1943) [1] developed models of neural networks based on their understanding of neurology. In the late 1940s psychologist Donald Hebb [2] created a hypothesis of learning based on the mechanism of neural plasticity that is now known as Hebbian learning. These ideas started being applied to computational models in 1948 with Turing's B-type machines Farley and Wesley A. Clark [3] (1954) first used computational machines, then called calculators, to simulate a Hebbian network at MIT. Other neural network computational machines were created by Rochester, Holland, Habit, and Duda [4] (1956). Rosenblatt (1958) [5] stirred considerable interest and activity in the field when he designed and developed the Perceptron. The Perceptron had three layers with the middle layer known as the association layer. (Three layers: I layer – output of feature space, II layer – conversion of the output; III layer decision-making). Another system was the ADALINE (ADAptive LInear Element) which was developed in 1960 by Widrow and Hoff (of Stanford University). In 1969 Minsky and Papert wrote a book in which they generalised the limitations of single layer Perceptrons to multilayered systems [7]. Grossberg's (Steve Grossberg and Gail Carpenter in 1988) influence founded a school of thought which explores resonating algorithms. They developed the ART (Adaptive Resonance Theory) networks based on biologically plausible models. Anderson and Kohonen developed associative techniques independent of each other. Klopf (A. Henry Klopf) in 1972, developed a basis for learning in artificial neurons based on a biological principle for neuronal learning called heterostasis. Werbos (Paul Werbos 1974) developed and used the back-propagation learning method, however several years passed before this approach was popularized. Amari (A. Shun-Ichi 1967) was involved with theoretical developments: he published a paper which established a mathematical theory for a learning basis (error-correction method) dealing with adaptive patern classification. While Fukushima (F. Kunihiko) developed a step wise trained multilayered neural network for interpretation of handwritten characters. The original network was published in 1975 and was called the Cognitron. The backpropagation algorithm was created by Paul Werbos [6] (1975).

The parallel distributed processing of the mid-1980s became popular under the name connectionism. The text by David E. Rumelhart and James McClelland [8] (1986) provided a full exposition on the use of connectionism in

* Corresponding author:
mariam.chkhaidze@bk.ru (Ramaz Khurodze)

Published online at http://journal.sapub.org/ajis

computers to simulate neural processes.

In the 1990s, neural networks were overtaken in popularity in machine learning by support vector machines and other, much simpler methods such as linear classifiers. Renewed interest in neural nets was sparked in the 2000s by the advent of deep learning.

Between 2009 and 2012, the recurrent neural networks and deep feedforward neural networks developed in the research group of Jürgen Schmidhuber at the Swiss AI Lab IDSIA have won eight international competitions in pattern recognition and machine learning. [9] For example, multi-dimensional long short term memory (LSTM) [10] [11] won three competitions in connected handwriting recognition at the 2009 International Conference on Document Analysis and Recognition (ICDAR), without any prior knowledge about the three different languages to be learned. Variants of the back-propagation algorithm as well as unsupervised methods by Geoff Hinton and colleagues at the University of Toronto [12] [13] can be used to train deep, highly nonlinear neural architectures similar to the 1980 Neocognitron by Kunihiko Fukushima, [14] and the "standard architecture of vision", [15] inspired by the simple and complex cells identified by David H. Hubel and Torsten Wiesel in the primary visual cortex. Deep, highly nonlinear neural architectures similar to the 1980 neocognitron by Kunihiko Fukushima [14] and the "standard architecture of vision" [15] can also be pre-trained by unsupervised methods [16] [17] of Geoff Hinton's lab at University of Toronto. A team from this lab won a 2012 contest sponsored by Merck to design software to help find molecules that might lead to new drugs. [18]. Nowadays, there are some scientists working in the field of Artificial neuroscience and making significant contribution to it: Yann LeCun, Pierre Sermanet and more. We are using the terms which are used already in the works: [19][20].

