

The Associative Recall of Spatial Correlated Patterns

Jana Štanclová

Department of Software Engineering,
Faculty of Mathematics and Physics, Charles University
Malostranské nám. 25, 118 00, Praha 1, Czech Republic
jana.stanclova@mff.cuni.cz

Abstract. The strategies for an associative recall can be based on associative memory models. However, the performance of standard associative memories is very sensitive to the number of stored patterns and their mutual correlations. With respect to huge amounts of spatial patterns (mostly correlated) to be processed, we have focused on an arbitrary number of associative memories grouped into several layers (Hierarchical Associative Memories - HAM). In the newly presented HAM2-model, the patterns are hierarchically grouped according to the “previous-layer” patterns. The HAM2-model uses the information recalled by the “previous-layer” to find an appropriate subset of “next-level” associative memories. To evaluate the performance of the HAM2-model, extensive simulations are carried out. The experimental results show the recall ability of the model in the area of associative pattern recall.

1 Introduction

Let us consider a situation when a traveler moves along a familiar scenery. In such a situation, he can usually see only his close surroundings. Based on his previous knowledge about the whole area, he might be able to recall also some part of the environment that he can not see yet, but will see soon as he moves. The newly recalled scenery in his mind can trigger other associations, he is also able to recall another part of the environment further ahead of him. Thus, he can recall in his mind the scenery of a wide environment by a chain of such recall processes. This process of an associative recall could be used e.g. when building an autonomous robot “warehouse keeper” or an automatic Hoover. It would help with robot localization and thus its better overall control.

For an associative recall of patterns, we use the approach introduced by Fukushima et al. [5]. The actual “seen” scenery is represented in the form of a spatial pattern with an egocentric coordinate system. During the movement, the actual area becomes shifted relatively to the previous position of the move (in order to keep the body always in the center of the pattern to be recalled). As the scenery image shifts following the movement, a vacant region appears in the “not yet seen” part of the image. The pattern with a vacant “not yet seen” region (the so-called incomplete pattern) is presented to an associative memory and the empty part of the pattern is expected to be filled (see Figure 1).

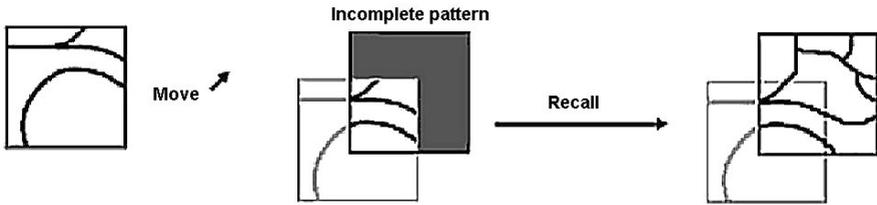


Fig. 1. The associative pattern recall inspired by Fukushima et al. [5]

Anyway, the associative memory by itself does not reliably recall shifted patterns, and each presented pattern has to be placed accurately at the location of one of the memorized patterns. Different approaches to an associative recall (e.g. in spatial maps) can be based on the so-called cognitive maps. Different approaches have been developed to build hierarchical cognitive maps (e.g. [2], [4], [12]).

For an associative recall, standard associative memories can be used. However, this model is not suitable for the associative recall mentioned above, as it cannot cope with the need for a high number of stored patterns and the fact that the patterns are not orthogonal. Therefore, we decided to focus our research on the models of Hierarchical Associative Memories (HAM) developed with an emphasis on the necessity to process large amounts of (correlated) data. They consist of an arbitrary number of associative memories grouped into several layers. In our original model HAM1 (described e.g. in [11]), the patterns are stored in “any suitable” associative memory. The HAM1-model does not use the “previous-layer” information to find an appropriate associative memory at the “next” layer. In this paper, we present the HAM2-model. The associative memories of the HAM2-model form a tree structure. A disjoint subset of associative memories is assigned to every “previous-layer” associative memory. We expect that the HAM2-model improves the HAM1-model and allows a reliable storage and “sufficiently reliable” recall of correlated patterns with respect to an associative pattern recall.

The organization of the paper is as follows: a basic concept of associative memories is reviewed in section 2. In section 3, the models HAM1 and HAM2 are described. In section 4, we present experimental results evaluating the performance of the models HAM1 and HAM2. The paper concludes with Section 5, which outlines the directions of our future work.

