

---

---

NEW TECHNOLOGIES  
IN MECHANICAL ENGINEERING

---

---

# Neural Network Modeling and Organization of a Hierarchical Associative Memory System

I. V. Stepanyan

*Mechanical Engineering Research Institute of the Russian Academy of Sciences, Moscow, Russia*

*e-mail: neurocomp.pro@gmail.com*

Received April 29, 2021; revised July 5, 2021; accepted July 16, 2021

**Abstract**—The organization of associative memory plays a crucial role in the efficiency of intelligent systems, software, and practical applications that require analysis and processing of big data and knowledge. A logical architecture of a neuro-like hierarchical pattern associative memory system, which provides storage of incoming information in the form of so-called patterns, is proposed and described. Patterns are hierarchically organized memory elements that make it possible to recover data from its part and find elements within automatically determined contexts on the basis of previously stored information. Modern approaches to the organization of associative memory in intelligent systems are considered. The main elements of pattern memory, i.e., pattern, neuron, mediator, and synapse, are defined. The basic concepts and principles of functioning of hierarchical pattern associative memory systems are described, and examples of practical implementation are given. Based on the principles described, it is possible to create specific memory versions for practical tasks, such as processing text, graphic, audio, and video information. Shared memory can be used to generalize various information flows into common logical chains of memories. It is concluded that the capabilities of the proposed hierarchical pattern memory systems for handling data make it possible to store information in intelligent systems and model the thinking process of an expert in the course of logical reasoning.

**Keywords:** associative memory, hierarchical pattern, neuron, synapse

**DOI:** 10.3103/S1052618821080148

The abstraction of associative memory is becoming increasingly important in computing systems. Technical and software tools based on neural network principles began to develop intensively in the early 1980s. The electronic brain and the biological brain have been constantly compared throughout the history of computing. Thanks to a significant breakthrough in the field of parallel computing using graphics accelerators and creation of specialized devices for such purposes, nowadays we have managed to obtain hardware–software complexes that surpass humans in a whole class of applied intellectual tasks.

Modern information retrieval and information-analytical systems are designed to store and process highly specialized information (text, hypertext, images, etc.). Factographic knowledge banks can also operate only in certain highly specialized subject areas, taking into account the structure and specificity of the processed knowledge and data. In this regard, there is a problem of developing such an associative memory organization system that implements the functionality of storage and processing of arbitrary data regardless of the internal structure of these data.

The research problem can be formulated as follows: the development of general principles of associative memory organization for intelligent systems. In order to solve this problem, it is advisable to rely on the principles of neural network organization.

## OVERVIEW OF THE CURRENT STATE OF THE PROBLEM

From the viewpoint of neurophysiology, associative memory implements the processes of learning and forgetting, which are at the basis of I.P. Pavlov's theory of the conditioned reflex. The associative memory scheme developed in [1] was designed to implement generalization and differentiation on the basis of Pavlov's model. In fact, the core associative memory architecture can be based on both exact matching and neural network models. Article [2] presented a classification of various software architectures for organizing associative computations.

Associative memory is intensively studied in the literature and has been the focus of a large amount of applied research. In [3], a set of approaches to managing associative memory with undesirable dynamics was presented. The analysis includes system dynamics during memory retrieval. Study [4] was devoted to the development of a logical model of two-level fuzzy associative memory with autoencoding, which consists of a pair of functional modules, one of which implements dimensionality reduction to fill the logically oriented associative memory. The optimization of associative matrices includes both gradient learning mechanisms and population optimization algorithms (particle swarm optimization and differential evolution). A similar approach was presented in [5], where the two-phase associative classification algorithm was generalized: the first phase was based on the use of autoassociative memory, and the second phase computed the normalized difference between the results of the first phase and each pattern of the initial set.

In [6], the Hopfield memory neural network model was analyzed, which considers training and inverse patterns as equivalent. In [7], an inference model of an intuitive search process in continuous distributed associative memory functioning in the character processing mode was proposed, and a computational model of logical inference as well as the occurrence of tree-like search behavior was shown. In [8], an associative model of robot memory influenced by emotions was developed.