Used Symbols:

Set of patterns to be recognized:

$$\{A\} = A_1,\ A_2,..........Ai,..........A_I,\ I = Card\{A\} \quad (1.1)$$

Set of given realizations for each pattern $\{X\}$

$$\{X\} = X_1, X_2, ..., X_i, ..., X_I\,; \quad (1.2)$$

A set of realizations for each pattern consists of learning, testing and unknown realization sets; for example, for $\{X_i\}$ set of A_i pattern we will have:

$$X_i = X_{1i}, X_{2i}, ..., X_{mi}, ..., X_{Mi}\,,\ m_i = \overline{1;M_i};\ i = \overline{1;I}\,;(1.3)$$

Any type of realization represents N dimensional vector with binary components. For example, for A_i type mi realization we will have:

$$X_i^{mi} \in \{X_i\};\ X_i^{m_i} = x_{1i}^{m_i}, x_{2i}^{m_i}, ..., x_{ni}^{m_i}, ..., x_{Ni}^{m_i}\,; \quad ; \quad (1.4)$$
$$i = \overline{1;I};\ n = \overline{1;N}$$

Spatial dimension for $\mathrm{N} = \mathrm{Card}(\mathrm{x_n})$ symbols; $\forall$

$$x_n^{m_i} = \{0;1\}.$$

Due to the fact that for different patterns we may have different quantities of learning set realizations, we will have:

$$X_{mi} = \left\{x_{ni}^{m_i}\right\},\ \text{where}\ \ m_i = \overline{1; M_i}\ \ ; \quad (1.5)$$

We have each neuron weighing coefficient for each sign of the sign space, for example, we will have a set of weighting coefficients $\{\mathrm{x_n}\}$ for a $\{\mathrm{w_n}\}$ number of symbols:

$$\{W\} = w_1, w_2, ..., w_n, ..., w_N\,; \quad (1.6)$$

For each $\mathrm{A_j}$ member of A set of patterns we will correspondingly have $\mathrm{W_i}$ set:

$$W_i = \{w_{ni}\}\,; \quad (1.7)$$

For each pattern we have one neuron designated as Ne, for example we will have $\mathrm{Ne_i}$ neuron for $\mathrm{A_i}$ pattern, which will be represented by its own weighting coefficient and in the form of neuron threshold we will accordingly have sets of $\mathrm{Ne_i}$neurons and $\mathrm{Z_i}$ thresholds :

$$\{\mathrm{Ne}\} = \mathrm{Ne_1}, \mathrm{Ne_2}, \dots, \mathrm{Ne_i}, \dots, \mathrm{Ne_I}, \quad (1.8)$$

$$\{\mathrm{Z}\} = \mathrm{Z_1}, \mathrm{Z_2}, \dots, \mathrm{Z_i}, \dots, \mathrm{Z_I}; \quad (1.9)$$

For the purpose of compactly describing the recognition process through neurons in the given work original symbols have been used; these symbols reflect separate stages of recognition.

1. The introduction (arrival) of unknown realizations for recognition at the neuron entry is expressed by means of the predicate (presentation):

$=\!\!\theta\!\!\Rightarrow$ “presentation”;

The predicate “presentation” implies that an unknown realization with all its components is included into neuron

2. Predicate “is accepted” implies the result obtained through some mathematical operation:

$\Rightarrow$ “ is accepted”.

By considering one or two points, the recognition process in neurons can be presented as follows:

$$\mathrm{X} =\!\!\theta\!\!\Rightarrow \mathrm{Ne}\{\mathrm{W}\} \Rightarrow \sum \mathrm{XW} = \mathrm{net}$$
$$\Rightarrow \mathrm{F}\{\mathrm{net}\} \Rightarrow \mathrm{Z} \Rightarrow \mathrm{out} = \{0; 1\}\,; \quad (1.10)$$

$\mathrm{F(net)}$ function is termed as the activation function which is nonlinear as a rule, e.g. sigmoid or hyperbolic tangent, or some other one chosen heuristically.

$\mathrm{F(.)}$, ”net” and “out” symbols are widely spread in the corresponding literature, therefore, we don’t change them. However, let us present their mathematical description and function:

$$net = \sum\nolimits_n x_n w_n\,; \quad (1.11)$$

which represents the coordinate product of the realization and neuron weighting coefficients by means of which the modeling of charge accumulation process in soma is performed;

$$\mathrm{out} = \{0; 1\}; \qquad (1.12)$$

Through this pattern the description of the decision making process at Ne neuron exit, namely:

If $\mathrm{out} = 1$, then $X \in A_j$, if $\mathrm{out} = 0$, then $X \notin A_j$

$$A_j \in \{A\}. \qquad (1.13)$$

2. Description of the Correct and Incorrect Recognition Processes

As the goal of our work is development of a neural network which recognizes correctly (without mistakes), then at first, explain the meaning of correct or incorrect recognition processes:

The result of the recognition process is correct, if the right decision is obtained due to it: the realization to be recognized belongs to “its own” type.