2 Models of Associative Memories

The standard associative memory is a neural network, for which all its neurons are input and output neurons simultaneously and there are oriented interconnections among all neurons. All their weights are symmetric and each neuron is connected to all other neurons except itself. Other basic notions and characteristics of this memory can be found e.g. in [7]. If the number of stored patterns does not exceed the limit ($0.15n$ where n is the dimension of stored patterns [7]), the models have robust recall ability. Although the robust recall ability is very attractive, patterns to be stored in the standard associative memory must be almost orthogonal one to each other. Storing

correlated patterns can cause serious problems and previously stored training patterns can even become lost because the cross-talk does not average to zero [1].

To overcome this problem, many researchers have extended the standard associative memory to process correlated patterns (e.g. [3], [6], [8], [9], [10]). Morita [10] proposed a model that enhances the ability of associative memories with non-monotonic dynamics. Gutfreund [3] proposed a model consisting of two associative memories - one for each level of the two-level hierarchy. The first associative memory (AM1) and the second one (AM2) store the first-level patterns (called ancestors) and their second-level patterns (called descendants), respectively. However, Gutfreund's model has a parameter on which the storage capacity strongly depends [3]. Hirahara et al. [6] proposed a model of Cascade Associative Memory (CASM) similar in the structure to Gutfreund's model. The CASM-model is characterized by the AM2 storing not the descendants but the so-called difference patterns. The difference patterns become sparser with increasing correlation, which allows the CASM-model to have a larger storage capacity. Unfortunately, the upper limit of the storage and the recall abilities of the CASM-model are bounded by that of the AM1 ($\sim 0.15n$) [6].

3 The Hierarchical Associative Memory (HAM)

The performance of standard associative memories is limited by the number of patterns which can be stored in it and the fact that the patterns have to be orthogonal. To avoid (at least to a certain extent) these limitations, we have designed the Hierarchical Associative Memory models. These models are based on the concept of the CASM-model [6]. Our goal is to use the CASM-model more generally by allowing an arbitrary number of layers with more memories grouped in each layer.

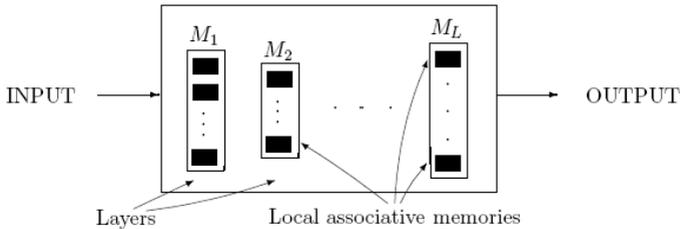


Fig. 2. The structure of the Hierarchical Associative Memory with L layers M_1, \dots, M_L

A Hierarchical associative memory H with L ($L > 0$) layers is an ordered tuple $H = (M_1, \dots, M_L)$ where M_1, \dots, M_L are finite non-empty sets of associative memories (the so-called local associative memories). A set M_k ($k = 1, \dots, L$) is called the layer of the memory H . $|M_k|$ denotes the number of local associative memories in the layer M_k ($k = 1, \dots, L$). Every local associative memory of the same layer has the same number of neurons n ($n > 0$). A training tuple T of H is an ordered tuple $T = (T_1, \dots, T_L)$ where T_k ($k = 1, \dots, L$) is a finite non-empty set of training patterns for the layer M_k . The structure of the HAM2-model, as well as the HAM1-model, is shown in Figure 2. The difference between the HAM1-model and the HAM2-model is in the network dynamics.

3.1 The HAM1-Model

To train the HAM1-model, we have designed the so-called dynamical layer training algorithm (DLT-algorithm) [11]. Each of the layers M_k ($k=1, \dots, L$) is trained separately. The training patterns from the set T_k are stored in local associative memories of the corresponding layer M_k . During the training of the layer M_k , training patterns from the set T_k are presented to the layer M_k sequentially. Each training pattern x is stored in such a local associative memory of the layer M_k where the pattern x (or its “noisy” version¹) is recalled correctly. If there is no “suitable” local associative memory, a new local associative memory is created, added to the layer M_k and the pattern x is stored in the newly created associative memory.