Fuzzy morphological associative memory is a generalization of fuzzy associative memory models. In [9], the theoretical foundations of interval fuzzy morphological associative memory, the weight matrices of which were constructed using representative interval-valued fuzzy operators, was presented. In [10], the concept of associative memories was supplemented and an augmentation of fuzzy clustering in the form of so-called collaborative fuzzy clustering was developed.

In [11], a quantum probabilistic associative memory was proposed that uses the inverse quantum Fourier transform and Grover's algorithm to recover existing or similar patterns in the memory. The content of the memory is created using a superposition state generator representing a given set of patterns. In [12], a passive bidirectional crosstalk predifferentiation architecture was applied to implement synaptic weights, and a compact, low-voltage neuron that consumes little energy was developed. Associative learning and forgetting processes can constitute a fully functional model of the emotion apparatus [13]. This model is closely related to the physical properties of memristors for the design of synaptic structures of intelligent machines.

#### PATTERN ASSOCIATIVE MEMORY MODEL: GENERAL PRINCIPLES OF ORGANIZATION AND EXAMPLES OF FUNCTIONING

In the present article, we consider a memory model based on some simplified neurophysiological principles. In order to build a model of associative memory, we take into account the properties of human memory, for which the neocortex is responsible. The neocortex stores sequences of patterns, recalls patterns autoassociatively, stores patterns in an invariant form, and stores patterns hierarchically.

These properties were implemented in the form of a prototype program in Python (we used this program for neural network modeling of pattern memory and demonstration of examples). At the same time, the proposed memory model is not a software implementation of a living brain or an individual part of it, but a set of algorithms that can be used to implement individual functions that provide behavior similar to that of a biological brain.

The development of pattern memory is based on the following definitions and principles. Knowledge is a network consisting of neurons with basic knowledge and synapses. A pattern is a sequence of interconnected basic knowledge with a fixed length. All information enters memory sequentially in the form of patterns. A neuron is a program object, a model of a biological neuron. A mediator is a program object that appears when knowledge is added. It emerges when a neuron appears/activates and has an attribute, power, which decreases with time. The mediator exists as long as its power is above zero. A synapse is a directed connection between neurons. Each synapse has a weight.

The synapses are formed with the help of mediators. A synapse is assigned an initial weight equal to its mediator power. The minimum amount of information is the basic knowledge that a single neuron can store. When memorizing patterns, the structure of the network is changed by adding new neurons and synapses to the already existing knowledge.

Let us consider an example of how the basic functions of pattern associative memory work. Here and below, for neurons, the upper index stands for the value and the lower index, for its sequence number. For synapses, the upper index means weight and the lower index, its ordinal number. For mediators, the upper index is the power (it is shown in parentheses for the neuron from which it was created) and the lower index, the ordinal number of the mediator.

**Step 1.** Memorization of the first pattern: P0 0 1 2 3 4 5 6 7 8 9. The first position contains the name of the pattern; the other positions code the pattern of ten elements (characters or a sequence of them). After remembering the pattern P0, 11 neurons  $\{n_1^{P0}, n_2^0, \dots, n_{11}^9\}$  and 20 synapses  $\{s_1, s_2, \dots, s_{20}\}$  are generated, where  $n_1^{P0}$  is the first neuron with a P0 value, and  $s_1$  is the first synapse. Each neuron  $n$  has a unique identifier and a value (base knowledge) which it stores (in our case, it is textual information).

All generated neurons have synapses  $\{s_1^2(n_1^{P0} \rightarrow n_2^0), s_2^1(n_1^{P0} \rightarrow n_3^1), s_3^2(n_2^0 \rightarrow n_3^1), s_4^1(n_2^0 \rightarrow n_4^2), \dots, s_{19}^1(n_{10}^8 \rightarrow n_{11}^9), s_{20}^1(n_{11}^9 \rightarrow n_1^{P0})\}$ , where  $s_1^2(n_1^{P0} \rightarrow n_2^0)$  is the first synapse with weight 2 from neuron  $n_1^{P0}$  to  $n_2^0$ . Each synapse has its own unique identifier, a pointer to the neuron with which it is associated, a pointer to the pattern in which it is involved and the weight. The symbol  $\rightarrow$  means a synaptic connection.