The result of the recognition process is incorrect if an incorrect decision is obtained due to it: the realization to be recognized does not belong to “its own” type, or belongs to a different type. For example, if A_j type realization is presented for recognition from X_j learning set, then the recognition is correct if making a decision results in

$$X_j \in A_{j,} \qquad (2.1)$$

And the recognition is incorrect if we have:

$$X_i \notin A_i.$$

Taking into consideration the symbols obtained in the previous chapter and 1,10 patterns, the correct and incorrect recognition processes can be described as follows:

Correct recognition

$$X_i \xRightarrow{\theta} \mathrm{Ne}_i \Rightarrow \sum X_i\, W_i = \sum\nolimits_n x_{ni} w_{ni} = \mathrm{net}_i$$
$$\Rightarrow F\{\mathrm{net}_i\} \xRightarrow{\theta} Z_i \Rightarrow \mathrm{out}_i = 1 \Rightarrow X_i \in A_i; \qquad (2.2)$$

incorrect recognition:

$$X_i \xRightarrow{\theta} \mathrm{Ne}_i \Rightarrow \sum X_i\, W_i = \sum\nolimits_n x_{ni} w_{ni} = \mathrm{net}_i$$
$$\Rightarrow F\{\mathrm{net}_i\} \xRightarrow{\theta} Z_i \Rightarrow \mathrm{out}_i = 0 \Rightarrow X_i \notin A_i. \qquad (2.3)$$

The process of correct recognition is completely described by 2.2 expression, whereas the incorrect recognition process can be varied. It depends on the number of patterns to be recognized, on the similarity measure, on the decision making algorithm. In our case we deal with the recognition process through neurons or a fixed similarity measure as well as the decision making algorithm. Therefore, the list of enumerated incorrect recognitions is considerably shortened. In particular, if the realization to be recognized is presented to a neuron of a “different” type, for example, $X_i \Rightarrow \mathrm{Ne}_j$, where $i, j = \overline{1; I}$, $i \neq j$, $X_i \in A_j$. When the number of recognitions is more than two in the decision making algorithm the neuron threshold initial (heuristically chosen) value is frequently admissible; if the value is exceeded, out = 1 for several patterns. In such a case the decision of attributing it to the type for which the given condition is true, the decision is made according to the net function maximum value or creating a different algorithm.

Later on we will discuss the process of correcting incorrect recognition results for 2.3 expression and for the cases when a realization of one type is presented for recognition to a “different” type of neuron. It should be taken into consideration that the above-mentioned situations fully comprise the list of incorrect (false) recognition processes. Let us imagine a process of incorrect recognition when X_j realization is to be recognized and is presented to A_i type Ne_i neuron:

$$X_i \xRightarrow{\theta} \mathrm{Ne}_i = W_i \Rightarrow \sum\nolimits_n x_{nj} w_{nj} = \mathrm{net}_{ji}$$
$$\Rightarrow F\{\mathrm{net}_{ji}\} \xRightarrow{\theta} Z_i \Rightarrow \mathrm{out}_{ji} = 1 \Rightarrow X_j \in A_i. \qquad (2.4)$$

In 2.3 and 2.4 expressions weighting coefficients $W_i = \{W_{ni}\}$ of one, namely, Ne_i neuron are used; these coefficients must ensure, in one case, the correct recognition of ”their own” realizations, in the other case, - discarding realizations of a different type.

Let us consider that $F(\mathrm{net})$ function is generally nonlinear but it has a linear section as well. Later on we will consider that $F(\mathrm{net})$ function parameters and weighting coefficients are chosen in such a way that the relation between net parameters and $F(\mathrm{net})$ function is linear and monotonously growing. Therefore, in the further process of correcting recognition errors we can directly use net function values considering the fact that scaling of net function values is possible during the process of correction as the need arises.

Let us consider $F(\mathrm{net})$ function in the given range as linear which satisfies monotony conditions. Then we will have the following expression for describing the correct and incorrect recognition process:

$$F(\mathrm{net}) = k \cdot \mathrm{net} = k \sum\nolimits_n x_{ni} w_{ni},$$
$$\text{where } k > 0, k = const. \qquad (2.5)$$

In fact k, represents a scaling coefficient which changes in the process of adjusting the weighting coefficient. Later on, in the process of correcting mistakes we will have that the initial value of k coefficients equals one and in the process of correlation the scaling coefficient is automatically established at the necessary level.