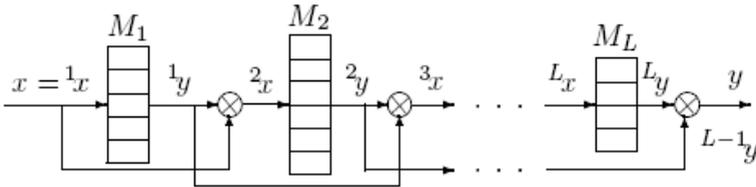


Fig. 3. The recall process in the HAM1-model with the layers M_1, \dots, M_L

The recall process of the HAM1-model is depicted in Figure 3. During the recall, a pattern x is an input to the HAM1-model. The input pattern x represents an input for the first layer M_1 (i.e. $x=^1x$). Within every time step k ($1 \leq k \leq L$), the layer M_k produces the corresponding output $^k y$. The output $^k y$ combined with the output $^{k-1} y$ of the “previous” layer M_{k-1} is used as the input $^{k+1} x$ to the “next” layer M_{k+1} . The function of the circle “ \otimes ” in Figure 3 is to produce the “next” input (in our implementation, the difference pattern calculation is used). The output $^L y$ of the “last” layer M_L combined with the output of the layer M_{L-1} represents the output y of the HAM1-model.

At the layer M_k , the input pattern $^k x$ is propagated to all local associative memories of the layer M_k and each of them recalls the corresponding outputs $^k y^i$ ($i=1, \dots, /M_k /$). The output $^k y$ of the layer M_k is an output $^k y^i$ which is “the most similar” to the input pattern $^k x$ (we use Hamming distance, but other metrics may be considered too).

3.2 The HAM2-Model

A basic concept of the HAM2-model is similar to the HAM1-model. The main difference is that the local associative memories of the HAM2-model form a tree structure. A disjoint subset of local associative memories is assigned to every “previous-layer” local associative memory. The HAM2-model groups the associative memories according to their “previous-layer” information. The subsets of local associative memories are organized in a tree structure (Figure 4).

¹ A pattern in which certain number of randomly selected elements change their value.

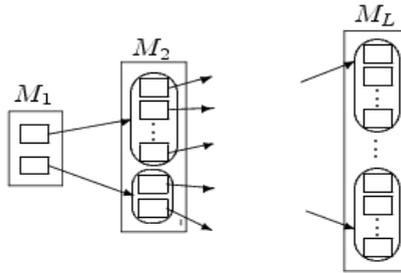


Fig. 4. The tree structure of the HAM2-model with the layers M_1, \dots, M_L

All layers M_k ($k=1, \dots, L$) in the HAM2-model are trained sequentially. The training patterns from the training set T_k are presented to the corresponding layer M_k sequentially. The training of every pattern \mathbf{x} consists of two steps. First, the pattern \mathbf{x} from T_k is presented to already trained layers M_1, \dots, M_{k-1} to be recalled. The output of the layer M_{k-1} provides the information into which subset of the currently trained layer M_k the pattern \mathbf{x} belongs (S denotes the corresponding subset of the layer M_k for the pattern \mathbf{x}). As the corresponding subset S is detected, the “suitable” local associative memory in the subset S of the layer M_k is found and the pattern \mathbf{x} is stored in it. If there is no “suitable” local associative memory in the subset S of the layer M_k , a new local associative memory is created and added to the subset S of the layer M_k . In this case, the training pattern \mathbf{x} is stored in the newly created associative memory.

The recall process is similar to the recall process of the HAM1-model, but it utilizes the tree structure of the HAM2-model. The recall process takes place sequentially in the layers of the HAM2-model. During the recall, a pattern \mathbf{x} is an input to the HAM2-model (i.e. $\mathbf{x} = {}^l\mathbf{x}$). At every time step k ($1 \leq k \leq L$), the layer M_k receives the input ${}^k\mathbf{x}$ and the recall process at this layer proceeds. First, the output ${}^{k-1}\mathbf{y}$ of the “previous” layer M_{k-1} ($k > 1$) gives the information into which subset S of the layer M_k the input pattern ${}^k\mathbf{x}$ belongs. Then, the pattern ${}^k\mathbf{x}$ is sent only to such local associative memories that belong to the subset S of the layer M_k . Each of these local associative memories recalls the outputs ${}^k\mathbf{y}^i$ ($i=1, \dots, |S|$). The output ${}^k\mathbf{y}$ of the layer M_k is an output ${}^k\mathbf{y}^i$ which is “the most similar” to the input pattern ${}^k\mathbf{x}$ (we use Hamming distance, but other metrics may be considered too). The output ${}^k\mathbf{y}$ combined with the output ${}^{k-1}\mathbf{y}$ of the “previous” layer M_{k-1} is used as the input ${}^{k+1}\mathbf{x}$ to the “next” layer M_{k+1} . The output of the “last” layer M_L represents the output of the HAM2-model.