Therefore, according to the algorithm, two outgoing synapses are created for all neurons from  $n_1$  to  $n_{k-2}$  and one for  $n_{k-1}$  and  $n_k$ . That is, the synapses specify connections between adjacent images and through one in the sequence (pattern). This is due to the principle of adding new neurons using mediators.

When  $n_i$  is added, a mediator  $m^{Pmax}(n_i)$  is formed from it. The mediator is the mechanism on the basis of which the synaptic connection will be set for the next cycle of information processing

$$n_i \Rightarrow m^{Pmax}(n_i),$$

where  $m^{Pmax}(n_i)$  is the mediator of power Pmax formed from the neuron  $n_i$ ; the symbol  $\Rightarrow$  means that the left part  $m$  at the previous iteration sets the right part for the next iteration. When a neuron  $n_{i+1}$  is added, the current mediator binds it, thus forming a synapse with a weight equal to the power index of the current mediator

$$m^p(n_i) \Rightarrow s^p(n_i \rightarrow n_{i+1}).$$

Therefore, the mediators set the number of connections in 1, 2, 3, etc., in the pattern when forming connections. In this example, the maximum power of the new mediator is  $Pmax = 2$ .

At each iteration, all mediators lose a unit of power:  $m_i^p > m_i^{p-1}$ , where the symbol “>” means that the left part goes to the right part at the next iteration. When  $P(m) = 0$ , the mediator is destroyed and no longer exists.  $m^0 > \emptyset$ , where  $\emptyset$  means that the object no longer exists. This mechanism makes it possible to implement the property of autoassociativity and an invariant form of patterns. The process of remembering the pattern P0 consists of the following steps:

1. When  $n_1^{P0} : n_1^{P0} \Rightarrow m^2(n_1^{P0})$  is added.

2. When adding  $n_2^0$ :

(1)  $m^2(n_1^{P0}) \Rightarrow s^2(n_1^{P0} \rightarrow n_2^0)$ ;

(2)  $m^2(n_1^{P0}) > m^1(n_1^{P0})$ ;

(3)  $n_2^0 \Rightarrow m^2(n_2^0)$ .

3. When adding  $n_3^1$ :

(1)  $m^2(n_2^0) \Rightarrow s^2(n_2^0 \rightarrow n_3^1)$ ;

(2)  $m^1(n_1^{P0}) \Rightarrow s^1(n_1^{P0} \rightarrow n_3^1)$ ;

(3)  $m^2(n_2^0) > m^1(n_2^0)$ ;

(4)  $m^1(n_1^{P0}) > m^0(n_1^{P0}) > \emptyset$ ;

(5)  $n_3^1 \Rightarrow m^2(n_3^1)$ .

The addition of neurons from n42 to n108 is similar to that described above and is not described in detail.

4. When adding  $n_{11}^9$ :

(1)  $m^2(n_{10}^8) \Rightarrow s^2(n_{10}^8 \rightarrow n_{11}^9)$ ;

(2)  $m^1(n_9^7) \Rightarrow s^1(n_9^7 \rightarrow n_{11}^9)$ ;

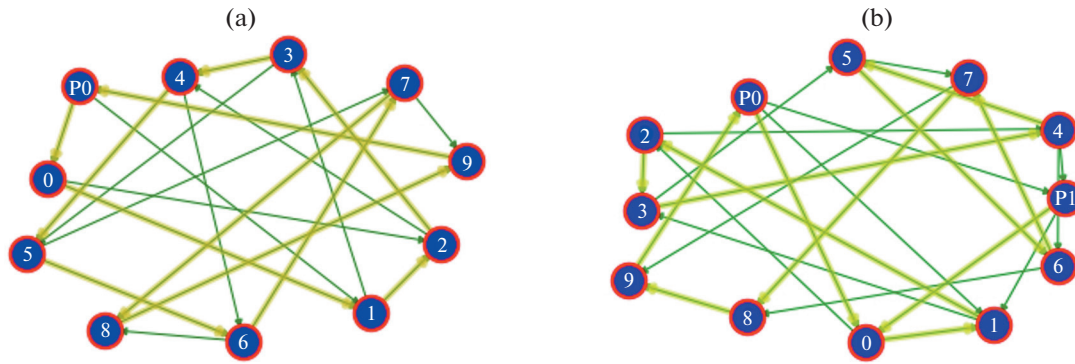


Fig. 1. Neuronal connectivity structure of (a) the pattern P0 and (b) the added pattern P1.