According to the above-mentioned, for describing the expressions 1.10 and 2.3 of the incorrect recognition process we will have:

1. While presenting its own realization we have ($X_i \xRightarrow{\theta} \mathrm{Ne}_i$)

$$k \sum\nolimits_n x_{ni} w_{ni} < Z_i\,; \qquad (2.6)$$

2. While presenting a different realization we have ($X_i \xRightarrow{\theta} \mathrm{Ne}_i$) Expressions 1,10 and 2,4

$$k \sum\nolimits_n x_{nj} w_{ni} \geq Z_i\,. \qquad (2.7)$$

Let us assume that $k = 1$, then 2.6 and 2.7 inequalities will be expressed as follows:

$$\sum\nolimits_n x_{ni} w_{ni} < Z_i \qquad (2.8)$$

$$\sum\nolimits_n x_{nj} w_{ni} \geq Z_i \qquad (2.9)$$

It should be taken into account that for accomplishing the recognition process (algorithm) it is necessary to give initial values to the weighting coefficients and thresholds. In literature, there are numerous methods and algorithms for forming such values. The majority of authors select $\{w_{ni}\}\{Z_i\}$ parameters heuristically and randomly. In our case, as it was mentioned above, in the process of recognition and correction we only use award procedures, e.g. for $\{w_{ni}\}$ coefficients we have:

$$w_{ni}[\rho+1] = w_{ni}[\rho] + \alpha \qquad (2.10)$$

where $\rho = 0,1,2,\ldots$ is the number of the step.

From 2.4 expression it follows that we can receive that

$$\forall w_{ni}[0] = 0, \quad n = \overline{1;N}, \quad i = \overline{1;I}, \qquad (2.11)$$

In a special case when the condition 1.10 and $\alpha = 1$ is fulfilled then we get the so-called statistical etalons [1] where we have:

$$w_{ni} = \frac{1}{M_i} \bigcup\nolimits_n X_{ni}^{m_i}, m_i = \overline{1;M_i},\ i = \overline{1;I},\ n = \overline{1;N} \ (2.12)$$

where M_i is the number of realizations in A_i type learning cluster. By means of 2.4 expression we get the realization of overlay procedure for a learning cluster of any type of set. The received set of weighting coefficients can be used for initial $W_n[0]$ values.

Finally, for the correct recognition the following inequality is true:

$$\sum\nolimits_n x_{ni} w_{ni} \geq Z_i \qquad (2.13)$$

If the recognition is incorrect, the following inequality is true:

$$\sum\nolimits_n x_{ni} w_{ni} < Z_i \qquad (2.14)$$

3. Correction of the Errors in the Process of Recognizing "Their Own" Realizations

The incorrect recognition process is described by means of 2.3 expression. If we take into account 1.10 expression, then the incorrect recognition process can be described in more detail by means of the following expression (2.3 expression modification):

$$X_i \overset{0}{\Longrightarrow} Ne_i \Rightarrow \sum\nolimits_n x_{nj} w_{ni} = net_{ji}$$
$$\Rightarrow F\{net_{ji}\} \overset{0}{\Longrightarrow} Z_i \Rightarrow out_i = 0 \Rightarrow X_i \notin A_i \qquad (3.1)$$

The expression 2.7 is analogous to 3.1 expression received by means of assumptions (error making) in the previous chapter; this expression can be presented in the contracted form as follows:

$$\sum\nolimits_n x_{ni} w_{ni} < Z_i \qquad i = \overline{1;I} \qquad (3.2)$$

For correcting the recognition mistake it is necessary to change the inequality symbol in 3.2 expression, which is possible through changing the elements of vectors (matrices) x_{ni} or w_{ni}. It is obvious that realization elements cannot be changed, therefore, it is only admissible to change or, in our case to increase the weighting coefficients which is termed awarding in the theory of neural networks. Awarding can be carried out by applying iteration procedures, e.g. through 2.10 expression where the initial values of the weighting coefficients can be given according to 2.11 expression.

Let us assume that according to step we received the $W_{ni}[\rho]$ set of weighing coefficients. In this case, if the error is not corrected, then 3.2 expression can be presented as follows:

$$\sum\nolimits_n x_{ni} w_{ni}[\rho] < Z_i \qquad (3.3)$$

If 3.3 expression is true, we will have to continue the iteration (awarding) process by 2.10 expression i.e. to increase $w_{ni} = [\rho]$ weighing coefficients until the inequality symbol in 3.3 expression is changed:

$$\sum\nolimits_n x_{ni} w_{ni} \geq Z_i \qquad (3.4)$$

where w_{ni} is a set of weighing coefficients for which 3.4 inequality is true.

If we use 2.10 procedure for all A_i–type realizations, where an error occurred, then due to the fact that applying the awarding procedure does not change the correct recognition results, we will get a correct recognition of A_i-type learning cluster of $\left\{X_i^{m_i}\right\}$ realizations.