3.3 Characteristics of the HAM2-Model

In the HAM2-model, there is no need to state the number of the local associative memories before the training process starts because the local memories are automatically added to corresponding layers during training. The training process can start with one local associative memory in every layer. The initial structure of the HAM2-model forms a path (i.e. a degenerated tree). Other local associative memories are added to the HAM2-model during the training process according to the incoming patterns. The number of local associative memories in the HAM2-model varies with

the structure of incoming data. The number of subsets in every layer depends on the structure of incoming patterns, too.

In the training algorithm, we use a simple heuristics how to choose the “suitable” local associative memory for storing a pattern. The above-sketched heuristics is quick, simple and easy to implement, but it is not optimal. A pattern remains stored in such a local associative memory (of the corresponding subset) where the pattern or its “noisy version” is recalled correctly. However, using this training algorithm we cannot predict anything about recalling previously stored patterns. After storing another pattern, some of the previously stored patterns can be recalled incorrectly or even become lost.

4 Experimental Results

One of the key points in neural network performance is the recall ability. Applying the HAM2-model to obtain an associative recall (e.g. in an autonomous robot), a robust recall of presented patterns (images) is required, often unknown in some parts of their surface. The experimental simulations are focused on the analyses of the HAM2-model recall abilities (in comparison with the HAM1-model and the standard associative memory model).

4.1 The Implementation of the HAM2-Model

Our experimental simulations are restricted to a two-layer hierarchy of the HAM2-model (and the HAM1-model as well). The two-layer hierarchy is chosen for an easy geometrical interpretation of data. We can believe that the patterns in an input space are grouped into clusters. In the centers of the clusters are representative patterns (called ancestors) that form the first-layer patterns. Other patterns similar (correlated) to any ancestor are distributed in a cluster around it. They represent the second-level patterns, called descendants. The generalization of the data structure to an arbitrary number of layers is straightforward but the geometrical interpretation does not have to be so simple (in the future, we plan to extend the experiments to a high number of layers).

With respect to the robust recall ability requirement, the second layer of the implemented HAM2-model deals with difference patterns instead of descendants. The difference patterns contain only the information on the differences between the descendants and the corresponding ancestor. During the training, the ancestors are stored in the first layer of the HAM2-model. Afterwards, the corresponding ancestor is found for every descendant (e.g. by recalling in the first - already trained - layer of the HAM2-model). The difference pattern (descendant – recalled ancestor) is calculated and stored in the second layer according to the training algorithm. The second layer of the HAM2-model is in the form of a pile of associative memory subsets. Each subset is responsible for recalling only the difference patterns belonging to the ancestor stored in the corresponding local associative memory of the first layer.

During recalling, an input pattern is presented to the first layer of the HAM2-model to recall the corresponding ancestor. Once the ancestor is recalled, the corresponding

difference pattern is calculated by combining the recalled ancestor with the input pattern. The difference pattern is propagated to the second layer of the HAM-model to be recalled. The output of the HAM2-model is produced by combining the recalled difference pattern with its recalled ancestor (by the operation inverted to the difference pattern calculation).

4.2 The Generation of Patterns

For experiments, we generate 100 sets of 100 randomly generated bipolar patterns; each of size 15×15 elements. In a bipolar pattern, every element takes the value +1 or -1. Every experiment is run on its set of patterns independently on other sets. Experiments are repeated for every data set. In our experiments, we process “relatively small” patterns, as it is necessary to perform huge number of experiments to analyze the HAM2-model recall ability. We have performed also experiments with “bigger” data (75×75 elements) and the results have been very similar (or even a little better). After pattern generation, the patterns of every data set with the smallest cumulative correlation between the respective patterns are chosen to be the ancestors and the remaining patterns are used to form the descendants. We define the pattern rate r as a ratio between the number of ancestors and descendants in a set of patterns.

4.3 The Associative Recall of Stored Patterns

First, we analyze the HAM2-model ability to recall the stored patterns and compare the HAM2-model abilities with the HAM1-model and standard associative memory abilities. We measure the percentage of patterns (from a set of patterns) that are recalled correctly. A pattern is recalled correctly if it coincides with its original. We define the capacity coefficient c that reduces the maximum number of patterns stored in a local associative memory. The maximum storage capacity α of a local associative memory is given by formula $\alpha = 0.15 \cdot n \cdot c$. The capacity coefficient c takes the value of 1 for a standard associative memory. For decreasing c , the maximum storage capacity is reduced and the HAM2-model recall ability rises. The results are shown in Figure 5.