$$(3) m^2(n_{10}^8) > m^1(n_{10}^8);$$

$$(4) m^1(n_9^7) > m^0(n_9^7) > \emptyset;$$

$$(5) n_{11}^9 \Rightarrow m^2(n_{11}^9);$$

(6)  $m^2(n_{11}^9) \Rightarrow s^2(n_{11}^9 \rightarrow n_1^{P0})$ , a finalizing synapse was formed, which is an iterative reference, the last element of the pattern closed to the first one;

$$(7) m^2(n_{11}^9) > m^1(n_{11}^9);$$

$$(8) m^1(n_{10}^8) > m^0(n_{10}^8) > \emptyset;$$

$$(9) n_1^{P0} \Rightarrow m^2(n_1^{P0}).$$

5. After all neurons have been added, all mediators are reduced:

$$(1) m^2(n_1^{P0}) > m^1(n_1^{P0});$$

$$(2) m^1(n_{11}^9) > m^0(n_{11}^9) > \emptyset.$$

Step 2. Memorization of the second pattern: P1 0 1 2 3 4. The result of this command is shown on the right-hand side of Fig. 1. The changes in this step are as follows:

1. When adding  $n_1^{P1}$ :

(1) The mediator  $m^1(n^{P0}) \Rightarrow s^1(n^{P0} \rightarrow n_1^{P1})$  which remained from the last step. This mechanism makes it possible to store pattern sequences.

$$(2) m^1(n^{P0}) > m^0(n^{P0}) > \emptyset;$$

$$(3) n_1^{P1} \Rightarrow m^2(n_1^{P1}).$$

2. When adding  $n_2^0$ :

$$(1) m^2(n_1^{P1}) \Rightarrow s^2(n_1^{P1} \rightarrow n_2^0);$$

$$(2) m^2(n_1^{P1}) > m^1(n_1^{P1});$$

$$(3) n_2^0 \Rightarrow m^2(n_2^0).$$

Since the neuron  $n_2^0$  is not yet part of a given pattern, when a value of 0 is added, the neuron  $n^0$  is not created, but the existing one is selected.

3. When adding  $n_3^1$ :

(1) The merger of an existing synapse and a mediator causes the weight of the existing synapse to change.  $m^2(n_2^0) \cup s^2(n_2^0 \rightarrow n_3^1) > s^4(n_2^0 \rightarrow n_3^1)$ . The symbol  $\cup$  means merging the elements (in this case, the mediator is merged with the synapse with an increase in the weight of the synapse, because there is already a connection). In this example, the weight of the synapse  $s^4(n_2^0 \rightarrow n_3^1)$  consists of the sum of the powers of the two patterns P0 and P1.

$$p = p_1(P1) + p_2(P2) = p(P1, P2),$$

where  $p$  is the power of the mediator; P1 is the first pattern;  $p(P1)$  is the power of the first pattern; and  $p(P1, P2)$  is the power equal to the sum of the powers of patterns P1 and P2.

$$s^p(n_1 \rightarrow n_2) \equiv p^{p(P1, P2)}(n_1 \rightarrow n_2),$$

where  $s^p$  is the synapse power  $p$ ,  $s^{p(P1, P2)}$  is the synapse power equal to the sum of the powers of patterns P1 and P2.