Let us carry out the above-mentioned error correction and the recognition process for the realization of all set patterns of learning sets, accordingly we will have the correct recognition of all the realizations for recognizing patterns of a learning set.

There is an obstacle in the described procedure of error correction; this obstacle is termed as "repletion" in the theory of neural networks. Repletion implies such a growth of the weighing coefficients in the process of error correction, when the values of a separate and total coefficients exceed the possibilities of a neural network and the computer. In literature we have numerous methods for overcoming this obstacle; the simplest of them implies the proportional decreasing of neural network parameters which corresponds to the division of both sides of 3.4 expression (inequality) by nonnegative integral number. This problem is topical when we have a great number of patterns or in case of an increased sign space dimension. In this respect the activation function nonlinearity is more important; it suppresses the values of the activation function which correspond to a high value of the argument. In addition, the method of awarding infinitely small positive quantities to initial values of weighting coefficients is applied.

3.1. Correction of the Errors in the Process of Recognizing "Different" Realizations

The error at recognizing a different type of realization is given in 2.3 expression and 2.9 expressions obtained through further assumptions and simplifications:

$$\sum\nolimits_n X_{nj} W_{ni} \geq Z_i$$

Due to the fact that w_{ni} weighing coefficients have changed in the process of error correction given in the previous paragraph, instead of 2.9 expression we will have;

$$\sum\nolimits_n x_{ni} w_{nj}^* \geq Z_i \qquad (4.1)$$

If 4.1 inequality is true, then we will have an incorrect recognition result $X_{nj} \in A_i$ (expression 2.4). To correct this error it is necessary to change $w_{ni}{}^*$ and Z_i parameters so that the results of correct recognitions obtained in the previous chapter are not replaced by an incorrect recognition. It means that $w_{ni}{}^*$ and Z_i parameters should be changed in such a way that the inequality sign in 4.1 expression is reversed.

Let us take into account that we use an awarding procedure (expression 2.10) as a learning method; this is caused by the necessity of error correction in the process of recognizing realizations of their own pattern.

Let us use the awarding procedure for the right side of 4.1 expression, that is for Z_i threshold of Ne_i neuron. Similar to 2.10 procedure we will have:

$$Z_i[\alpha] = Z_i[\alpha - 1] + \Delta Z \qquad (4.2)$$

Where $\alpha = 0,1,2,\dots$ is the step number. We increase the neuron threshold by 4.2 iterative expression until the inequality sign in 4.1 expression is reversed; as a result we will get the following expression:

$$\sum\nolimits_n x_{ni} w_{ni}^* < Z_i[\alpha] = Z_{\mathrm{i}}^* \qquad (4.3)$$

where $Z_i{}^*$ is the required value of the threshold.

If 4.3 expression is true, then the recognition process described by 4.1 expression is incorrect, that is the error made in the process of recognizing X_j realization is corrected.

In order to avoid errors that Z_i threshold increase may cause during the process of recognizing by Ne_i neuron its own $X_i$realizations it is necessary to take certain measures that will enable us to determine that error making in the process of recognizing its own realizations can be avoided. Let us consider that set pattern realizations are binary (condition 1.4), which enables us to make a list of possible situations for different patterns of realizations, for example, X_i and X_j realizations for X_{ni} and X_{nj} components (table 4.1). It is evident, that realizations may belong to any two different patterns; the n index value of their properties (characteristic features) is the same.

Table 4.1

Situation	1	2	3	4
X_{ni}	0	1	0	1
X_{nj}	1	0	0	1

The recognition of A_i type X_j realizations is correct if 3.4 inequality is true. If we consider that the neuron threshold has increased according to 4.2 and 4.3 expressions, then 3.4 expression will be presented as follows:

I) $\sum\nolimits_n x_{ni} w_{ni} \geq Z_i^*$ recognition is correct or

II) $\sum\nolimits_n x_{ni} w_{ni} < Z_i^*$ recognition is incorrect (expression 4.4).

Situation 1: $X_{ni} = 0; \quad X_{nj} = 1$

In the first case which is described by 4.4 expression, the recognition is correct, so it is not necessary to change w_{ni} weighing coefficient; in the second case the increase in w_{ni} will give no result (no error connection) because $w_{ni} = 0$, as we see in case of situation 1 the increase in w_{ni} weighting coefficient is ineffective.