In Figure 5, the surface represents the average number of patterns recalled correctly. If $c \leq 0.5$, the results are very similar in both models (for every pattern rate r). In this case, more than 97% of stored patterns are recalled correctly. As the capacity coefficient c (and the pattern rate r) increases, the differences between the HAM1- and the HAM2-model become significant. The increasing pattern rate r corresponds to the increasing number of ancestors (and the increasing number of local associative memories in the first layer). The recall abilities are enhanced in the HAM2-model (especially for $r > 1/2$) due to the influence of the tree structure.

The same experiments are performed with the standard associative memory. As the number of patterns to be stored exceeds the theoretical storage capacity [7], the standard associative memory is not able to recall correctly any of the stored patterns. The experiments show that the HAM2-model has a robust recall ability to recall correctly the stored patterns. The performance of the model depends on the capacity coefficient c and the pattern rate r . Several patterns are recalled with an error, but the error does not exceed 2.5%.

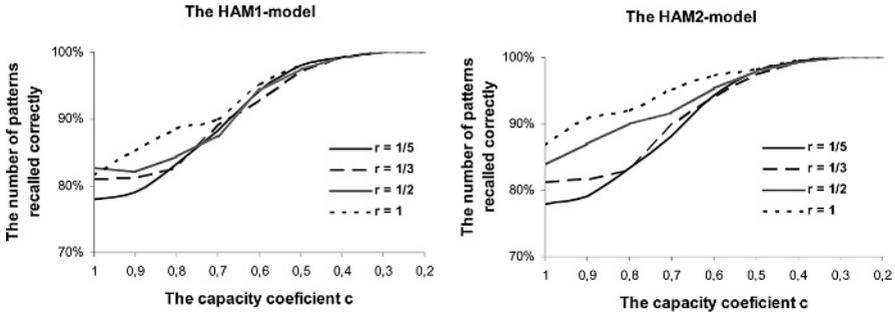


Fig. 5. The number of patterns recalled correctly in the HAM1-model (left) and the HAM2-model (right) with respect to the capacity coefficient c and the pattern rate r

4.4 The Associative Recall of Incomplete Patterns

Now, we focus on the HAM2-model ability to recall incomplete patterns (e.g. images). According to the above-mentioned results (Figure 5), we concentrate on the HAM2-model with the capacity coefficient $c=0.7$ and the pattern rate $r=1$. We create three groups of incomplete patterns containing 13%, 25% and 36% of unknown elements, respectively. Each of the groups corresponds to a diagonal shift of the pattern (image) by 1, 2, or 3 points, respectively. We process four types of diagonal directions: \nwarrow (northwest), \nearrow (northeast), \swarrow (southwest) and \searrow (southeast). In practice, other directions are also possible. The experiments are processed for every diagonal direction separately and the results are averaged for the corresponding size of the shift.

We analyze the model ability to recall the incomplete patterns with respect to the different size of the shift. With the increasing number of the „unknown“ elements, the recall ability decreases. In practice, it is not necessary to recall the incomplete patterns correctly (i.e. a small error is acceptable). The results of incomplete pattern recall with the shift by 1 and 2 points are shown in Figure 6.

The horizontal axis in Figure 6 corresponds to the acceptable error (the number of elements in a pattern that can be recalled incorrectly). The vertical axis shows the number of patterns in which the recall error is below the corresponding acceptable error level. Figure 6 shows the recall abilities of the HAM2-model are higher than the HAM1-model but the differences are just minor. We also perform the same experiments with the standard associative memory. Because of the incapability to recall any of the stored patterns, the standard associative memory is not able to recall the incomplete patterns correctly either. The detailed results of the HAM2-model for two selected acceptable errors are summarized in the Table 1.

With increasing error acceptance the number of patterns increases. When the “unknown” area is small and the error acceptance is about 6%, the HAM2-model has a relatively robust recall ability with respect to the incomplete pattern recall (82%). Moreover, theory says that for every pattern associative memories store an inverse pattern as well. It is possible that the inverse pattern is recalled instead of the original one. In this situation, the pattern is considered to be recalled incorrectly. If we could detect the situation, the recall ability of the HAM2-model would be improved.