$$(2) m^1(n_1^{P1}) \Rightarrow s^1(n_1^{P1} \rightarrow n_3^1);$$

$$(3) m^2(n_1^{P1}) > m^1(n_1^{P1});$$

$$(4) m^1(n_1^{P1}) > m^0(n_1^{P1}) > \emptyset;$$

$$(5) n_3^1 \Rightarrow m^2(n_3^1).$$

4. The weights of the following synapses also increased:

$$(1) s^2(n^1 \rightarrow n^2) > s^4(n^1 \rightarrow n^2);$$

$$(2) s^2(n^2 \rightarrow n^3) > s^4(n^2 \rightarrow n^3);$$

$$(3) s^2(n^3 \rightarrow n^4) > s^4(n^3 \rightarrow n^4);$$

$$(4) s^1(n^0 \rightarrow n^2) > s^2(n^0 \rightarrow n^2);$$

$$(5) s^1(n^1 \rightarrow n^3) > s^2(n^1 \rightarrow n^3);$$

$$(6) s^1(n^2 \rightarrow n^4) > s^2(n^2 \rightarrow n^4);$$

$$(5) \text{ A finalizing synapse } m^2(n_6^4) \Rightarrow s^2(n_6^4 \rightarrow n_1^{P1}) \text{ is formed;}$$

$$(6) n_1^{P1} \Rightarrow m^1(n_1^{P1}).$$

The rest of the network elements did not change. After memorizing the patterns P0 and P1, we obtain the network shown in Fig. 1. In the figure, neurons are shown as network nodes, while the border shows that there are outgoing connections. The arrows show the synapses. The thick arrow denotes a synapse with a greater weight.

**Step 3.** Data duplication. Pattern P2 1 2 5 1. Major changes:

$$(1) m^2(n_2^1) \cup s^4(n_2^1 \rightarrow n_3^2) > s^6(n_2^1 \rightarrow n_3^2);$$

(2) A new neuron  $n_5^1$  with a value of 1 was added. This neuron has the same value as the previously created neuron  $n_2^1$ , but it has a unique identifier. This mechanism of creating similar neurons within the same sequence with the same values ensures duplication of information and makes it possible to avoid looping a sequence of elements of the same pattern. The following connections with this neuron were formed:

$$(1) m^1(n^2) \Rightarrow s^1(n^2 \rightarrow n_5^1);$$

$$(2) m^2(n^5) \Rightarrow s^2(n^5 \rightarrow n_5^1);$$

$$(3) m^1(n_5^1) \Rightarrow s^2(n_5^1 \rightarrow n_1^{P2}).$$

The rest of the changes are made according to the rules described above.

**Step 4.** Memorization of three additional patterns P3 0 1 9 3 2 0 6, P4 4 4 4 4, and an upper level PP1 P0 P1 P2 P3 consisting of patterns instead of symbols (similar to a sentence made of words).

**Step 5.** Retrieval of a pattern from memory. It is possible to search for a particular pattern (neurons in the initial sequence) by name. When searching for pattern P0, we find P0; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9.

**Step 6.** Retrieval of multiple patterns. It is possible to select neurons that are part of several patterns. By patterns P1 and P2, we obtain: P1; 0; 1; 2; 3; 4; P2; 1; 2; 5; 1.

**Step 7.** A hierarchical pattern consisting of other patterns. The pattern PP1 will consist of the previously memorized patterns P0 P1 P2 P3.

**Step 8.** Search for a pattern by its elements. If we specify the elements 1 2 5, then P2 will be found.

**Step 9.** Search with missing elements.

When searching for a pattern by several elements of the sequence 3 2 6 with a missing element (0 between 2 and 6 is missing), the pattern P3 will be found. The number of missing elements is less than Pmax.

**Step 10.** Search by name and element. The search uses the name and element of the pattern. For example, a search by P2 and P3 will return P3 and PP1. The former result is due to the fact that P3 was set immediately after P2 and was automatically associated with P2.

**Step 11.** Search for an unknown sequence. Failure to find a pattern by an unknown sequence.

**Step 12.** Search for the next elements. When attempting to search for the next elements after 1 and 2, the following will be found: 3, weight 12; 4, weight 9; 5, weight 4; and 1, weight 3. If a particular pattern is not specified, the next elements are sought by possible patterns and the next elements in those patterns.