Situation 2: $X_{ni} = 1; \quad X_{nj} = 0$

Proceeding from 4.4 expression the increase in w_{ni} coefficients helps to make the first inequality of the expression stricter or, which is the same, strengthens the process of correct recognition; in case of the second inequality the increase in Wni weighting coefficient becomes the necessary condition for correcting the recognition error. Consequently, it becomes evident that in case of situation 2 the increase in weighting coefficient is effective according to the increase in the reliability of the recognition

Situation 3: $X_{ni} = 0; \quad X_{nj} = 0$

In case of situation three, on the basis of 4.4 expression the recognition result (correct or incorrect) will not change, as we have weighing coefficients multiplied by zero.

Situation 4: $X_{ni} = 1; \quad X_{nj} = 1$

In case of situation four, the recognition of X_i realization will improve. But it is possible to make errors in the X_j realizations recognition process, which occurs when $Z_j < Z_i$, so we must consider other circumstances in case of the increase in w_{ni} coefficients, or we must avoid the increase procedure and search for another way out to correct the erroneous recognition.

The analysis of the above-given situation shows that for the correct recognition of any type of realization it is necessary for the etalon description of this pattern to have at least one sign whose realization probability for one pattern is equal to one, whereas it is equal to zero for other patterns. If there is such a sign, then the correction of the recognition error is possible through using an awarding procedure.

In other cases it is necessary to use the existing methods for evaluation or to work out a new one.

4. Signs Evaluation (Ranging) According to Learning Set Realizations

Let us assume that we have chosen (heuristically or according to other factors) signs-characteristic features that represent A set patterns in this sign space. The presentation of patterns in the sign space occurs by means of realizations whose totality for each pattern forms or must form a separate cluster (group). It is implied that the number of clusters must be equal to the number of patterns or exceed it. The position of clusters in the sign space determines their separateness degree and, accordingly, the reliability of recognition process. Namely, the more separated the type-reflecting clusters are from one another, the more reliable the recognition process is and vice versa; in case of closely positioned clusters the recognition process becomes more complicated and the recognition becomes unreliable. The process of selecting signs, as well as its result, greatly influences the position of clusters in the sign space, so the selection of signs should be based on the existing scientific information (if it is possible) in the given sphere. The next stage may be selected according to evaluating signs together or separately.

In our case the process of evaluating-selecting signs is considered according to its reliability which makes it necessary to introduce some concepts – terms through which we will characterize certain properties (signs) from the point of view of recognition, content by means of formalizing these concepts later on.

To evaluate a separate sign in the sign space let us use general and content criteria:

To what extent a concrete pattern is characterized by a given sign.

To what extent a given concrete pattern of a sign differs from the same sign of another pattern.

Characteristic features represent the property-characteristic of the object which the object possesses always or very frequently; this is expressed by the fact that we have this property in the object realizations or if it is measurable, its values are steadily placed in a certain line of values; the sign possesses a property distinguishing it from other patterns, if it satisfies the condition of specificity and, at the same time, this condition is not fulfilled clearly in relation to other signs.

According to the condition, 1.4 expression, the sign space and, correspondingly, pattern realizations are binary. At the same time, any realization of any pattern (vectors or matrices) has equal dimensions, which makes it possible to present the concepts “characteristic” and “distinguishing” in a formalized way.

Let us place the realizations of a learning cluster according to types, after this with the help of superposition (overlay) let us make the so-called “statistical” description (etalon) for each type separately [1]. Let us designate the A_i type statistical etalon with S_i (vector or matrix) the coordinates of which are:

$$S = s_1, s_2, \dots, s_{ni}, \dots, s_{Ni}, \dots \quad i = \overline{1; I} \tag{5.1}$$

Where for $\forall s_{ni}$ we have:

$$S_{ni} = \frac{R_i}{M_i} \qquad i = \overline{1; I} \tag{5.2}$$

Where R_i is the number of those realizations in A_i pattern learning set in which X_{ni} sign gets the value equal to one. Proceeding from 5.2 expression we can formalize “characteristic” and “distinguishing” signs:

Sign x_{ni} is absolutely characteristic of A_i type if $S_{ni} = 1$.

Sign x_{ni} is absolutely distinguishing for A_i type if $S_{ni} = 1$ and, $S_{nj} = 0$, $j = \overline{1; J}$, $j \neq i$

For the situations discussed in chapter 2 x_{ni} sign is absolutely characteristic for A_i pattern if $x_{ni} = 1$ (situation 2 and situation 4) for all the realizations of A_i pattern learning set.

The same sign is absolutely distinguishing for A_i pattern if the following condition is fulfilled:

$x_{ni} = 1$; $x_{nj} = 0$, (situation 2). $\forall i, j \in I; i \neq 0$.