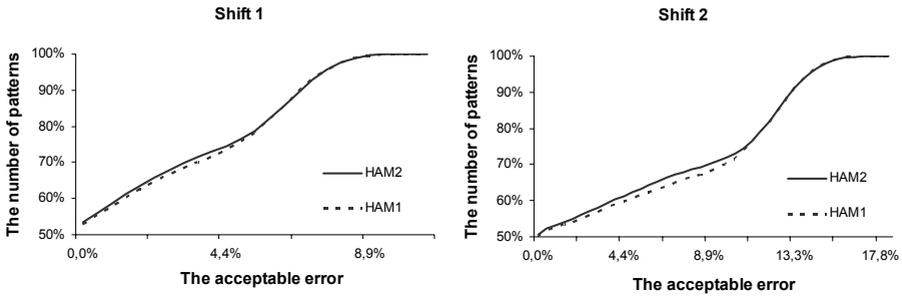


Fig. 6. The number of incomplete patterns recalled by the HAM2-model (the solid line) and the HAM1-model (the dotted line) with the error below the acceptable error. The “unknown” elements correspond to the diagonal shift by 1 point (left) and 2 points (right)

Table 1. The number of incomplete patterns recalled by the HAM2-model where the recall error is below 1% and 6%

Acceptable error	Shift by 1	Shift by 2	Shift by 3
$\leq 1\%$	59%	53%	51%
$\leq 6\%$	82%	64%	57%

Experiments for the HAM2-model with different pattern rates r are also performed. The results show that the HAM2-model ability to recall incomplete patterns are increased with the increasing pattern rate r .

5 Conclusions

Our current research in the area of associative memories is focused on an associative pattern recall with necessity to handle large number of correlated patterns. Here, we have presented the HAM2-model that improves the storage/recall ability of standard associative memories and our previous HAM1-model as well. The HAM2-model has a tree structure and the patterns are hierarchically grouped according to the “previous-layer” information. The recall ability of the HAM2-model can be further improved by restricting the storage capacity of local memories. The experiments carried out have confirmed the legitimacy of the proposed HAM2-model and shown promising results for the associative pattern recall. However, for real applications it is necessary to further improve the robustness of recall process with respect to incomplete patterns.

The right choice of a local associative memory for storing a pattern represents an important point of a successful overall performance of the model. We have used a basic straightforward method, which we try to improve. We hope it will further increase the robustness of the HAM2-model. In the future, we plan to compare the HAM2-model with the CASM-model of Hirahara [6] and carry out the experiments with the HAM2-model consisting of more than two layers.

Acknowledgments. This work was partially supported by the Ministry of Education of the Czech Republic (grant MSM0021620838).

References

1. Amit, D. J., Gutfreund, H., Sompolinsky, H.: Information storage in neural networks with low levels of activity, in: *Physical Review A*, 35, 2293-2303, 1987
2. Donnar, J. Y., Meyer, J. A.: Hierarchical-map building and selfpositioning with Mona-Lysa, *Adaptive Behavior*, 5(1), 29–74, 1996
3. Gutfreund, H.: Neural networks with hierarchically correlated patterns. *Physical Review A*, 37, 570–577, 1988
4. Kuipers, B.: The spatial semantic hierarchy, *Artificial Intelligence*, 119, 191–233, 2000
5. Fukushima, K., Yamaguchi, Y., Okada, M.: Neural Network Model of Spatial Memory: Associative Recall of Maps, *Neural Network*, Vol. 10, No. 6, 971-979, 1997.
6. Hirahara, M., Oka, N., Kindo, T.: A cascade associative memory model with a hierarchical memory structure, *Neural Networks*, Vol. 13, No. 1, 41-50, 2000.
7. Hopfield, J. J.: Neural Networks and physical system with emergent collective computational abilities, *Proc. Natl. Acad. Sci. USA*, 79, 2554-2558, 1982.
8. Kimoto, T., Okada, M.: Coexistence of memory patterns and mixed states in a sparsely encoded associative memory model storing ultrametric patterns, *Biological Cybernetics*, 90, 229–238, 2004
9. Matsumoto, N., Ide, D., Watanabe, M., Okada, M.: Synaptic Depression Enlarges Basin of Attraction, *Neurocomputing*, 65-66, 571-577, 2005
10. Morita, M.: Associative memory with nonmonotone dynamics. *Neural Networks*, 6, 115–126, 1993
11. Štanclová, J., Zavoral, F.: Hierarchical Associative Memories: The Neural Network for Path Prediction in Spatial Maps, *Proc. ICIAP 2005*, Springer-Verlag LNCS3617, 786-793, 2005
12. Voicu, H. : Hierarchical cognitive maps, *Neural Networks*, Vol. 16, pp. 569-576, 2003