**Step 13.** Search for the next element after 3 2 0. It will return only one result, 6, weight 6.

**Step 14.** Search for the next element skipping elements 3 0. It will return 6, weight 3. Searching for a sequence with a skipped element reduces the weight of the result.

In order to demonstrate the dependence of the results on the number of memorizations, consider the single memorization of patterns P0 1 2 3 4 5 6 7 8 and P1 2 4 6 8. A pattern search by elements 2 and 4 will return P1; P0. The former result P1 is the best. Search for the next elements after 2 and 4 will return 6, weight 6; 5, weight 3; and 8, weight 3.

Repeated memorization of patterns leads to rearrangement of the weights of synapses and other results. When the pattern P0 1 2 3 4 5 6 7 8 is memorized five times, the best search result for elements 2 and 4 will change to P0. Search for the next elements after 2 and 4 will return 5, weight 15; 6, weight 14; and 8, weight 3. For this query, the weights of the results of  $n^5$  and  $n^6$  changed, after repeated memorization. Therefore, in memory after 2 and 4 the best result will be 5.

It is possible to evaluate the effect of the number of elements in the query. A larger number of elements in a search pattern changes the result. When one tries to find the pattern 2 4 6, the result will be P1; P0, and when one tries to find the next element after 2 4 6, the result will be 8, weight 21, and 7, weight 20. In contrast to the previous step, the result from the pattern P1 is obtained. After 2, 4, and 6, the next one in memory is 8.

With an additional four memorization iterations of the pattern P0 1 2 3 4 5 6 7 8, the best result changed to P0, and the next one after 2 4 6 changed to 7, weight, 36, and 8, weight 33. Therefore, without using additional features, the best answer is the result that was memorized more times.

Let us consider an algorithm for applying the pattern associative memory model to the textual information processing.

**Step 1.** Word memorization. When memorizing words, each word is represented as a pattern of a sequence of letters. The name of the top-level pattern (sentence) is generated automatically in the form of a globally unique identifier (GUID), which is a statistically unique 128-bit identifier. Consider the example of memorizing the words “птица летает высоко” (bird flies high). Patterns of each word consisting of letters and the top-level pattern consisting of words were memorized.

**Step 2.** Word search. After additional memorization of the words “пицца лежит на столе” (pizza lies on table), we obtain the following search results: “пицца” (pizza) returns the word “пицца” (pizza), “птица” (bird) returns the word “птица” (bird), and the word “пица” (brd) returns two variants “пицца” (pizza) and “птица” (bird). The words can be found even if some letters are missing. It is possible to skip unknown letters, e.g., search for the word “птица” (biZd) will return “птица” (bird).

**Step 3.** Selection of the words within known contexts. When searching for the word “пица лежит” (brd flies), the result will be “пицца лежит” (pizza lies), and for “пица высоко” (brd high), the result will change to “птица высоко” (bird high). Therefore, the missing letter is corrected depending on the attributes.

**Step 4.** The hierarchy level in the output. When searching for elements on “пицца лежит” (pizza lies), the following results are obtained: “на” (on), weight 4; “л” (l), weight 3; “стол” (table), weight 3; and “е” (e), weight 2. When searching for “пицца на” (pizza on), the output is “стол” (table), weight 3. When searching for “птица летает” (bird flies), the output is “высоко” (high), weight 4; “л” (l), weight 3; and “е” (e), weight 2.

**Table 1.** Values of the search results for the sequences of text elements

No.	м о ж	м о ж е	м о ж н
1	“е”, weight: 540586	“т”, weight: 516182	“о”, weight: 567726
2	“н”, weight: 519084	“м”, weight: 148250	“ы”, weight: 190055
3	“т”, weight: 322163	“н”, weight: 102988	“с”, weight: 126629
4	“о”, weight: 313992	же, weight: 78278	должен, weight: 88595
5	“м”, weight: 105815	“с”, weight: 64027	“а”, weight: 70580