It is evident that the greater the number of the absolutely characteristic and distinguishing signs in the description of the given pattern, the higher the chance of correct recognition for this pattern of realization and vice versa.

Unfortunately, the combined occurrence of characteristic and different signs for a real object is quite rare; due to this it becomes necessary to introduce intermediate concepts and their formalized definitions. In particular, let us assume that x_{ni} xni sign is partially characteristic of A_i pattern if the following condition is fulfilled:

$$0{,}5 \leq S_{ni} \leq 1 \tag{5.3}$$

x_{ni} sign is partially distinguishing for A_i pattern if 5.3 condition and the following condition are fulfilled:

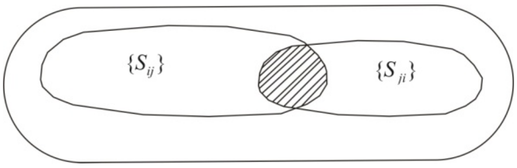
$$0 \leq S_{ni} \leq 0{,}5 \qquad j = \overline{1; J}, i \neq j \tag{5.4}$$

It is obvious that the greater the number of signs satisfying the 5.3 and 5.4 conditions, the higher the chance of correct recognition and vice versa.

It can be said that through carrying out the learning processes discussed in chapters 3 and 4 the error-free recognition of learning set realizations is achieved. If there are no absolutely characteristic and distinguishing signs for the patterns, then it becomes necessary to find satisfying signs for conditions 5.3 and 5.4; this will make it possible to correctly evaluate the recognition process for such cases. Let us designate the set of A_i pattern signs for A_j type which satisfy 5.3 and 5.4 conditions $\{S_{ij}\}$ and by S_{ji} set - A_j pattern signs which satisfy the same conditions in relation to A_i pattern; the correlation of the whole set of signs $\{X\}$ and S_{ij} S_{ji}, S_{ji} sets is given in Picture 5.1:

Let us assume that the number of signs in S_{ij} set is equal to R_{ij}, whereas in S_{ji} set it is equal to R_{ji}; for the corresponding indexation of these signs in order to describe their total sequence let us use symbols $r_{ij} = \overline{1; R_i}$; $r_{ji} = \overline{1; R_j}$; let us consider expressions:

$$\sum\nolimits_r S_{rij} = Q_{ij}; \quad \sum\nolimits_r S_{rji} = Q_{ji} \tag{5.5}$$



Picture 5.1

As in equalities $R_{ij} < N$ and $R_{ji} < N$ are fulfilled, the following inequalities are also true:

$$Q_{ij} \le \sum\nolimits_n S_{ni}]C_i; \quad Q_{ji}\sum\nolimits_n S_{nj} = C_j \qquad (5.6)$$

We can characterize (range) separate elements or groups of elements of a sign set from the point of view of recognition reliability which depends on how big is Q_{ij} value which constitutes C_i value and on how big is Q_{ji} value which constitutes C_j value.

The following inequalities are received considering 5.6 expression:

$$C_i - Q_{ij} \ge 0; \quad i, j = \overline{1, I}, \quad i \ne j \qquad (5.7)$$

1. A sign is absolutely useful if 5.7 inequality is true for ∀i and ∀j. In this case there is at least one sign in the sign space which ensures the A_i pattern error-free recognition for learning set realizations of the given set of patterns.
2. A sign is useful if 5.7 expression is fulfilled for a part of patterns, e.g. for half of the patterns:

$$C_i - Q_{ij} \ge 0; \quad j = \overline{1, I_1}, \quad 0{,}5I \le I_1 < I \qquad (5.8)$$

3. A sign is less useful if 5.7 expression is true for less than a half of patterns;

$$C_i - Q_{ij} \ge 0; \quad j = \overline{1, I_2}, \quad I_2 < 0{,}5I \qquad (5.9)$$

The I1, I2 coefficient values are taken heuristically, their improvement is possible and should be made for a concrete set of types and signs.

It is obvious that absolutely useful signs for real objects and the sign space are very rare, whereas useful and less useful signs for the vast majority of patterns and signs occur quite often.

It is evident that the greater the number of absolute or useful signs is, the greater is the possibility of correct recognition of learning processes described in the previous chapters and vice versa.

5. The Plan and Algorithm for Experimental Research

The error correction methods and sign evaluation algorithms given in the previous chapters are based on the exact knowledge of the elements of pattern realizations set as well as the realization of their learning set. According to the condition the set of signs for all the patterns and accordingly for all the realizations should be equal and binary. To carry out the realizations of learning set patterns it must be determined which pattern each of them belongs to; this must be considered attentively as otherwise this will cause the incorrect description of patterns in the learning process and eventually we will get incorrect recognition results.