## RESULTS AND DISCUSSION: EXAMPLE OF USING PATTERN MEMORY

Consider memorization of Ludwig Wittgenstein’s *Tractatus Logico-Philosophicus* (Russian translation, 2342 sentences). The most recurrent patterns were identified in descending order (the first 20 variants are presented): и; в; не; что; как; есть; [and, in, not, what, how, to be] 5; мы; предложение; —; если; может; предложения; из; то; это; [we, sentence, —, if, can, sentences, from, then, it] 4; быть; с; [to be, with] 6. The most popular patterns by their part are “мож”: “можно”; “можем”, and “можем,” [“can,” “it can,” “we can”, “we can,”]. A comparison of the search results of the obtained elements is presented in Table 1 (a single character is given in single quotes, and a pattern is given without quotes).

The results vary depending on the initial search conditions (Table 1). The results display both individual characters, which occurred directly after the specified sequence and the next nearest but with smaller weight. In some cases, the words that were associated with the given sequence come right after the desired sequence in memory. The number of outputs depends on the data that has been memorized.

The conducted model experiments made it possible to reveal the important properties of the pattern memory. The network method of storing information is used, which is close to the neural network approach of memory organization. At the same time, pattern memory is deterministic, nonlinear, and aperiodic, which is a fairly accurate analog of the special properties of living systems [17].

The process of adding new knowledge and searching through the pattern memory system is sequential and depends linearly on the total number of already available elements, which is a disadvantage of the proposed model. It is possible to build an algorithm for determining and making decisions taking into account the time, the amount of currently available computational resources, and the overall affective evaluation.

At the same time, the proposed concept can technically be extended by adding the ability to create top-level patterns automatically by finding parts of patterns that are already known to combine them with new data, using affective evaluations to calculate the mediator power, incorporating the dominant model [15] for predictions based on current needs [16], and targeting learning to find new hypotheses and solutions.

A direction for further research could be the use of the memory organization proposed in A.A. Zhdanov’s method of autonomous adaptive control [14] for a variant with the use of images, actions, results, and affective evaluations in one pattern and mechanisms of automatic linking of different patterns with images and actions to new images. In this way, it is possible to solve the problem of establishing cause-and-effect relations between the actions performed and the occurring events.

## CONCLUSIONS

The presented principles of logical organization of a neuro-like hierarchical pattern associative memory system make it possible not only to store and retrieve information but also to perform operations simultaneously, including nonlinear identification of functions, time series prediction, etc. It is possible to implement a concept in which one subsystem of pattern memory can produce automatic responses (Pavlov’s reflexes) and the second system can produce algorithmic results (P.K. Anokhin’s functional system).

Associative memory was designed to process large amounts of information and assumes an efficient way to implement mechanisms aimed at storing and processing all elements. The problem of creating devices that would not simply replicate biological processes but could approach the human brain is one of the key areas. The speed of individual elements of microelectronic solutions is millions of times faster than that of biological systems. Nevertheless, the efficiency of universal decisions related to survival and decision-making in the natural environment is higher in living systems. If we follow the analogy with the natural brain of animals or humans, there is no doubt that artificial intelligence should also consist of a hierarchy of subsystems responsible for certain functions similar to living nature.

Despite certain drawbacks, the stated principles of logical organization seem promising for neuroinformatics and require further research and generalizations.

## ACKNOWLEDGMENTS

We thank S.P. Smirnov, Candidate of Engineering Sciences, for scientific ideas and participation in discussions during the research.

## CONFLICT OF INTEREST

The author declare that they have no conflicts of interest.

## REFERENCES

1. Meijia, S. and Xiaoping, W., A memristor-based circuit design for generalization and differentiation on Pavlov associative memory, *Neurocomputing*, 2020, vol. 389, pp. 18–26.  
<https://doi.org/10.1016/j.neucom.2019.12.106>
2. Jalaliddine, S.M., Associative memories and processors: The exact match paradigm, *J. King Saud Univ. - Comput. Inf. Sci.*, 1999, vol. 11, pp. 45–67.  
[https://doi.org/10.1016/S1319-1578\(99\)80003-2](https://doi.org/10.1016/S1319-1578(99)80003-2)
3. Tiba, A.K.O. and Araujo, A.F.R., Control strategies for Hopf bifurcation in a chaotic associative memory, *Neurocomputing*, 2019, vol. 323, pp. 157–174.  
<https://doi.org/10.1016/j.neucom.2018.09.078>
4. Li, L., Pedrycz, W., Qu, T., and Li, Z., Fuzzy associative memories with autoencoding mechanisms, *Knowl.-Based Syst.*, 2020, vol. 191, p. 105090.  
<https://doi.org/10.1016/j.knosys.2019.105090>
5. Ramírez-Rubio, R., Aldape-Pérez, M., Yáñez-Márquez, C., López-Yáñez, I., and Camacho-Nieto, O., Pattern classification using smallest normalized difference associative memory, *Pattern Recognit. Lett.*, 2017, vol. 93, pp. 104–112.  
<https://doi.org/10.1016/j.patrec.2017.02.013>
6. Kobayashi, M., Chaotic pseudo-orthogonalized Hopfield associative memory, *Neurocomputing*, 2017, vol. 241, pp. 147–151.  
<https://doi.org/10.1016/j.neucom.2017.02.037>
7. Miyata, M. and Omori, T., Emergence of symbolic inference based on value-driven intuitive inference via associative memory, *Procedia Comput. Sci.*, 2018, vol. 145, pp. 370–375.  
<https://doi.org/10.1016/j.procs.2018.11.087>
8. Masuyama, N., Loo, C.K., and Seera, M., Personality affected robotic emotional model with associative memory for human-robot interaction, *Neurocomputing*, 2018, vol. 272, pp. 213–225.  
<https://doi.org/10.1016/j.neucom.2017.06.069>
9. Sussner, P. and Schuster, T., Interval-valued fuzzy morphological associative memories: some theoretical aspects and applications, *Inf. Sci.*, 2018, vol. 438, pp. 127–144.  
<https://doi.org/10.1016/j.ins.2018.01.042>
10. Pedrycz, A., Bidirectional and multidirectional associative memories as models in linkage analysis in data analytics: conceptual and algorithmic developments, *Knowl.-Based Syst.*, 2018, vol. 142, pp. 160–169.  
<https://doi.org/10.1016/j.knosys.2017.11.034>
11. De Paula Neto, F.M., da Silva, A.J., de Oliveira, W.R., and Ludermir, T.B., Quantum probabilistic associative memory architecture, *Neurocomputing*, 2019, vol. 351, pp. 101–110.  
<https://doi.org/10.1016/j.neucom.2019.03.078>
12. Majdabadi, M.M., Shokouhi, S.B., and Ko, S.-B., Efficient hybrid cmos/memristor implementation of bidirectional associative memory using passive weight array, *Microelectron. J.*, 2020, vol. 98, p. 104725.  
<https://doi.org/10.1016/j.mejo.2020.104725>
13. Wang L. and Zou H., A new emotion model of associative memory neural network based on memristor, *Neurocomputing*, 2020, vol. 410, pp. 83–92.  
<https://doi.org/10.1016/j.neucom.2020.05.002>
14. Zhdanov, A.A., *Avtonomnyi iskusstvennyi intellekt (Autonomous Artificial Intelligence)*, Moscow: Laboratoriya Znaniy, 2015.
15. Zueva, E.Yu. and Efimov, G.B., Principle of dominance of Ukhtomsky as approach to describing life, *Preprint of Keldysh Inst. for Appl. Math.*, Moscow, 2010, no. 16.
16. Savel'ev, A.V., Ontological extension of the theory of functional systems, *Zh. Probl. Evol. Otkrytykh Sist.*, 2005, no. 1, p. 7.
17. Kulmagambetov, I.R., Koichubekov, B.K., and Riklefs, V.P., Basic concepts of the theory of deterministic chaos and its applied aspects in physiology, *Med. Ekol.*, 2007, no. 1, p. 42.  
<https://cyberleninka.ru/article/n/osnovnye-polozheniya-teorii-determinirovannogo-haosa-i-ee-prikladnye-aspekty-v-fiziologii>

*Translated by O. Pismenov*