The stages of experimental research are as follows:

Building neural networks or selecting them from the existing ones;

The setting up – calculating the threshold coefficient for each neuron in the network;

Awarding (calculating) the weighting coefficient for each sign;

Error correction in the process of recognizing their own realizations;

Error correction during the recognition of different realizations;

Carrying out feature's evaluation (algorithm).

In order to carry out the above mentioned stages it is necessary to create the so-called "statistical" descriptions, after which it becomes possible to calculate the threshold for each neuron by way of conducting the recognition process (process algorithm). If necessary, it is possible to get the error-free recognition of learning set patterns realizations by means of changing-improving the weighting coefficients of a neuron. This article has a theoretical character, so explained method may be used for practical tasks, which are executed by neural nets.

REFERENCES

[1] McCulloch, Warren; Walter Pitts. "A Logical Calculus of Ideas Immanent in Nervous Activity". *Bulletin of Mathematical Biophysics* 5 (4): 115–133. 1943, doi:10.1007/BF02478259.

[2] Hebb, Donald. *The Organization of Behavior*. New York: Wiley, 1949.

[3] Farley, B.G.; W.A. Clark "Simulation of Self-Organizing Systems by Digital Computer". *IRE Transactions on Information Theory* 4 (4): 76–84, 1954. doi:10.1109/TIT.1954.1057468.

[4] Rochester, N.; J.H. Holland, L.H. Habit, and W.L. Duda. "Tests on a cell assembly theory of the action of the brain, using a large digital computer". *IRE Transactions on Information Theory* 2 (3): 80–93. 1956 doi:10.1109/TIT.1956.1056810.

[5] Rosenblatt, F. "The Perceptron: A Probalistic Model For Information Storage And Organization In The Brain". *Psychological Review* 65 (6): 386–408. 1958, doi:10.1037/h0042519. PMID 13602029.

[6] Werbos, P.J.. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, 1975.

[7] Minsky, M.; S. Papert. *An Introduction to Computational Geometry*. MIT Press. ISBN 0-262-63022-2, 1969.

[8] Rumelhart, D.E; James McClelland Parallel Distributed Processing: Explorations in the Microstructure of Cognition.

Cambridge: MIT Press, 1986.

[9] http://www.kurzweilai.net/how-bio-inspired-deep-learning-keeps-winning-competitions, 2012 Kurzweil AI Interview with Jürgen Schmidhuber on the eight competitions won by his Deep Learning team 2009–2012.

[10] Graves, Alex; and Schmidhuber, Jürgen; *Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks*, in Bengio, Yoshua; Schuurmans, Dale; Lafferty, John; Williams, Chris K. I.; and Culotta, Aron (eds.), *Advances in Neural Information Processing Systems 22* (*NIPS'22*), *December 7th–10th, 2009, Vancouver, BC*, Neural Information Processing Systems (NIPS) Foundation, 2009, pp. 545–552.

[11] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber. A Novel Connectionist System for Improved Unconstrained Handwriting Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, 2009.

[12] http://www.scholarpedia.org/article/Deep_belief_networks /

[13] Hinton, G. E.; Osindero, S.; Teh, Y. "A fast learning algorithm for deep belief nets". *Neural Computation* 18 (7): 1527–1554. doi:10.1162/neco.2006.18.7.1527.PMID 16764513, 2006.

[14] Fukushima, K. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". *Biological Cybernetics*36 (4): 93–202. doi:10.1007/BF00344251. PMID 7370364, 1980.

[15] M Riesenhuber, T Poggio. Hierarchical models of object recognition in cortex. Nature neuroscience, 1999.

[16] Deep belief networks at Scholarpedia.

[17] Hinton, G. E.; Osindero, S.; Teh, Y. W. "A Fast Learning Algorithm for Deep Belief Nets". *Neural Computation* 18 (7): 1527–1554.doi:10.1162/neco.2006.18.7.1527. PMID 167645 13. edit, 2006.

[18] John Markoff. "Scientists See Promise in Deep-Learning Programs". *New York Times*. November 23, 2012.

[19] Verulava O., Khurodze R. “Theory of Rank of Links: Modeling of Recognition Process”, Nova Science Publishers, Inc. New York, 2011.

[20] Verulava O. Khurodze R., Clustering Analysis and Decision-making by “Rank of Links”, Mathematical Problems in Engineering, 2002, Vol. 8(4-5), pp. 475-492, Taylor & Francis group.